Governing Governance
A formal framework for analysing institutional design and enactment governance

King, Thomas

DOI
10.4233/uuid:82438672-3e8b-477a-a39e-0ce189639e88

Publication date
2016

Document Version
Final published version

Citation (APA)
King, T. (2016). *Governing Governance: A formal framework for analysing institutional design and enactment governance*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:82438672-3e8b-477a-a39e-0ce189639e88

Governing Governance

A Formal Framework for Analysing Institutional Design and Enactment Governance

Governing Governance

A Formal Framework for Analysing Institutional Design and Enactment Governance

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. Ir. K.Ch.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 27 oktober 2016 om 10.00 uur

door

Thomas Christopher King

Bachelor of Science in Computer Science from King's College London, United Kingdom
geboren te Londen, Verenigd Koningkrijk

This dissertation has been approved by the promotors:

Promotor: Prof. dr. C. M. Jonker
Copromotor: Dr. M. V. Dignum
Copromotor: Dr. M. B. van Riemsdijk

Composition of the doctoral committee:

Rector Magnificus,	chairman
Prof. dr. C. M. Jonker,	Delft University of Technology, promotor
Dr. M. V. Dignum,	Delft University of Technology, copromotor
Dr. M. B. van Riemsdijk,	Delft University of Technology, copromotor
Independent members:	
Prof. dr. K. Atkinson,	University of Liverpool
Prof. dr. F. Brazier,	Delft University of Technology
Prof. dr. J. Pitt,	Imperial College London
Prof. dr. L. van der Torre,	University of Luxembourg
Reserve:	
Prof. dr. C. Witteveen	Delft University of Technology

SIKS Dissertation Series No. 2016-41
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.
The research reported in this thesis was funded by the SHINE project of TU Delft.

ISBN 978-94-6186-726-1

An electronic version of this dissertation is available at
http://repository.tudelft.nl/

SUMMARY

This dissertation is motivated by the need, in today's globalist world, for a precise way to enable governments, organisations and other regulatory bodies to evaluate the constraints they place on themselves and others. An organisation's modus operandi is enacting and fulfilling contracts between itself and its participants. Yet, organisational contracts *should* respect external laws, such as those setting out data privacy rights and liberties. Contracts *can* only be enacted by following contract law processes, which often require bilateral agreement and consideration. Governments need to legislate whilst understanding today's context of national and international governance hierarchy where law makers shun isolationism and seek to influence one another. Governments *should* avoid punishment by respecting constraints from international treaties and human rights charters. Governments *can* only enact legislation by following their own, pre-existing, law making procedures. In other words, institutions, such as laws and contracts are designed and enacted under constraints.

The common thread shared by these examples is that institution designers, such as organisations and governments, are constrained in two senses. They are loosely tied in how the institutions they enact *should* be designed and strongly tied in how and when they *can* enact institutional changes. We can clearly see such constraints exist in written form, even though we cannot physically see how the machinery that applies and tightens those constraints around the institutional designers works. It is consequently hard to grasp exactly how institution designers need to operate under these constraints.

This dissertation addresses this issue by contributing a formal framework for analysing institutional design and enactment governance. Through formalisation, the framework provides a mathematically rigorous account of constraints placed on institutional designs and enacting institutional changes. Hence, the unseen constraints become seeable not as physical bindings, but as symbols on a page defining general institutional reasoning. From a conceptualisation standpoint, the main benefit is that we are able to understand the institutional constraints and identify any conceptual flaws. Or, in the words of Leibniz, "The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance".

There are three main practical benefits. Firstly, the formal reasoning is disseminated in clear mathematical language. Institution designers and judiciaries apply exactly the same reasoning, making judgements predictable, providing they share the same legal rule bases. Moreover, citizens can have common knowledge of when regulatory changes are enacted by applying the same institutional reasoning. Secondly, the framework facilitates automation of otherwise cognitively difficult tasks in understanding the constraints placed on institution designers. Thirdly, the framework deals with the pragmatics with applications to real world case studies, in order to capture the meaning of institutional constraints as they are used.

In summary, this dissertation introduces the PARAGon framework for **P**ractical **A**utomated **R**easoning for **A**ssessing **Go**ver**n**ance of institution design and change.

Central to the PARAGon framework is Searle's well known institutional constitutive counts-as rules of the form “A counts-as B in context C”. For example, “a piece of paper with certain European Union symbols counts-as money in the context of the European Union”. These rules build a social reality from brute facts and regulate social reality change when the brute facts change. PARAGon makes three main contributions founded on counts-as rules.

PARAGon contributes formalised reasoning for a governance architecture called *multi-level governance* where institutions operate at different governance levels. The lowest-level institutions prescribe concrete regulations to govern societies (e.g. national legislation), whilst higher-level level institutions impose abstract regulations on the concrete regulatory outcomes of lower-level institutions (e.g. directives, human rights charters and supranational agreements). The PARAGon framework derives from counts-as rules whether concrete regulations at lower governance levels can be applied in social contexts such that they violate abstract regulations as set out in directives, human rights charters, etcetera.

PARAGon contributes a computational mechanism for finding explanations for non-compliant institution designs used to rectify non-compliance. The explanations are minimal counts-as rule additions, deletions and modifications that ensure compliance. Moreover, the explanations keep institution designs as close as possible to designers’ original intentions.

PARAGon formalises constraints called secondary legal rules, which define when institutions and institutional changes can be enacted. PARAGon formalises such secondary rules as *rule-modifying* counts-as rules, which ascribe rule change at various points in time. For example, a government voting for a rule change *counts-as* enacting a rule change. Determining rule change legality is difficult, since changing counts-as rules alters social contexts, which rule changes are conditional on. PARAGon contributes formal reasoning for determining when rule changes count-as legal rule changes.

The PARAGon framework was developed under the SHINE (Sensing Heterogeneous Information Network Environment) project[1], which aims to form large-scale heterogeneous sensor networks, using existing sensors in the environment belonging to external stakeholders. PARAGon aims to support automated governance for large scale heterogeneous ‘SHINE’ sensor networks. Firstly, by automating the application of multi-level governance to forming SHINE sensor super-systems of sub-systems, comprising a thin SHINE institution layer abstractly governing the design of sub-system institutions towards coordinated regulations for collecting environmental data collection. Secondly, PARAGon supports sensor network stakeholders with automated institution re-design recommendations to ensure the institutions are designed compliantly. Thirdly, PARAGon supports automated reasoning for how and when sensor system regulations are changed over time as different governance needs arise (e.g. as the data needs change or as it emerges existing rules are inadequate).

The PARAGon framework makes both formal and practical contributions. From the formal side, this dissertation aims to formalise previously informal notions provided by political science and legal philosophy. Particular attention is paid to understanding the new conceptualisations and testing them against a number of case studies to assess whether the formal contributions provide ‘correct’ inferences. From the practical side, the contributed reasoning is either coupled with an implementation or a computational characterisation of the formal concepts providing necessary details for implementation. This dissertation lies at the intersection of *legal philosophy and symbolic artificial intelligence*.

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

Samenvatting

Ten grondslag aan deze dissertatie ligt de behoefte in de hedendaagse globalistische wereld aan een precieze methode om overheden, organisaties en andere regelgevende instanties in staat te stellen de beperkingen die ze opleggen aan zichzelf en anderen te evalueren. De modus operandi van een organisatie bestaat uit het instantiëren en uitvoeren van verbintenissen tussen zichzelf en haar deelnemers. Organisationele verbintenissen dienen echter wel de externe wetgeving te respecteren, zoals de wetgeving waarin de rechten en vrijheden op het gebied van gegevensbescherming zijn neergelegd. Verbintenissen kunnen alleen worden geïnstantieerd door verbintenissenrechtelijke processen te volgen, die vaak bilaterale overeenstemming en afweging vereisen. Overheden moeten wetten opstellen en daarnaast inzicht hebben in de hedendaagse context van nationale en internationale governance-hiërarchie waarin wetgevers isolationisme uit de weg gaan en elkaar proberen te beïnvloeden. Overheden dienen sancties te voorkomen door zich te houden aan de beperkingen die zijn opgelegd via internationale verdragen en mensenrechtenhandvesten. Overheden kunnen wetten alleen maar instantiëren door hun eigen, reeds bestaande wetgevingsprocedures te volgen. Met andere woorden: het ontwerpen en instantiëren van instituties, zoals wetten en verbintenissen, is onderworpen aan beperkingen.

De rode draad bij deze voorbeelden is dat ontwerpers van instituties, zoals organisaties en overheden, in twee opzichten beperkt zijn. Ze zijn lichtelijk gebonden in hoe de instituties die ze instantiëren moeten worden ontworpen, en sterk gebonden in hoe en wanneer ze institutionele veranderingen kunnen instantiëren. We zien duidelijk dat dergelijke beperkingen op schrift bestaan, ook al kunnen we niet fysiek zien hoe de machinerie werkt die de beperkingen toepast en ze rond de institutionele ontwerpers aantrekt. Derhalve is het moeilijk precies te begrijpen hoe ontwerpers van instituties moeten opereren te midden van deze beperkingen.

Deze dissertatie gaat in op deze problematiek door een formeel raamwerk aan te reiken voor het analyseren van institutioneel ontwerp en instantiatie-governance. Door formalisering biedt het raamwerk een mathematisch rigoureus overzicht van de beperkingen die opgelegd zijn aan institutionele ontwerpen en instantiatie van institutionele veranderingen. Daarmee worden de onzichtbare beperkingen zichtbaar, niet in fysiek opzicht, maar als symbolen op een pagina waarop algemeen institutioneel redeneren wordt gedefinieerd. Vanuit het oogpunt van conceptualisatie is het grootste voordeel dat we in staat zijn de institutionele beperkingen te begrijpen en eventuele conceptuele tekortkomingen te identificeren. Oftewel, in de woorden van Leibniz, "De enige manier om onze redeneringen te corrigeren, is ze net zo tastbaar te maken als die van de Mathematici, zodat we een fout van ons in één oogopslag kunnen ontwaren".

Er zijn drie grote praktische voordelen. In de eerste plaats wordt formeel redeneren alom verspreid in heldere mathematische taal. Institutie-ontwerpers en rechters passen exact dezelfde redeneringen toe, waardoor uitspraken voorspelbaar worden, mits ze uitgaan van dezelfde juridische grondslagen. Bovendien kunnen burgers door toepassing van

dezelfde institutionele redeneringen gemeenschappelijke kennis hebben van het moment waarop wijzigingen in regelgeving worden geïnstantieerd. In de tweede plaats faciliteert het raamwerk automatisering van anderszins cognitief lastige taken bij het begrijpen van de beperkingen die zijn opgelegd aan ontwerpers van instituties. In de derde plaats gaat het raamwerk in op de pragmatica middels toepassingen op casestudies uit de praktijk, om zo de betekenis van gehanteerde institutionele beperkingen te kunnen begrijpen.

Samenvattend, introduceert deze dissertatie het PARAGon-raamwerk voor **P**ractical **A**utomated **R**easoning for **A**ssessing **G**overnance met betrekking tot het ontwerpen en wijzigen van instituties.

Een centrale plaats in het PARAGon-raamwerk wordt ingenomen door Searle's welbekende institutionele constitutieve "geldt als"-regels in de vorm "A geldt in context C als B". Voorbeeld: "een stuk papier met bepaalde aanduidingen van de Europese Unie geldt in de context van de Europese Unie als geld". Deze regels bouwen een sociale realiteit van brute feiten op en reguleren veranderingen in de sociale realiteit wanneer de brute feiten veranderen. PARAGon levert drie hoofdbijdragen die gebaseerd zijn op geldt-als-regels. PARAGon biedt geformaliseerd redeneren voor een governance-architectuur, multilevel governance genaamd, waarbij instituten op meerdere governance-niveaus opereren. De instituties op het laagste niveau schrijven concrete reguleringen voor om samenlevingen te besturen (bijv. nationale wetgeving), terwijl instellingen op hoger niveau abstracte reguleringen opleggen met betrekking tot de concrete regulerende uitkomsten van lagere instituties (bijv. richtlijnen, mensenrechtenhandvesten en supranationale overeenkomsten). Het PARAGon-raamwerk leidt van geldt-als-regels af of concrete reguleringen op lagere governance-niveaus zodanig kunnen worden toegepast in sociale contexten dat ze een inbreuk vormen op abstracte reguleringen als neergelegd in richtlijnen, mensenrechtenhandvesten, enzovoort. PARAGon reikt een berekeningsmechanisme aan voor het vinden van verklaringen voor niet-compliante institutie-ontwerpen die gebruikt worden om non –compliantie te herstellen. De verklaringen zijn ten opzichte van geldt-als-regels minimale toevoegingen, verwijderingen en aanpassingen die zorgen voor compliantie. Verder houden de verklaringen de institutie-ontwerpen zo dicht mogelijk bij de oorspronkelijke bedoelingen van de ontwerpers.

PARAGon formaliseert beperkingen, de zogenaamde secundaire rechtsregels, die definiëren wanneer instituties en institutionele wijzigingen geïnstantieerd kunnen worden. PARAGon formaliseert dergelijke secundaire regels als rule-modifying geldt-als-regels, die op verschillende momenten wijzigingen van regels toekennen. Een stemming in het parlement om een regel te wijzigen geldt bijvoorbeeld als het instantiëren van een regelwijziging. Het bepalen van de legaliteit van regelwijzigingen is lastig aangezien een verandering van geldt-als-regels leidt tot een verandering van sociale contexten, waar regelwijzigingen afhankelijk van zijn. PARAGon biedt een formele redenering voor het bepalen wanneer regelwijzigingen gelden als wijzigingen van rechtsregels.

Het PARAGon-raamwerk is ontwikkeld in het kader van het project SHINE (Sensing Heterogeneous Information Network Environment)[2], dat beoogt grootschalige heterogene sensornetwerken te vormen met behulp van bestaande sensoren in de omgeving die toebehoort aan externe belanghebbenden. PARAGon beoogt ondersteuning van geautomatiseerde governance voor grootschalige heterogene SHINE-sensornetwerken. In de eerste

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

plaats door automatisering van de toepassing van multi-level governance op de vorming van SHINE sensor-supersystemen van subsystemen, bestaande uit een dunne SHINE institutielaag die op abstracte wijze het ontwerp van subsysteem-instituties aanstuurt richting gecoördineerde regulering voor het verzamelen van milieugegevens. In de tweede plaats ondersteunt PARAGon belanghebbenden in het sensornetwerk met aanbevelingen voor het geautomatiseerd herontwerpen van instituties om zo te waarborgen dat de instituties op compliante wijze worden ontworpen. In de derde plaats ondersteunt PARAGon geautomatiseerde redenering voor hoe en wanneer regels van sensorsystemen in de loop der tijd worden gewijzigd naarmate andere governance-behoeften ontstaan (bijv. als de data gewijzigd moeten worden of als blijkt dat bestaande regels ontoereikend zijn).

Het PARAGon-raamwerk levert zowel formele als praktische bijdragen. In formeel opzicht streeft deze dissertatie naar het formaliseren van voorheen informele noties die vanuit de politieke wetenschap en de rechtsfilosofie werden aangeleverd. Bijzondere aandacht wordt besteed aan het begrijpen van de nieuwe conceptualisaties en het toetsen ervan aan een aantal casestudies teneinde te beoordelen of de formele bijdragen resulteren in 'correcte' inferenties. In praktisch opzicht wordt de aangedragen redenering gekoppeld aan ofwel een implementatie ofwel een rekenkundige karakterisering van de formele concepten die de noodzakelijke details aanleveren voor implementatie. Deze dissertatie bevindt zich op het raakvlak van rechtsfilosofie en symbolische artificiële intelligentie.

Contents

List of Figures

List of Tables

1

Introduction

The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate [calculemus], without further ado, to see who is right.[153]

Gottfried Wilhelm Leibniz

In today's increasingly connected world, governments, organisations and other regulatory bodies do not operate in isolate free from control. Let us take a look at three examples. Our first example concerns the SHINE[1] project under which this dissertation's research was conducted. SHINE aims to form systems comprising heterogeneous environmental sensors (e.g. cellphone cameras, weather radars) contracted from the sensor owners to collect environmental data. However, constraints are placed on forming contracts, in the sense that a sensor owner is only likely to agree to a contract if it meets their own policies stating what rights the contract should confer and what liberties it should not take away. Our second example concerns European Union (EU) law. In the EU, member states' governments enact legislation to meet governance aims. When the EU council wishes to coordinate legislation across the union, an EU directive is issued. Directives constrain and direct member states to implement regulations that meet cross–national aims. For example, retaining communications data for EU–wide criminal investigations [71]. In turn, national legislation and EU directives are also required to confer rights and uphold liberties specified in the EU Charter of Fundamental Rights [72]. Our third example concerns the rules that make enacting and changing regulations possible. An organisation can only enact a legally valid and binding contract if the rules and processes that state how and when contractual regulations are enacted are followed [117, p.96]. The EU council is only able to legislate directives according to the rules that give it rule–making powers [73, Art. 288]. The United States government has the power to enact laws by following law making procedures, but laws cannot be created that apply to the past [221, Art. 1 Sec. 9 Cl. 3]. In other words, governments, organisations, contract writers and regulatory bodies are constrained in their regulatory activities.

These examples are all about institutions and the constraints placed on institution designers. Institutions are sets of rules and regulations, such as national legislation and contracts [27, 208]. Institution designers are governments, contract writers, organisations and other regulatory bodies. Institutions regulate, organize and guide individuals' behaviour in a society towards collaboratively meeting societal aims [186]. Societal participants are *autonomous agents*, such as people or software agents, which are liable to act in their own self interests. Hence, institutions pair regulations with penalties to ensure it is in an individual's own self–interest to comply for the greater good of the governed society [15, 41][44, p.80]. Institution designers craft institutions to regulate agents towards societal aims, such as people contractually participating in an organisation collecting and providing data, or citizens participating in a society whilst respecting one another's rights. By regulating rather than regimenting (forcing certain) agent behaviour, institution designers can achieve societal aims whilst preserving agents' autonomy.

Institution designers are also autonomous agents and hence are liable to act in their own self–interests. Consequently, regulations are placed on institution designers to reign their behaviour in. The first two examples showed how institution designers are governed in the institutions they design. Some institutions are designed to govern societies, such as national legislation. But, such institutional designs are in turn governed by other institutions, such as EU directives and human rights charters. The third example shows how institution designers are regulated in their ability to enact institutional changes by establishing new institutions or change existing institutions. By enactment, we mean in the sense of passing of a new law or a change to legislation, making a contract legally binding or more generally

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

the social action of making an institutional change legally valid and imposed on agents. For example, enacting contracts or changing existing institutional rules to meet new aims. Governing regulatory change is defined by rules that stipulate when institutional changes are enacted. Hence, institution designers autonomously design and enact institutions, but institution designers are also governed in the institutions they *design* and the institutional *enactment* process.

In today's increasingly technological world, operationalising institutions by applying rules and regulations is not just left up to opaque human reasoning. Instead, the fields of *Normative Multi–Agent Systems* and *AI and Law* are concerned with automated reasoning for institutions by contributing formal frameworks that interpret and apply institutions (we discuss background on implemented systems later in Chapter 2, and a literature survey is provided by [12]). Formalisation, in general, takes informal reasoning, that is ambiguous and unclear, and exposes it in plain sight with precise mathematically rigorous definitions.

Yet, thus far the reasoning involved in the governance placed on institution designers has not been formalised. We can see a common thread is shared by our examples, in that institution designers are constrained loosely in how the institutions they enact *should* be designed and strongly in how and when they *can* enact institutional changes. Yet, although we can clearly see such constraints exist in their written form, we cannot physically see precisely how the machinery that applies and tightens those constraints around the institutional designers works. It is consequently hard to grasp how exactly institution designers need to operate under institutional design and enactment constraints. This dissertation addresses this knowledge gap, by contributing *a formal framework for analysing institutional design and enactment governance*. To understand more specifically why such formal reasoning is important, let us first take a closer look at the concepts behind institutional design and enactment governance.

From the institutional *design* governance perspective, institution designs are governed by other institutions in what is called *multi–level governance* [121]. In this dissertation, multi–level governance is conceived as higher–level institutions designed to govern and guide the institution *designs*, enacted by autonomous institution designers, operating at lower governance levels. Multi–level governance facilitates institution designs in being coordinated, when viewed as being related in a wider multi–institution system (e.g. [244]). For example, at a national level a government enacts institutions to govern a nation. At a cross–national level institutions are designed to guide national governments in enacting institutions with coordinated regulations. Moreover, designers are guided in ensuring their designs do not take away rights and liberties, such as by human rights charters. In multi–level governance, institution designers have autonomy to design institutions according to their aims, but their institution designs are also subject to being governed and guided by higher–level institutions.

Multi–level governance creates the possibility for institution designs to be non–compliant. A non–compliant institution design is problematic. From the higher governance levels' perspective non–compliant institution designs are uncoordinated with other institutions governing separate jurisdictions, do not uphold rights and/or do take away liberties. From an institution designer's perspective, they are liable to face punishment for non–compliant design. When societal members act in a non–compliant way they are liable to being penalised in order to guide society towards compliance[11, p. 279]. Hence, when an institution designer designs a non–compliant institution, they are liable to being fined or having institution designs annulled [210] by judiciaries. Non–compliance in multi–level governance should be detected and avoided before institution enactment by an institution designer in order to avoid punishment, and detected after institution enactment by a judiciary to issue punishment and thus incentivise institution designers in enacting compliant institutions.

From the institutional change *enactment* governance perspective, institutional rules define the legislative actions that constitute a valid institutional change enactment conditional on the social context [20]. In a simple case, a legislature voting by majority on an institution enactment change suffices. In other cases, there are further constraints on valid institution change enactment, such as there being no valid way to change an institution in the past (retroactively) [221, Art. 1 Sec. 9 Cl. 3] "No Bill of Attainder or ex post facto Law shall be passed". In general, there are rules that state how and when rules are changed. According to the influential legal philosopher Hart, these are *secondary* institutional rules and they create the *possibility* for governments to change legislation and citizens to create contracts [117, p.81]. When an institution designer designs or changes an institution outside of the secondary rules the "enacted" changes are *invalid*. For example, enactment does not occur merely by physically writing institutional rules without following the necessary voting procedure set out by secondary rules. From an institution's perspective, invalid enactments or changes do not take place. Hence, the system or society the institution governs should ignore such invalid enactments in order to have the correct and shared view of an institution's rules.

Formal reasoning is important for the governance of institution design and enactment. Specifically, for precision and automation:

> **Precision**. In general, formalisation removes ambiguity. Without ambiguity, rigorous interrogation of the underlying reasoning is possible. Hence, reasoning flaws can be found and fixed, which can have real positive and negative consequences for agents being rewarded or punished. Moreover, formalisation communicates in clear mathematical terms the reasoning involved. Hence, given an institution and the facts of a case, agents can independently come to the same conclusions by applying identical reasoning.
>
> From a multi–level governance perspective, an institution designer can predict whether a judiciary would find the institution design non–compliant. *With one caveat, the designer and judiciary must have the same legal rule base, including any unwritten rules such as conceptual interpretations*. Prediction benefits an institution designer who can choose to not enact a non–compliant institution in order to avoid punishment.
>
> In the case of governing the institutional change enactment process with secondary rules, all agents governed by an institution are able to come to the same understanding over the changes made to the rules they are governed by. A common understanding benefits two agents in neither having a different understanding of the regulations in place nor misunderstanding what they ought to do according to the institutions governing them.
>
> **Automation**. Formalisation removes ambiguity, which is a necessary pre–requisite

for automation since it means, potentially, the formal reasoning can be implemented in a computational language as a program for a computer to execute.

From the multi–level governance perspective, automation lowers compliance checking costs. A judiciary does not face the dilemma over either arduously determining if an institutional design is compliant or forego judgement and face the possibility that institution designs are allowed to be uncoordinated or take away agents' rights and liberties without punishment. Automation also lowers compliance checking costs for institution designers and helps rectify non–compliance. Fixing non–compliance is not necessarily easy. There can be multiple possible explanations for why an institution is non–compliant in there there is often a space of many compliant institution designs. For example, EU member states are at liberty to comply with an EU directive in many ways [76, p.5]. Yet, some compliant designs will meet an institution designers' objectives more than others. By automating compliance checking, searching for explanations for non–compliance that can be used to rectify the underlying causes is also automatable in a way that meets an institution designer's aims the most.

It is non–trivial to determine how an institution is changed according to secondary rules governing rule change. For example, a new rule might be enacted stating rules can be changed by majority vote. Enacting such a rule affects which future rule changes can be enacted. More generally, without wishing to delve into the temporal details at this point, rule changes can also be applied to the past, present or future and have many complex interacting affects with other rule changes. Automation takes over the cognitively difficult task for a human of determining when rule change enactments are legally valid.

In order to address the need for precision and automation, institutional design and enactment governance should be formalised, but there is a lack of formal work in this area. This dissertation addresses this problem by proposing the PARAGon[2] formal framework for **P**ractical **A**utomated **R**easoning for **A**ssessing **Go**ver**n**ance of institution design and enactment. The formalism comprises precise definitions for compliance in multi–level governance, a mechanism to determine explanations and rectifications for non–compliance, and precise definitions for institution enactment validity according to secondary rules.

Moreover, this dissertation adopts a practical approach in two senses. Firstly, it provides definitions that are either coupled with a corresponding computational implementation or are defined in such a way as to make it obvious how the reasoning can be implemented. Thus, the formalisation provides *automated* reasoning. Secondly, the conceptualisations are aimed at capturing a number of real–world case studies in a realistic setting that includes factors such as time and change. The framework allows an institution designer to specify the institutions in a formal grammar and determine using the reasoning whether the design is compliant in multi–level governance and rectify any non–compliance. Moreover, institution designers and agents governed by institutions alike can apply the framework to determine which rule changes are validly enacted and when. The framework users need

[2]Just as an institution defines ideality, a paragon is an example of a person or thing regarded as a perfect example of such an ideal [191]

not understand the underlying mechanics, in terms of how it is *decided* what the regulatory effects of an institution are. Rather, users only need to understand what the concept of an institution being used by the system is and correctly provide as input the various institutions that are governing/being governed and the actions of various agents that are occurring. Consequently, the reasoning burden can be delegated to a computer to mechanically determine institution design compliance, rectifications for non–compliance and valid rule change enactment.

This chapter proceeds to introduce the research questions in section 1.1. We describe how this dissertation fits into the SHINE project in section 1.3. Then, the research approach is described in section 1.2. Finally, the outline of this dissertation is given in section 1.4.

1.1. RESEARCH QUESTIONS

As we discussed, there is a lack of formal and practical reasoning for governing institution design and enactment that needs addressing. This leads to the main research question:

> How can institutional design and enactment governance be supported with formal reasoning?

The main research question is broken down into five sub–questions.

The idea is to contribute formal reasoning to support human stakeholders in understanding the constraints placed on institution designers. Stakeholders need a way to represent institutions and their governance relationships. This leads to our first research question.

> **Sub–research question 1:** What is a suitable representation to specify institutional design and enactment governance?

It is important for the representation to be natural, by which we mean with a clear correspondence to how written and verbal institutions are represented, for two reasons. Firstly, from the precision perspective we are interested in formalising institutional governance to make it clear what we mean by the relevant concepts. The idea being, the concepts are exposed and can be interrogated, and discussed and debated for 'correctness'. A natural representation supports this aim by demonstrating to what extent the formalised reasoning is doing all the work of coming to correct conclusions and to what extent it is actually highly dependent on the way an institution is written. The latter case is far from desirable. For example, if we have to represent an institution in a very convoluted or procedural way to get the correct reasoning results, it is reasonable to assume that the informal reasoning has hardly been formalised at all. From the automation perspective, a 'user' [3] needs to face as little burden as possible, meaning they should be able to specify an institution close to how they would a written law without thinking about how a computer might interpret it. To give a contrasting example, we could simply provide the user with an abstract computational machine (e.g. a Turing machine [225]) and give the user all the flexibility to specify institutions in whichever way they want as procedural code. However, by giving users an

[3] a user of the PARAGon framework can be a human agent interested in reasoning about institutional design and change enactment governance, such as to avoid punishment for non–compliant design. A user can also be a researcher wishing to understand the precise meaning of the concepts involved.

'unnatural' language to specify institutions, we have contributed nothing to automating institutional governance reasoning. Hence, a natural representation is central to being able to critique the PARAGon framework, and the framework itself providing the benefits of formalisation for institutional design and enactment governance.

The PARAGon framework answers this question by providing different ways to represent institutions depending on whether we are interested in the governance of institution design or change enactment. The first step in answering the question is to gather background knowledge on the kinds of institutional rules and regulations we wish to represent and to what extent there are already suitable grammars to represent those concepts in Chapter 2. Then, in Chapter 3 and 4 we provide a representation for multi–level governance and apply it to European Union laws. A simpler representation for governing institutional designs is provided in Chapter 5 and applied to a crowdsourced mobile sensing scenario inspired by the SHINE project. In Chapter 6 we provide a further representation language, this time focussed on secondary rules used to govern institutional change enactments. Once we have a way to represent institutions, we need a way to reason about them, leading to our next four sub–research questions.

The first aspect we look at is *compliance* of institution designs in multi–level governance. Although it is expected that institution designers enact compliant institutions, due to designers' autonomy it is not guaranteed. Compliance of institution designs should be evaluated to enable institution designers to understand whether their designs are compliant before enactment, and higher governance levels' judiciaries in punishing institution designers for offending institution designs. Compliance should be assessed in a precise and consistent way, such that the same notion of compliance is shared by institution designers and judiciaries. Hence, compliance requires formalisation. Since this dissertation takes a practical approach, addressing real–world needs, compliance should be formalised to capture its real–world informal notion. This leads to our next sub–research question:

> **Sub–research question 2:** How can we formalise compliance in multi–level governance?

In general, compliance involves adhering to regulatory requirements [172]. In multi–level governance, compliance represents whether from the perspective of institutions designed at higher governance levels (e.g. [76, p.5]), lower–level institutions regulate in a 'good' way to meet particular higher–level governance aims. There is a difference between societal compliance and institution design compliance. Societal compliance focuses on agent behaviour, which is already widely formalised [12], such as agents being compliant if they do not murder one another according to institutions that prohibit murder. On the other hand, institution design compliance focuses on whether the effects of regulations are good, such as by being coordinated with other institutions' regulatory effects (e.g. controlling carbon emissions through high–level international institution design coordination [188]), conferring human rights (e.g. prohibiting murder) and respecting liberties. So, compliant institution design is formalised by defining when the regulatory effects at lower governance levels are compliant with the regulations at higher governance levels.

The formal notion of compliance in multi–level governance must, however, also take into account *abstraction*. In multi–level governance, there are conceptual differences between institutions operating at different governance levels. Institutions embody institutional concepts defined with context–sensitive conceptualising rules [214]. For example, a rule stating that a soldier killing another person who is not a member of an opposing army is murder. Such conceptual rules serve an important role, since they allow regulations to be succinctly defined over high–level concepts [32]. For example, a regulation prohibiting murder as opposed to a regulation prohibiting a soldier from doing such and such. Separate institutions can potentially define the same concepts differently or define different concepts altogether. Moreover, since institutions define regulations over separate concepts, the regulations themselves have different, context–sensitive, meanings. In multi–level governance, we can see higher governance levels in fact use more abstract concepts. For example national legislation might concretely require communications data to be stored (e.g. [226]). Yet, human rights charters describe abstract notions such as the right to a private and family life [48]. Therefore, compliance should be defined in terms of concrete lower–level governance regulations being compliant with abstract higher–level governance regulations. Moreover, compliance should account for the meanings of the regulations potentially being entirely different depending on the social context.

The PARAGon framework provides a formal definition of compliance in multi–level governance in Chapter 3, reconciling the different, context–sensitive, regulation meanings from a conceptually concrete lower–level institution compared to a conceptually abstract higher–level institution. By defining compliance in multi–level governance, we are able to unambiguously determine compliance with mathematical rigour and regularised predictable results for a real world setting.

Having a formal definition of compliance does not necessarily reveal how to practically reason about compliance in a computational way. This leads us to the third sub–research question:

> **Sub–research question 3:** How can institutional design compliance in multi–level governance be computationally verified?

Computationally verifying institution design compliance in a practical way is desirable to reduce compliance checking burdens placed on the various actors involved in multi–level governance. Whilst each institution designer at each level of governance must provide the institutions being governed and doing the governing as input, there should be no burden on those designers to implement or understand the mechanics of compliance checking. As Sergot exemplifies using the law [216], institutions are sets of declarative rules. Whilst rules can influence each other, and rules can define procedures which should be followed, the ordering of the rules themselves should not impact their meaning. What we can take from this is that institutions are not algorithmic and any algorithmic reasoning applied to an institution is not visible in an institutional specification. Consequently, institution designers are not particularly concerned with the reasoning mechanisms used to operationalise institutions. So, automated compliance checking is needed to ensure institution designers do not need to take extra effort in applying the formalised reasoning. That is, by providing a computational framework which hides the mechanics (i.e. algorithms or semantics) for compliance checking. So, what we aim for when answering this research question, is a computational framework that takes a natural representation of institutions as input and determines if an institution design is compliant in a sound and complete way (i.e. equivalent) with respect to our formal notion of compliance.

The PARAGon framework contributes computational compliance checking in multi–level governance in Chapter 5, which is sound and complete with respect to our formal notion of compliance. This supports judiciaries in determining penalties issued on institution designers for non–compliant designs. Moreover, it supports institution designers in avoiding penalties by indicating a non–compliant institution design should perhaps not be enacted.

For an institution designer it might be simplistic to state that they can simply avoid punishment by not enacting a non–compliant institution design. Firstly, in the case that an institution designer is actually obliged to enact an institution in order to implement the imperatives issued by a higher–level institution, as is required in complying with EU Directives and other cross–national institutions. Secondly, in the case that an institution designer prefers to enact an adjusted and compliant institution rather than not enacting an institution at all. This leads us to our fourth sub–research question:

> **Sub–research question 4:** How can non–compliant institution designs be explained in order to rectify non–compliance according to the institution designer's objectives?

Institutions are designed with a purpose, to guide societies towards an ideal envisaged by the institution designer or guide and coordinate other institution designs towards an ideal. We are interested in resolving non–compliant designs. Mitchell [174], provides a number of reasons for (non–)compliance in general, such as the non–compliance if agentive actions in society. Pertinent to institution design, non–compliance can occur due to preference, in our case because institution designers prefer a non–compliant institution design due to its positive governance effects, even in the face of penalties for non–compliance. Non–compliance can also occur due to incapacity, in our case simply because the institution designers do not know how to design a compliant institution.

These reasons for non–compliance should be taken into account when rectifying non–compliant institution designs. From the non–compliance due to designer preference perspective, any non–compliance resolution should balance the requirement to successfully adjust the design to be compliant against the objectives of the institution designer. Assuming the non–compliant design was crafted to achieve particular governance aims, resolving non–compliance must remain as closely as possible to the original design's regulatory effects. From the non–compliance due to designer incapacity perspective, non–compliance resolution should take into account the fact that incapacity can be due to institutions being complicated, comprising many interrelated rules and regulations [97, p.14] [171, p.2]. This means it can be non–obvious to an institution designer which rules and their interactions are causing non–compliance.

Assuming an institution designer wishes to resolve non–compliance in a way they understand, non–compliance resolution should seek the simplest and most general explanations for non–compliance, to support the institution designer in understanding and remedying the underlying problems. For example, the following case of non–compliance could have multiple explanations, which resolve the non–compliance – a lower governance level institution is governed by a higher–level institution in only obliging agents that are adults in providing personal information, but the lower–level institution obliges both children and adults to provide personal information. One possible explanation for non–compliance is that it is due to there existing a rule that obliges people to provide personal data and that rule should be removed to ensure compliance. A second explanation for non–compliance that gives less drastic institution re–design advice and is, is that the rule obliging people to provide personal information is too general and it should be modified to only applying to adults. In summary, non–compliance resolution should adhere to an institution designer's own objectives and recommend the simplest explanations and rectifications for non–compliance.

The PARAGon framework answers this question in Chapter 5 with a computational non–compliance resolution mechanism.

The previous research questions address institution design governance, our fifth and last research question addresses institution change enactment governance:

> **Sub–research question 5:** How can we formally define when legally valid institutional change enactments occur?

Institutional change enactment is constrained in its validity according to secondary legal rules. We need a formal definition of when, for a set of secondary legal rules, particular physical behaviour such as 'the thing we call' voting on a rule change causes the social action of rules actually being changed. Such secondary rules create the possibility to take the social action of changing rules [20]. These rules making rule change possible are also themselves changeable [220]. Formalising the legal validity of rule change enactment must take into account the fact that changing rules can affect further rule change enactments. Since we are dealing with real–world case studies the temporal aspect of secondary rules is a factor. It is possible, for example, for secondary rules to make past rule change enactments possible. Institution designers have enacted past rule changes in order to 'undo' the consequences of 'bad' institution design decisions [192]. The implication is that, since changing rules can affect rule change enactments, changing rules in the past can affect rule changes at various other points in time, including in the more recent past, the present or the future. So, what this research question requires is that we formalise rule change enactment validity whilst taking into account complex temporal interdependencies between secondary rules and rule changes.

The PARAGon framework answers this research question in Chapter 6 with a formal framework for determining when rule changes count–as legal rule change enactments. In answering this question, special attention is paid to formalising in such a way that demonstrates the computational mechanism to decide whether a rule enactment is valid. Moreover, an account of the temporal aspects is given, demonstrated against a number of real and imagined case studies.

By addressing these research questions the following argument is made:

> This dissertation formalises governing governance by giving a rigorous mathematical definition for institutions being governed in how they *should* be designed and how institutional change enactments *can* be made.

Now we proceed to the approach we take to answering the research questions.

1.2. RESEARCH APPROACH

This dissertation's result is the PARAGon framework for formally reasoning about governing institution design and enacting changes. A research approach was followed to develop the PARAGon framework and its constituent parts. The approach is to start with a literature review and then *per research question* gather case studies, develop a formal framework and apply model checking to assess the framework against the case studies. Each step is described as follows.

Literature review: this step achieves three aims. This research started out in the broad area of institutional and normative reasoning. The first aim was to understand informal definitions of the main concepts involved, namely institutions and norms, governance and closely related concepts. The second aim was to identify an area of research which was both useful to the SHINE project of governing large–scale sensor systems (and systems of systems) and had not been looked at previously from an Artificial Intelligence perspective. Namely, the governance of institution design and enacting changes. The third aim was to identify a single underlying formalism providing more primitive concepts (institutions and regulations) on which the PARAGon framework can use as foundations.

Framework development cycle: The next step was, per research question, to iterate the following three sub–steps. The iterative cycle terminates when applying the PARAGon framework to case studies results in a natural representation and judgements that correspond to real–world judgements or what we would intuitively expect. Although, 'natural representation' and 'intuitive expectations' are subjective, the representation and reasoning are made precise so that they can be argued for and against.

- **Gather case studies:** Each research question addressed a real–world governmental process to formalise and automate. To answer the research questions, the first step was to gather or synthesise relevant case studies against which the framework was developed, to ensure it was grounded in a realistic setting.
- **Formalisation:** The next step was to take the relevant case studies and understand the specific reasoning involved in order to come to the intuitive outcome for each case study. Then, to provide a more general account of the reasoning through *formalisation*, defining a formal syntax and semantics, potentially coupled with a corresponding computational mechanism. Following Hansson's argument [114], the purpose of formalisation is not to produce an empirically supported theory, but rather to precisely define previously informal concepts (in our case secondary rules, governance of institution design and so on).
- **Model checking case studies:** given a formal language comprising a representation and semantics, it is possible to formally represent the case studies and construct *models* for those case studies. By a model, we mean a logical model, which for a logical theory is a structure that *satisfies* the theory (if such a structure exists). By constructing a model, it is possible to check various properties of the theory. In our case, the logical theory is an institutional specification and

a series of events. The events either occur hypothetically as a part of an offline institution design check (e.g. checking how the institution behaves when an agent decides to collect data) or events which occur in reality (e.g. agents voting on rule enactments). The properties being checked are whether an institution design is compliant, ways to rectify non–compliance and which rule changes have validly been enacted and when. During this step, it was often revealed that there were counter–intuitive results for the developed semantics and given case studies, consequently further refinement of the formalisation was required and the previous steps repeated.

Putting it all together: The final step in the research approach was to apply the results and reflect on open questions. This step takes the application further to show its practical relevance by implementing part of the PARAGon framework in a prototype system described in Chapter 7. Each component of the PARAGon framework has its own implications. In this step those implications are compared and synthesised into an overall conclusion and set of unanswered questions in the final chapter, Chapter 8.

1.3. THE SHINE PROJECT

This research was initiated and supported by the SHINE project of TU Delft. SHINE was a large interdisciplinary research project. It aimed to develop techniques for acquiring and coordinating large numbers of heterogeneous data resources (e.g. cellphone sensors, radars and people). The idea was to use these sensors to gather a wide range of detailed environmental data (e.g. rainfall and pollution levels). In so doing, various stakeholders (e.g. citizens, municipalities) can gain answers to questions pertaining the environment (e.g. 'how do I get from A to B whilst avoiding flooding?'). SHINE looked at the problem from many different angles, such as algorithms for configuring sensors, user modelling, governing and coordinating resources, and visualising the acquired data to help answer questions.

This dissertation contributes techniques for formalising governance, with a focus on formalising governance which is, arguably, particularly suitable for governance and coordination of 'SHINE–like' sensor systems of various types. For example, governing and coordinating systems of cellphones gathering geospatial audio data to determine crowding levels or systems of weather radars used to determine rainfall levels. The formalisation is applied to a mixture of governance case studies, from national and international data and human rights laws to imagined SHINE sensor system regulations. The idea is to formalise compliance in multi–level governance and institutional enactment validity as found in the social world to support automated 'SHINE sensor system' governance.

Multi–level governance is relevant to SHINE according to the argument that it is a necessary governance architecture for governing heterogeneous sociotechnical systems (e.g. what is also called polycentric governance [197, 198]). We will exemplify why using examples concerning SHINE systems. On the one hand, a homogeneous sensor system comprising users crowdsourced into donating their cellphone sensors can, arguably, be governed with a single set of related regulations. For example, regulations defining a communal economy with a single incentive for users to contribute data (i.e. receiving data from a common pool in return). On the other hand, SHINE aims to form heterogeneous sensor systems which, arguably, are unsuitable for a 'one–size–fits–all' set of regulations. For example, a system comprising weather radars contributed by organisations could operate best as a market economy, where organisations are incentivised to join the system in order to trade data. In this case, the regulations for a crowdsourcing sensor system operating as a communal economy are entirely inappropriate. One possibility is to view each set of sensors as a separate sensor *sub–system* and write regulations to govern those sub–systems separately to form a super system of sub–systems. However, this places a burden on the SHINE–system institution designer in writing appropriate regulations for each sensor sub–system.

Multi–level governance offers an architecture to move the burden of writing specific regulations from the SHINE–system institution designer to the sensor sub–system stakeholders. The architecture proposes a solution, in the same vein as existing proposals of poly–centric governance for smart–grids comprising heterogeneous energy sub–systems [57]. The idea being, that the stakeholder wishing to form a heterogeneous SHINE sensor system crafts an institution at a second governance level which governs the design of institutions governing the separate sub–systems. The 'SHINE institution' is a thin governance layer, comprising abstract regulations requiring that sub–sensor–systems are governed by institutions which regulate resources towards collecting useful environmental data and punishing sensor owners for contributing erroneous data. Appealing to the principle of subsidiarity, what can be done at the local level should be left up to the local level, the SHINE institution would give space for the sensor sub–system stakeholders to determine which data is collected and what the incentives are (e.g. a market economy or a communal economy). In return, sensor sub–system stakeholders can design institutions to govern those sub–systems in order to join the SHINE super–system and gain data from other sub–systems in return. The PARAGon framework helps to operationalise multi–level governance for forming SHINE systems governed by a higher–level SHINE institution, by automating compliance checking and non–compliance rectification.

The automated reasoning for compliance checking is also applicable to the SHINE project's aim of crowdsourcing the existing sensors people already own and their time. For example, crowdsourcing people in donating their cellphone audio sensors in order to detect geo–spatial crowding, or crowdsourcing people into taking photographs of the sky when requested to determine pollution levels. This type of sensing is dubbed by the SHINE project as 'request driven social sensing' and a key idea behind it is that people are offered contracts for use of their devices and time. Since people are ideally offered many contracts to address many data needs as and when they arise, it is important cellphone users can automatically accept or reject contracts on the basis of policies they define stating how, when and for whom their cellphone sensors can be used. In this situation, a policy stating sensor usage governs offered contracts. The automated multi–level governance reasoning can be used to automate such contract rejection and is implemented in a prototype simulated crowdsourced mobile sensing system described in Chapter 7.

Secondary rules governing institutional enactment are relevant to SHINE from the perspective that they provide sensor system stakeholders a flexible and automated way to govern sensor systems' regulatory change. Flexibility is meant in two senses. Firstly, stakeholders are able to make how regulatory change legally operates flexible. This is important since the sensing aims or dynamics of the system are liable to changing over time and hence regulatory changes need to be enacted to meet stakeholders new aims or system participant's changing behaviour. Secondly, the way in which the regulatory change enactment process is defined is flexible. For example, one sensor system can define regulatory change as requiring sensor system participant's democratic vote, this might be suitable for a system of crowdsourced cellphone users that donate their cellphone sensors to the system partly due to having a say in how that system is run. In another case, where designing regulations requires highly–technical knowledge of sensors' operation, a sensor system's regulatory change may be defined on the basis of elected technocrats coming to an agreement. Hence, secondary rules allow a diverse range of governance and regulatory change enactment styles to be defined for different sensor systems in a way that enables those system's regulations to adapt, as deemed appropriate, to new aims and needs.

The PARAGon framework supports realising these benefits by contributing formalisation to automatically determine how and when regulatory changes are legally enacted. For example, such as due to a vote to move a system from a market economy where data is traded to a communal economy where data is contributed to a common pool and shared. Moreover, formalisation means the way in which changes are enacted is automatically changed according to legal rule changes, such as moving from a directed democracy to an elected technocracy. Automation means system stakeholders can operate a flexible governance system at lower cost and therefore makes systems where flexible governance is necessary, such as for diverse SHINE sub–systems, more viable.

1.4. DISSERTATION OUTLINE

The dissertation outline is depicted in figure 1.1. This dissertation is broken up into the introduction (Chapter 1) and the background (Chapter 2), then the main contributions of formal reasoning for governing institution design and change enactment (chapters 3 to 6), and finally an illustration of the framework's application (Chapter 7) and the conclusions (Chapter 8). The main technical contributions begin by looking at *soft constraints*. These are constraints that create the possibility for (non–)compliance and, respecting agents' autonomy, cannot be guaranteed to be complied with. In our case, the soft constraints specify how an institution *should* be designed (chapters 3 and 4). We subsequently contribute a mechanism for revising an institution design to be compliant with such soft constraints (Chapter 5). Then, we move to reasoning about governance in the form of *hard institutional constraints*. These are constraints that are not violatable. In our case, the hard constraints are secondary institutional rules defining when institutional change enactment is *possible* (Chapter 6).

The following chapters are described in more detail:

Chapter 2 provides background starting with an overview of the informal concepts we wish to formalise and reason about. Then, proceeding to analyse the existing knowledge and its gaps in formalising governance of institution design and change enactment. Next, suitable frameworks that provide preliminary formalisation on which to build the PARAGon framework are compared. Finally, suitable foundational formalisations are selected on which this dissertation builds.

Chapter 3 contributes a formalisation of compliance in multi–level governance, providing a way to precisely determine compliant institution designs in a predictable way.

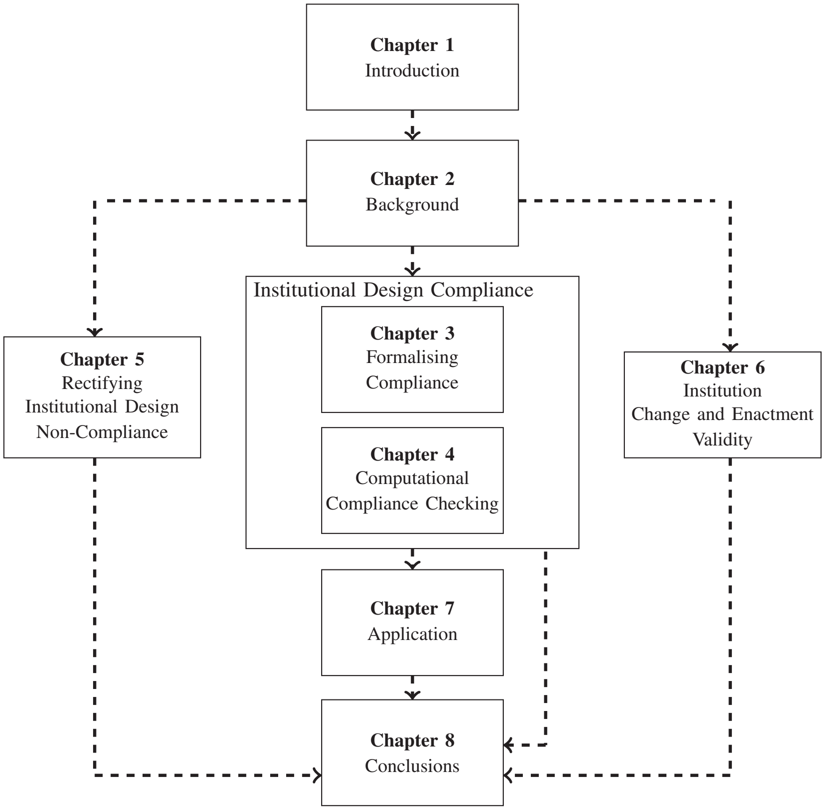

Figure 1.1: An overview of the dissertation with suggested reading orders.

Chapter 4 contributes the computational means to go about detecting compliance in multi–level governance, furthermore this chapter contributes a computational framework which is sound and complete with respect to the more theoretical formalism in the previous chapter.

Chapter 5 contributes a computational mechanism to automatically rectify non–compliant institution designs.

Chapter 6 contributes a practical formalisation of institution design validity, looking at when rule changes count–as legal rule change enactments in the face of secondary rules governing the rule change enactment process.

Chapter 7 describes a prototype application of compliance checking institution designs (contracts in this case) to forming networks of crowdsourced users and their cellphones in order to collect and aggregate weather data whilst giving users autonomy over how, when and for whom their devices are used.

Chapter 8 concludes with discussion on the contributions, implications and directions for future work.

1.5. LIST OF PUBLICATIONS

The chapters in this dissertation are based on the following publications:

King, T. C., Li, T., De Vos, M., Dignum, V., Jonker, C. M., Padget, J., & van Riemsdijk, M. B. (2016, July). Automated Multi–level Governance Compliance Checking. *Journal of Autonomous Agents and Multiagent Systems (JAAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems. (**In Submission**)

King, T. C., Dignum, V., & Jonker, C. M. (2016). 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. (pp. 3 – 11). IOS Press. http://doi.org/10.3233/978–1–61499–672–9–3

King, T. C., Li, T., Vos, M. De, Jonker, C. M., Padget, J., & Riemsdijk, M. B. Van. (2016). Revising Institutions Governed by Institutions for Compliant Regulations. *Coordination, Organizations, Institutions, and Normes in Agent Systems XI: COIN 2015 International Workshops, COIN@ AAMAS, Istanbul, Turkey, May 4, 2015, COIN@ IJCAI, Buenos Aires, Argentina, July 26, 2015, Revised Selected Papers.*, 9628, 191 – 208.

King, T. C., Li, T., De Vos, M., Dignum, V., Jonker, C. M., Padget, J., & Riemsdijk, M. B. Van. (2015). A Framework for Institutions Governing Institutions. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)* (pp. 473–481). Istanbul, Turkey: International Foundation for Autonomous Agents and Multiagent Systems.

King, T. C., Liu, Q., Polevoy, G., Weerdt, M. de, Dignum, V., Riemsdijk, M. B. van, & Warnier, M. (2014). Request Driven Social Sensing (Demonstration). In A. Lomuscio, P. Scerri, A. Bazzan, & M. Huhns (Eds.), In *Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)* (pp. 1651 – 1652). Paris, France: International Foundation for Autonomous Agents and Multiagent Systems.

King, T. C., Riemsdijk, M. B. Van, Dignum, V., & Jonker, C. M. (2015). Supporting Request Acceptance with Use Policies. 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 (pp. 114 – 131). Springer.

2

Background

There is a lot of work which tries to do sophisticated statistical analysis. You know, Bayesian so on and so forth. Without any concern for the actual structure of language. As far as I'm aware that only achieves success in a very odd sense of success [...] I don't know anything like it in the history of Science. [196]

Noam Chomsky

This chapter makes the following contributions:

- An overview of the informal background knowledge from multi–agent systems, philosophy of language and law, and political science. The focus is on institutions, and institutional design and enactment governance.
- Formalisation requirements for the PARAGon framework.
- A comparison of existing formal approaches and the extent to which they fulfil the PARAGon framework's requirements.

This dissertation formalises reasoning for understanding institutional design and enactment governance. The formalisation supports institution designers in designing compliant institutions and enacting institutional changes in a valid way. The same formalisation allows judiciaries to determine if punishments should be imposed for non–compliant institution designs. Moreover, the formalisation supports agents in understanding when institutional rules are changed. Generally, the objective is to *formalise* the *informal* concepts involved in governing institution design and rule change enactment to the extent that the reasoning can be executed by a computer.

In this chapter, we give background on the research contributed by this dissertation. First, we situate the research within the wider field of autonomous agents and multi–agent systems in Section 2.1. Then, Section 2.2 gives a broad overview of institutions and norms from an informal point of view. Section 2.3 summarises the formal foundations required for this dissertation to build on and the new formal building blocks this dissertation develops in order to formalise institutional design and enactment governance. Section 2.4 outlines a number of relevant formalisations of institutions on which the contributions of this dissertation can be based. Special attention is paid to identifying to what extent existing approaches can be built on by this dissertation and where there are knowledge gaps. We summarise the knowledge gap in Section 2.6 and outline the approach we take.

2.1. AGENTS AND MULTI–AGENT SYSTEMS

The research presented in this dissertation overlaps with the field of Agents and Multi–agent systems. Examples include socio–technical systems and human societies. To situate this dissertation, it is important to give some background on the general concepts used by the field.

When we talk about an agent, we mean an entity which has the ability to act on the world, or in other words possesses agency [211]. Philosophy contributes many agent concepts, primarily dealing with human agents and action. Agency as the ability to act intentionally is a view taken by Bratman [31], Goldman [96] and Mele [169]. Under this definition agents are able to bring about changes of their own volition to realise their intentions. The view that agency is the ability to initiate action spontaneously regardless of intention is taken by Ginet [95], Lowe [161] and O'Connor [184], among others. A more complicated agent, called a 'person' is described by [84] as being able to act both on the world and also on their own internal mental states (e.g. realising second–order desires). Agents can be significantly simpler – Dennett [56] questions the conceptual utility in requiring mental attitudes as pre–requisites for agency. He undercuts arguments for requiring mental attitudes in the agent concept with Occam's razor – they should only be ascribed to agents if they help with predicting action. Agents need not be human or biological at all, Wooldridge [240, 241] offers a frequently used definition of an artificial autonomous agent in the MAS community. According to Wooldridge, an agent is software able to act in an environment to meet its own goals, often those given to it at design–time. Without attempting to resolve the differences between these stances the key point subscribed to by this dissertation and the wider field is that agents are able to act on their environment, by extension act in ways that affect one another and importantly act *autonomously* in ways that cannot be controlled directly.

A system comprising agents is called a multi–agent system (MAS). Such systems can be biological (e.g. a nation state comprising humans), artificial (e.g. a software system of agents controlling sensors) or mixed (e.g. a system comprising teams of humans and robots). An MAS is often formed with a goal in mind or an idea of how the agents should behave, from the perspective of the system's stakeholders (e.g. the members of a nation state or the designer of a software system). For example, a nation state may have the societal goal to minimise the number of car accidents. Another example is a system comprising software agents on people's cellphones for collecting and aggregating location and sound level data to determine large crowds of people [162]. To achieve the MAS' ideal and system goals as envisioned by the relevant stakeholders, the agents in an MAS need to act collectively. For example, driving on the same side of the road to avoid collisions or configuring sensors appropriately (e.g. ensuring a microphone is on) for the data being gathered. An MAS is a system comprising agents which is typically designed with an aim or ideal for agents to collectively achieve.

The autonomy of agents in an MAS makes it difficult to ensure agents collectively achieve the MAS' goals. Agents are liable to acting in their own self–interests (e.g. turning cellphone sensors off to conserve cellphone energy rather than contributing data) and if agents are left to their own devices their actions are liable to be uncoordinated (e.g. driving on different sides of the road). Mechanisms for system–wide agent control are required to ensure agents achieve the MAS' goals.

Different mechanisms to control and coordinate agents are available. We split these mechanisms into *bottom–up control*, *top–down governance* and *bottom–up compliance with top–down control*. Which approach is appropriate depends on the level of control a system stakeholder (e.g. software designer, national government) has over the agents and conversely, the level of agency an agent has.

- **Bottom–up control** – this approach relies on being able to control artificial agents directly. Specifically, by programming agents to coordinate their behaviour and collectively achieve the MAS' goals. Even basic software agents only capable of reacting to a single percept with no memory of past percepts and actions have been shown to exhibit complex emergent coordination such as decentralised gathering and collecting of objects [89]. For more complex cases agent programming languages have been proposed, often based on an architecture where reasoning is compartmentalised into mental constructs such as beliefs, desires and intentions (BDI). In these architectures agents can communicate and therefore be programmed to coordinate their behaviour (e.g. [52, 55, 119, 202, 205]), in the coo–BDI agent architecture for BDI (Belief, Desires, Intentions) agents' cooperation and coordination is supported with specific constructs for sharing plans [8]. Overall, the *bottom–up control* approach works by coordinating an MAS from the *bottom* level comprising the agents and their internal reasoning by regimenting the behaviour of each individual agent, such as by programming their behaviour in being coordinated and collaborative.

- **Top–down governance** – this approach deals with cases where agents cannot be controlled directly. For example, when the agents are biological (e.g. humans). Since such agents cannot be controlled directly they are liable to act in their own self–interests. A common example is the tragedy of the commons [115], where given no constraints it is optimal for agents to act in their self–interests based on the premise that other agents will be doing the same. A top–down approach is also required when an *open artificial multi–agent system* [118] is designed comprising heterogeneous agents that are unknown *a–priori*. In an open artificial MAS agents are liable to changing over time and are contributed by various programmers such that they are not (easily) reprogrammable and hence should be guided in behaving in the right way rather than being directly controlled. In top–down governance regulations prescribe how agents should behave (e.g. agents should drive on the left) coupled with rewards/punishments to give agents reason to comply (e.g. fines). Whilst in this situation direct control of agents cannot be automated, the top–down governance can. For example, formal and/or automated reasoning for: monitoring agents' compliance [34, 38, 75, 79, 175]; determining which regulations apply and when and which agents are compliant and should be rewarded or punished [25–28, 40, 42, 43, 58, 63, 97–103, 103, 130–132, 159, 160]; synthesising institutions [176–178]; verifying institution designs meet certain correctness properties and do not enable agents to exploit regulations [148, 167]. In this approach societal coordination is realised by *top–down governance* in the sense that it operates at a level sitting above the entire system and governs agents rather than directly controls them.

- **Bottom–up compliance with top–down governance** – this approach lies in between the two previously mentioned approaches as a way to automate compliance with top–down governance. In this approach, norm–aware agents are programmed. A norm–aware agent could be an artificial agent contributed to an open MAS with top–down governance. Here, the agent can act appropriately to comply with the top–down governance with no assumptions that the regulations placed on the agent will be known *a–priori*. Research in this area has focussed on agent programming languages, semantics and reasoning for agents acting in a compliant or preferable way way with the regulations and organizations the agents might join (e.g. [6, 50, 52, 124, 128, 129, 152, 170, 218, 222]).

This dissertation provides reasoning for both top–down governance and bottom–up compliance with top–down governance. Top–down governance reasoning is provided for determining whether institutions are being designed in a bad way (i.e. the designs themselves are non–compliant in multi–level governance) or institutional changes are enacted without following procedure (i.e. secondary legal rules are not adhered to and thus institutional changes are invalid). Bottom–up compliance with top–down governance is provided for institution designers themselves, so that they can design institutions in a compliant way and validly enact institutional changes. In the former case, bottom–up compliance is supported with automated reasoning to find explanatory rectifications for non–compliant institutions. In the next sections we will give background on institutions and governing institutional design and enactment.

2.2. GOVERNING MULTI–AGENT SYSTEMS

In this section we discuss governance of multi–agent systems from a conceptual point of view. This dissertation builds on concepts of institutions, discussed in Section 2.2.1 and norms discussed in Section 2.2.2. This dissertation contributes new formal conceptualisations involved in institutional design and enactment governance. Background informal concepts for institution design and enactment governance are described in Section 2.2.3.

2.2.1. INSTITUTIONS

Institutions provide top–down governance of agents. A definition from economics is given by North [182], as a set of rules regulating behaviour enforced with rewards and punishments. For example, fining agents or simply deviating from certain rules being considered a societal taboo and therefore affecting trust. Ostrom [185] provides a similar definition, put best in her own words as “the prescriptions that humans use to organize all forms of repetitive and structured interaction”. In legal theory Ruiter [208] agrees with North and also analogously refers to institutions as a kind of “social agent”, realised as agents practising the institution with behaviour that complies with the institution’s rules. John Austin [14] viewed the law (legal institutions) as rules laid down by one individual who has power over another. Alternatively, Ruiter [208] also views contracts between agents as legal institutions. In legal positivism, legal rules’ validity depends on the source (i.e. whether legislated by a legislator) [88]. According to these definitions regulative rules are an essential part of an institution and when an institution is legislated, such as by a parliament, it is a *legal* institution.

In philosophy [173] an institution is viewed as comprising regulative rules but also rules establishing the language and concepts with which agents and a social reality are talked about. For example, a rule stating ‘I do’ establishes what we call ‘marriage’. Such rules come about through social acceptance. Concepts such as marriage established by an institution, according to Searle [213, 215], are institutional facts and come about only because an institution makes social actions (e.g. marrying) *possible*. These institutional facts describe a *social reality* ascribed by the institution, giving an institutional interpretation of the *brute* (actual) reality. Searle refers to the facts that are independent of an institutional interpretation as *brute facts*, referring to a ground truth. For example, an institutional fact might be ‘paper’ which refers to a brute fact ‘the thing we call paper’. Searle proposes constitutive institutional counts–as rules of the form “A counts–as B in a context C”, to which from brute facts (As) ascribe more abstract institutional facts (Bs) in a particular social context entailed by the social reality (Cs). For example, the thing we call paper counts–as paper. In turn institutional facts may count–as further institution facts. For example, an agent does not just possess paper with 10 euros written on it, various watermarks and an identification number. Rather, that piece of paper is collectively agreed as counting–as money in the context of a particular nation state and so the agent possesses money in the right context. Counts–as rules establish a concise social description of reality, which can feasibly be talked about at a more abstract, social level, than would otherwise be possible (e.g. 'money' is far more concise than a fine–grained description of everything that constitutes money).

Searle argues counts–as rules also ascribe special statuses and deontic powers to various concepts [215]. For example, a piece of paper that counts–as money gives an agent possessing it the special deontic power to purchase goods. But, only because there is a social institution ascribing a special status to the money. On the deontic dimension of institutions, Searle says that various obligations are entailed by an institutional fact having a certain status. For example, a police person has a socially recognised status where they are obliged to uphold the law. In fact, Searle states that counts–as rules ascribing institutional facts and assigning statuses to those facts are necessary for there to be a deontology – "*No language, no status functions. No status functions, no institutional deontology*" [215, p.14]. Searle states institutions contain constitutive counts–as rules, which create a social reality from brute facts, and counts–as rules are necessary for a deontology.

Ricciardi [206] views related constitutive rules (e.g. all the rules ascribing meaning and status to money) as *constituting* the institution they belong to (e.g. the 'money' institution). A classic example from Ricciardi is the rules of chess constituting the game of chess and the *institution* of chess. Moves can physically be made on a chess board that do not conform to chess' rules, but then the game of chess is not being played and the chess institution does not recognise those moves. Hence, an institution (e.g. chess) *is* its set of rules.

The key points in these definitions subscribed to by this dissertation are:

- Institutions can be created by law makers (legislation), between agents (contracts) or through social acceptance. This dissertation mostly focuses on legal case–studies, but we do not tie ourselves to any particular institution source.

- Institutions *are* a set of related counts–as rules, which ascribe a social meaning to brute facts by building an institutional reality comprising institutional facts (n.b. this differs from the more complicated notion of an institution as a social agent). Chapter 3, and Chapter 6 focus on formalising constitutive rules.

- Institutions are used to govern and guide agents towards an ideal with rules and regulations (a deontology), rewards and punishments. Chapters 3 to 5 focus on this regulatory aspect of institutions.

- Counts–as rules and a social reality are necessary for an institution to have a deontology.

In the next section we will describe in more detail the deontic aspect of institutions and its informal representation.

2.2.2. NORMS

In this section we discuss institutional regulations, called norms, further. Norms state what the social reality *should* be (e.g. an agent should pay for a train ticket before boarding a train). With a suitable mechanism to ensure norms are abided by, norms guide agents towards an ideal. Two mechanisms to ensure norms are abided by are discussed in the literature: regimentation and enforcement.

Regimentation [135] forces norm compliance on agents by controlling their behaviour directly. For example, by programming agents so they cannot board a train unless they have paid for a ticket or by constructing a barrier at the train station that does not let agents onto the train platform unless they have a ticket. One drawback of regimentation is that it assumes hard constraints can be implemented (e.g. an agent can be directly programmed or there are physical barriers that can be constructed). Another drawback is that regimentation does not acknowledge that there are exceptional circumstances under which norms should not be followed – for example when an agent needs to drive faster than the speed limit in the case of an emergency. Regimented norms act as hard constraints on agents' actions, which is not always possible or in the best interests of the agents and the system. We do not consider norm regimentation in this dissertation.

Enforcement [107] gives agents the option to comply with norms that act as soft constraints on agents' behaviour. To give reasons for agents to comply, secondary reward and punishment norms come into force if a primary norm is violated. For example, a norm stating an agent should pay a fine for boarding a train without a ticket. In turn, the secondary norms can have further enforcing norms. For example, a train guard that finds an agent who refuses to pay a fine should throw the non–compliant agent off the train. This dissertation considers norms that can be violated and are supported through enforcement.

Norms take two common forms in the literature. Firstly, an *evaluative* representation lacking an explicit deontic modality (obligation/permission/prohibition). Secondly, a *modal* representation where the deontic modality is explicit.

Evaluative norms are found in some formal work [9, 105]. In an evaluative representation norms are represented as counts–as rules denoting that "A counts–as being good/bad/-compliant/a violation in a context C" with no reference to obligation/permission/prohibition. To give an example based on the UK's data retention regulations [226] – a communications provider (e.g. telephonics company) retaining all of a person's communications meta–data (e.g. the time a phone call was made) counts–as compliance in the context that the communications provider is operating in the United Kingdom. An evaluative norm creates evaluative institutional facts in the social reality. For example, creating 'compliance' if meta–data is stored. The social reality does not represent which norms are in force (what should be done). If we want to know whether the UK's data retention regulations require meta–data to be stored then the query is "do the rules state storing meta–data counts–as compliance in a context that holds in the present social reality?". Determining which evaluative norms are in force requires assessing the constitutive rules for whether they ascribe a concept counting–as compliance/violation in a context that holds in the social reality.

Modal norms are counts–as statements ascribing explicit deontic (obligation/prohibition/permission) positions in the social reality. Specifically, as Searle proposed in [213, p. 63] "A counts–as undertaking an obligation to do B in context C". For example, communicating using a communications provider's services establishes an obligation for the communications provider to store your meta–data in the context that they are operating in the United Kingdom. A modal representation creates a deontic position in the social reality. For example, an obligation to store communications meta–data. The social reality explicitly represents what should be the case or should be done. If we want to know whether the UK's data retention regulations require meta–data to be stored then we can look at the social reality for whether storing meta–data obliged. Determining which modal norms are in force requires inspecting the built social reality for which deontic modal statements hold.

In this dissertation, one objective is to reason about norms that govern other norms. Expressing and determining that particular norms should (not) be imposed differs between modal and evaluative norms. There are two ways for a norm to express that it is *required that other norms do not require* storing communications metadata in the context that a user has not consented to their communications metadata being stored.

In the evaluative form one possible representation is to nest a counts–as rule within another counts–as rule. For example, "the rule (storing metadata counts–as compliance in a context C) counts–as a violation if context C is somehow compatible with the user not consenting)". There may be other evaluative representations, but it appears to be the simplest that fully captures a norm stating the requirement for other norms to not require a specific social fact. In a modal form an unconditional constitutive rule seems to express the same thing – "it is prohibited to oblige a user's metadata to be stored in the context that they have not consented".

The modal norm appears simpler and with little difference between prohibiting an obligation to store metadata and the modal norm obliging metadata to be stored. We would expect that determining compliance is more or less the same for a modal norm obliging storing metadata compared to a modal norm prohibiting obliging storing metadata. In comparison, the evaluative version seems far more complex and is not a simple generalisation of evaluative norms about events. This dissertation considers a modal norm representation for simplicity of reasoning about norms that regulate other norms.

Norms prescribe a social reality and are often used together with counts–as rules ascribing non–normative institutional facts. According to Sergot [216, 217], and Jones and Sergot [135], a legal system (what we call an institution) largely consists of rules that define concepts (what we call counts–as rules ascribing institutional facts). Secondary in occurrence to rules defining concepts are the institution's norms. Definitional rules are used for succinctness, according to Ross [207]. For example, we can define storing communications metadata or content data as counting–as storing personal data. If we want to describe a new concept relating to personal data then we can refer to personal data rather than everything that constitutes personal data. For example, disrespecting someone's privacy is storing their personal data without their consent, rather than storing their metadata or content data or ... without their consent. Such definitional rules, according to Grossi et al. [109] and Breuker et al. [32], allow norms to be issued more succinctly over abstract concepts. For example a norm stating "you should not disrespect someone's privacy" is more succinct than "you should not store content data without consent", "you should not store metadata without consent", etcetera. Institutions largely comprise concept defining rules, which ascribe an abstract social reality, over which norms are concisely defined.

The key points in these definitions subscribed to by this dissertation are:

- Norms can take an evaluative or a modal form. An evaluative form does not make explicit that there is an obligation/prohibition/permission in force, it simply states a social action will cause compliance or a violation. A modal form makes explicit that something is obliged/prohibited/permitted in the social reality. The modal form offers a simpler way to represent norms governing other norms and a straightforward way to evaluate such norms. Chapters 3 to 5 formalise modal norms.

- Two ways to ensure norms are abided by are found in the literature – regimentation and enforcement. Regimentation assumes agents can be directly controlled. Enforcement does not assume agents' actions can be controlled, instead it aims to guide agents' behaviour. This dissertation does not make assumptions that agents can be controlled directly. In chapters 3 to 5 we look at soft constraints which guide, rather than regiment, agents in their actions within an MAS and how they design institutions. However, in Chapter 6 we look at a kind of hard constraint in the form of rules stating which institutional rule changes are possible. These are not hard constraints in the traditional sense (e.g. physically preventing an agent to board a train or directly controlling an agent's reasoning to comply with norms). So, we call these hard *institutional* constraints.

- Norms are defined in terms of an abstract social reality created by counts–as rules. Counts–as rules and norms are closely tied, counts–as rules by building an abstract social reality provide a pragmatic means to author concise norms. The link between constitutive rules ascribing abstract concepts and norms defined over these abstract concepts is focussed on in Chapter 3 and Chapter 4. Counts–as rules that make the social action of changing rules possible are addressed in Chapter 6.

2.2.3. GOVERNING INSTITUTIONAL DESIGN AND ENACTMENT

In this section we discuss how institution designers, such as legislators, are governed. The need to govern institution designers is motivated by the fact that they are autonomous agents. Being autonomous agents, institution designers are liable to behaving in ways that are sub–ideal. For example, designing an institution that takes away agents' rights or enacting institutional rule changes without following relevant procedure, such as obtaining a majority vote from a parliament. Hence, we look at how institutional designers are governed in their institutional designs and institutional change enactments, which we describe further from an informal point of view in this section.

GOVERNING INSTITUTIONAL DESIGN: MULTI–LEVEL GOVERNANCE

The problem with agency is that agents are liable to acting undesirably and hence need institutional governance; the problem with institutional governance is that autonomous agents can design subjectively 'bad' institutions. For example, an institution can lack coordinated regulations with institutions governing other jurisdictions or place unacceptable limits on agents' rights. One solution is to override an institution designers' regulations. But this would violate their autonomy and the principle of subsidiarity – what can be legislated at the local level should be left up to the local level. Hence, institution designers need to be guided in designing subjectively good institutions. In MAS agents are guided towards good behaviour with institutions governing their actions. In order to guide institution designers, institutions are used to govern institution *designs*.

Multi–level governance is an umbrella term for governance styles where institution designs themselves are governed. We depict it abstractly in Figure 2.1. The term multi–level governance is taken from political science [121] (alternatively called multi–tier and polycentric governance). An early conceptualisation by Marks [166] views multi–level governance as negotiation between state actors in the European Union to achieve coordinated

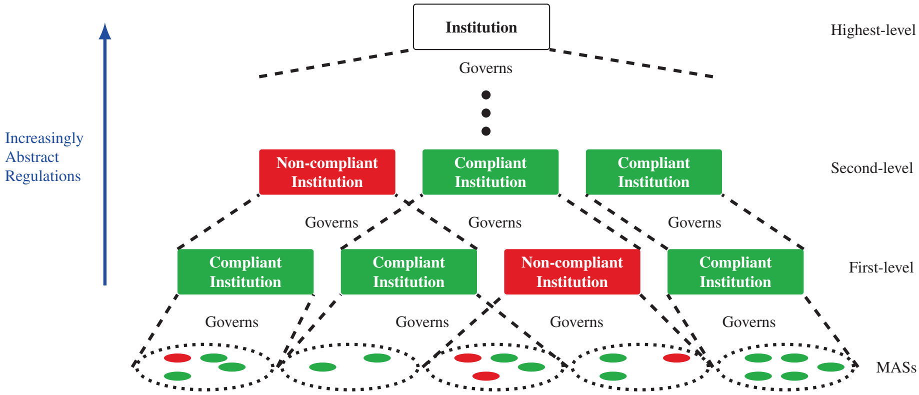

Figure 2.1: A high–level overview of multi–level governance. At the bottom level there exists MASs (e.g. nation states, sensor networks). The first level comprises institutions governing MASs (e.g. national legislation, or institutions governing artificial societies such as sensor networks). The second–level comprises institutions governing the first–level institutions (e.g. EU directives, cross–MAS objectives). The highest–level comprises institutions governing lower–level institutions (e.g. human rights charters and constitutions).

goals through regulatory change. The negotiated outcome being binding agreements/institutions between EU member states to design their institutions (national legislation) in such a way that they implement those binding agreements. In effect, the negotiations establish a hierarchy of institutions. The negotiated binding agreement being a higher–level institution governing the design of member states' legislation, which are lower level institutions. More recently, Börzel and Risse [30] conceive multi–level governance as existing at sub–national, national and supranational levels. Conceptually, elected institutional designers (i.e. local councils, national governments, EU governing bodies) create institutions to govern institutions at lower governance levels and comply with institutions enacted at higher governance levels. Hooghe and Marks [121] classify multi–level governance into two types. In the first type, an institution at a higher–level governs a jurisdiction of institutions nested at a level below. In this type lower–level institutions do not belong to any other jurisdiction and are only governed by a single institution at the level above. In the second type institutions at each level can belong to multiple jurisdictions and therefore be governed by multiple higher–level institutions. More generally, multi–level governance comprises institutions at higher governance levels (e.g. EU directives) governing the institution designs operating at lower governance levels (e.g. national legislation) which in turn might govern institution designs at even lower governance levels (e.g. sub–national regulations) etcetera.

Multi–level governance guides institution designers towards enacting ideal regulations. Eichener [65] showed multi–level governance in the European Union successfully resulted in cooperating legislation on occupational safety. By issuing higher–level institutions, an EU directive, the EU established workplace health and safety regulations across the EU. The health and safety regulations were coordinated in the sense that no country had more relaxed regulations allowing them to out–compete other member states on labour costs. Hence coordinated regulations ensured no 'race to the bottom' was triggered. From a human rights point of view, multi–level governance has ensured legislation that violated certain rights of agents is revoked, such as the right to privacy set out by EU human rights law [48, 70]. In Artificial Intelligence, Diaconescu and Pitt [57], Pitt and Diaconescu [198], and Jiang, Pitt and Diaconescu [133], view the multi–level governance paradigm as being applicable to artificial societies such as smart grids. In [57] sub–systems of energy users are governed by institutions, each sub–system (e.g. a small housing community) has its own governance to manage its own energy production and usage. Sub–systems can cooperate, requiring coordinated regulations, such as to ensure greater energy stability by sharing energy. Coordinated regulations can be achieved with hierarchical governance, where an overarching institution ensures institution's regulations at lower–levels are compatible. Multi–level governance helps to ensure that regulations are subjectively ideal in the sense of being coordinated across jurisdictions and upholding agents' rights.

In multi–level governance, institution designs are governed with regulations that govern other regulations. Since multi–level governance is regulatory it implies compliance is possible but not guaranteed. Hence, regulations governing regulations also give the possibility for non–compliant institution designs. In order to ensure institution designers comply, non–compliance can result in legal action being brought about [210], which Smith [219, p. 15] argues is necessary for effective multi–level governance.

From a philosophical tradition, Von Wright argues that [236] regulation governing regulations (or in his words, higher-order norms) are really there to *transmit will*. By which we mean, for example, a regulation requiring another regulation to oblige communications meta-data is stored, is really issued because the issuer wishes not just that there is an obligation for meta-data to be stored, but also that meta-data is *actually* stored. That is, the issuer is transmitting their will for meta-data to be stored via an intermediary regulator, operating at a lower governance level, acting as a conduit for their will. Following this idea, it has been shown by Boella and van der Torre [29] that in order to transmit will via regulation governing regulations there must be regulation *enforcement* with reward and/or punishment. Two types of enforcement required for effective regulation governing regulations, need to be distinguished. First, enforcement of the regulation governing regulations, such as punishing EU member states for not implementing an EU directive, which helps to ensure the will is transmitted from the higher-authority to a lower-authority. Secondly, the enforcement of the regulations governing society, which are designed to be compliant with regulation governing regulations. For example, the Data Retention Directive [71] requires other regulations to not just oblige meta-data to be stored but also to enforce that obligation by punishing for not storing meta-data. Without enforcement, a regulation could be compliant with a directive by obliging meta-data to be stored and yet that obligation would be meaningless since it is unenforced. Moreover, enforcement is required so that the will of the higher-authority (e.g. an issuer of an EU directive) is transmitted all the way down to the societal level. Philosophically, regulation governing regulations only make sense if firstly they are enforced through reward/punishment for (non–)compliance and secondly if they also require the enforcement of the regulations that they require to be implemented (if any).

Determining such (non–)compliance in multi–level governance is different from determining the compliance of societal members with an individual institution. The main difference is that in multi–level governance, when compared to single–levelled societal governance, regulations operate at different levels of abstraction. Regulations are the most abstract in the highest–level institutions and the most concretely defined in the lowest–level institutions. 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, such as supranational agreements or EU directives, the terminology is more precise but countries can comply in different concrete ways. For example, the Data Retention Directive [71] states that member states should legislate for communications metadata (e.g. the time of a phone call) to be stored between 6 and 24 months. The directive’s regulation is far clearer than human rights regulations, but does not provide the precise data retention time. At a slightly lower–level, such as at the level of nation–states, regulations are more concrete. For example, providing a precise time in which data should be stored. In multi–level governance increasingly abstract regulations are prescribed at increasingly higher levels of governance which can be interpreted in many different ways, making determining compliance of institutional designs different from determining compliance of a society’s members.

Institutional design compliance is determined by legal monitors such as courts, which interpret concrete regulations to determine if they violate more abstract regulations. To give an example, the European Court of Justice [70] determined that the Data Retention Directive’s relatively concrete requirement for communications metadata to be stored violated the EU Human Rights Charter’s requirement for personal data to be processed fairly [72]. The judgement was based on the 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 personal data 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 Searlian institution concept we adopt, the context–sensitive rules linking concrete and abstract social facts are constitutive rules. Hence, the relation between concrete and abstract *norms* as found in multi–level governance is derived from constitutive rules.

To summarise, institutions govern the design of other institutions in what is known as multi–level governance, the main points are:

- Constitutive rules provide context–sensitive links between concrete and abstract concepts and through a derivation, concrete and abstract norms. Chapter 3 and Chapter 4 adopt constitutive rules to determine links between concrete and abstract norms in different social contexts.

- In multi–level governance abstract regulations at higher–levels are used to govern concrete regulations at lower–levels. Non–compliant concrete regulations results in punishment. Chapter 3 and Chapter 4 look at formalising and automatically detecting (non–)compliance in multi–level governance. The objective is to flag problems to institution designers before they enact non–compliant institutions (and hence are subject to punishment) and to support higher–levels of governance in monitoring the compliance of lower–level institution designs. In order to support institutional designers in enacting compliant designs, Chapter 5 looks at finding explanatory rectifications for non–compliant institutional designs.

- Enforcement should be ensured for regulation governing regulations in two senses.

> Firstly, the regulation governing regulations should be supported with rewards and/or punishments. We do not look specifically at formalising reward and punishment for regulation governing regulations, but the fact that it should and does exist motivates the contribution of automated compliance checking in Chapter 3 and Chapter 4. Secondly, regulations governing regulations issued by higher-level authorities really require that lower-level authorities do not just implement certain regulations in order to be compliant, but also enforce those implemented regulations. We do not formalise the general principle of both implementation and enforcement required by regulation governing regulations, but we do formalise the *specific* requirement of the Data Retention Directive [71] to be supported with punishments for non-compliance in Chapter 3 and Chapter 4.

GOVERNING THE INSTITUTION DESIGN PROCESS: CONSTITUTIVE SECONDARY LEGAL RULES ASCRIBING RULE CHANGE

Constitutive rules define how a social reality grows out of brute facts describing the ground truth. Moreover, constitutive rules make certain social actions, such as marriage, possible. Since brute facts (the ground truth) are liable to changing, it follows that the social reality is liable to changing too and so too the social actions that are possible. Consequently, constitutive rules both describe how the social reality is built and how it evolves over time. Biagoli [20] views an institution itself as having similar dynamic qualities, referring to institutions (legal systems) as organic sets of rules that in themselves change over time. Constitutive rules regulate the change of the social reality but are also subject to changing themselves.

According to Hart [117] the complexity of institutions necessitates institutional rules with the sole function to regulate rule change. Hart refers to these rule–change regulating rules as *secondary rules*. Secondary rules state how and when rules can be changed, modified, repealed and by whom. To give an example, the Italian constitution states that after a law is voted to be enacted by one of the houses of government it is [204][Art. 73] "promulgated by the President of the Republic within one month of their approval". Such rules regulate rule change by describing the institution design *process* that must be followed for rule changes to take place. In other words, rules exist with the exclusive function to state what actions constitute rule changes and thereby make the social actions of changing institutional rules possible.

Biagoli [20] views rules regulating rule change as *constitutive* rules that ascribe what constitutes a rule change. From this perspective, a physical rule change does not necessarily count–as a rule change. Physically changing a rule in the Italian rule book (i.e. writing it down) does not count–as changing the rule on its own. Rather, only the president approving a law previously voted for by one of the government houses counts–as a rule change, making the social action of rule change possible.

Following this idea, just as constitutive rules ascribe a social reality, making the social reality possible, constitutive rules ascribing rule change make rule changes possible. An analogy can be made to Ricciardi's argument [206] of the rules of chess constituting the game of chess where moves made outside of chess' rules can physically be made but then the game of chess is not being played. Likewise, physical rule modifications can be made that do not follow an institution's constitutive rule change rules, but then according to the institution the rule change has not actually taken place. Taking this view to its logical conclusion – an institution's rule–modifying counts–as rules constitute its own rule–change system and ascribe the rule changes that *can* be made.

The philosopher Suber [220] famously describes changing institutional rules as a game commonly known as Nomic. In Nomic, a move is proposing a rule change, debating it and then applying it according to the constitutive rules of the game (e.g. by majority vote or some other mechanism). Changing the system's rules can affect which rule changes are possible in the first place. For example, changing which agents are constituted as being able to change rules by participating in a vote. Suber observes many paradoxes can also arise, in the simplest case by modifying rules that make rule modifications possible. It follows that an agent wishing to change institutional rules to meet their goals, must understand how rule changes affect the built social reality and similarly which rule changes are possible.

The effects of rule change on the social reality and possible rule changes also takes a temporal dimension. At the very least, changing rules in the present affects the social reality and possible rule changes in the future. In some institutions it is only possible to change rules from the present onwards – such as ascribed by the United States Constitution [221, Art. 1 Sec. 9 Cl. 3] "No Bill of Attainder or ex post facto Law shall be passed". In other institutions rule modifications in the past (retroactive modifications) are possible, such retroactive modifications have been made in the United Kingdom [192, Sec. 58]. Given that changing rules affects which rule changes are possible, changing rules in the past, present and future can affect which rule modifications are possible or even happened up until the present.

In other cases, constitutive rules ascribe rule changes conditional on the effects of the rule changes. An example is found in the European Convention on Human Rights [48, Art. 7], which explicitly states that retroactive changes to rules are only possible if they do not criminalise formerly innocent people in the past. In order for an agent to determine what rule changes they can make or what rule changes actually took place according to the institution, the hypothetical effects of rule changes must be accounted for.

The main points are:

- Institutional rules are dynamic and subject to being changed over time. In order to determine what the social reality is at any given point we need to understand how rules have been modified and the effects of modification. Chapter 6 provides reasoning for rule modifications.

- Constitutive rules state what rule changes *can* be made, that is, the rule changes that are recognised by the institution as being valid. Chapter 6 adopts these kinds of constitutive rules, earlier chapters 3 to 5 adopt constitutive rules in general but not rule–modifying constitutive rules.

- Rule changes are conditional on the built social reality, the rule changes that have and will take place and the potential effect of the rule changes on the social reality. Chapter 6 pays special attention to an interdependency between rule changes that are conditional on the built social reality and are able to change the social reality.

2.3. FORMAL FOUNDATIONS

In the previous section we saw that institution designers are governed in what institutions they *should* design and institutional rule enactments they *can* make. The idea of this dissertation is to formalise practical reasoning in a *single framework*: to determine which institutions should be designed, how institutional design non–compliance can be explained to support rectification and what rule modifications can be made. Several foundations for the framework are elicited based on the requirement for a practical framework, and our overview of the informal concepts we wish to formalise given in the previous section. These foundations are:

- A temporal setting – the physical reality is not static, brute facts are subject to change, consequently the social reality is also subject to change. When a currency is decommissioned, the paper that counted–as money no longer counts–as money. If you are an academic, you are obliged to submit papers to conferences before their paper submission deadlines. Practical reasoning implies reasoning for realistic institutions and consequently reasoning for a temporal setting.
- Constitutive rules – institutions do not govern over a brute reality, but rather a social reality established by constitutive rules. Constitutive rules are a theory of institutional language and to paraphrase Searle [215, p. 13] if there is no language (e.g. represented as constitutive rules) then there can be no deontology.
- Modal norms – institutions govern and guide agents towards an ideal with norms. It is simpler to reason about and represent regulations governing regulations using a modal rather than evaluative norms, as this dissertation sets out to do.

On top of these foundations, this dissertation lays the following building blocks for formalising the governance of institutional design and enactment:

- Automatically detecting non–compliance in multi–level governance:
  - Regulations governing regulations – in multi–level governance institutions act to govern other institution designs. The instruments governing institution designs are regulations governing other regulations.
  - Abstraction based on constitutive rules – in multi–level governance institutions are designed at different levels of abstraction. Lower–level institution's concrete regulations are interpreted for whether they comply with the abstract regulations at higher–levels of governance. The interpretation between concrete and abstract concepts is based on constitutive rules. In turn, the interpretation of concrete norms in terms of the abstract norms that govern them is derived from relationships between concrete and abstract concepts defined by constitutive rules.
- Automatic resolution of non–compliance – when an institution is designed that is non–compliant, punishments can be issued. In order to avoid punishment an institution designer should rectify the underlying causes of non–compliance rather than enacting a non–compliant institution. The assumption is that it is preferable to rectify

non–compliance whilst remaining as closely to the institution's original design goals rather than not enact an institution in the first place.

- Determining which rule–changes *can* be made:
  - Rule–modifying constitutive rules – institution designers are constrained in which rule modifications they can make by constitutive rules. These constitutive rules ascribe rule modifications based on the context in which the modification takes place, including the hypothetical effects of rule modification.
  - Modifiable institutions in the past, present and future – institution designers can make modifications to institutions. These modifications can be at any point in time made possible by the rule–modifying constitutive rules.

In summary, this dissertation builds on foundational concepts comprising institutional reasoning in a temporal setting, constitutive rules and modal norms. Building on these concepts, this dissertation contributes novel formalisations of institutional notions for institutions governed in multi–level governance, explanations for non–compliant institution designs and institutions comprising rule–modifying constitutive rules, where the institution can be modified over time. Collectively, these novel formal building blocks allow us to reason about institutional design and enactment governance.

2.4. FORMAL APPROACHES

In this section existing formal approaches for normative and institutional reasoning are compared. We focus on approaches that provide useful historical and conceptual context or provide foundations on which to base our framework. In formal philosophy, deontic logic is the field dedicated to the study of 'ought' and other normative statements relevant to institutional reasoning. This section overviews a few systems of deontic logic and related developments.

2.4.1. STANDARD DEONTIC LOGIC

Von Wright's Standard Deontic Logic (SDL) [1] [234] is the first deontic logic widely considered to be a viable formalisation of ought (see [86, p. 5] for a historical overview dating back to medieval times). SDL is situated in a propositional setting. It introduces modal operators over propositional formulae p to express the deontic modalities of obligation ($\mathbf{O}p$), prohibition/forbidden ($\mathbf{F}p$) and permission ($\mathbf{P}p$).

Von Wright introduced an axiomatisation for the modal deontic operators which, among other axioms, provided equivalences between the deontic modalities that have been used in many other systems of deontic logic. These equivalences are, what is forbidden is obliged to the contrary ($\mathbf{F}p = \mathbf{O}\neg p$) and what is permitted is not obliged to the contrary/not forbidden ($\mathbf{P}p = \neg\mathbf{O}\neg p$) (i.e. permission is the dual of obligation). This scheme gives us the well known deontic square of opposition depicted in Figure 2.2, succinctly summarising the relationships between the different modalities.

[1] n.b. Standard is a moniker and does not denote that SDL is by any means *the* standard [86, p. 39], hence we do not restrict our search for an appropriate formalism to just SDL.

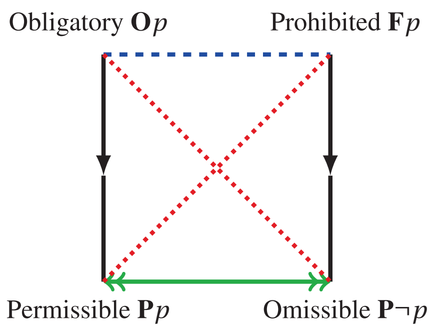

Figure 2.2: The Deontic Square of Opposition [168]. Contraries are denoted with - - - , contradictories are denoted with ······, implications are denoted with ➛ and sub-contraries are denoted with ➛➛. For further geometric analyses of deontic and related logics see [179].

SDL is not without its problems, rather it is susceptible to a number of so–called paradoxes. In SDL conditional norms, such as if there is a fence (f) it should be a white fence (w) are represented with formulae mixing propositional sentences and deontic operators ($f \rightarrow \mathbf{O}w$). A problem arises if we extend the previous example with a prohibition on there not being a fence ($\mathbf{F}f$), a fact that there is a fence (f) and the implication that a painted fence implies there is a fence ($p \rightarrow f$). The problem is that it leads to a contradiction in SDL – there ought to be a fence and not a fence ($\mathbf{O}f \wedge \neg f$) [201]. However, in principle, there should not be a contradiction. In general, a problem arises where there is a primary norm (forbidden for there to be a fence) that is supported with a secondary norm that when applied represents a level below what is ideal (the fence should be painted white). Potentially, such a secondary norm can also be used to represent a punishment for sub–ideal behaviour or circumstances, for example, you should not speed but if you do then you should pay a fine. Such statements are known as contrary–to–duty norms (CTDs). They are viewed as being an important aspect of the law where norms are commonly used to define punishments for non–compliance or state what the sub–ideal circumstances are [134, 135]. Hence, it is important contrary–to–duty norms are handled adequately.

Several proposals address the problems caused by CTDs in SDL. A common view is to distinguish between *prima facie* oughts, the oughts that on the face of it hold, and *all things considered* oughts [86, p. 256]. For example, *prima facie* it is forbidden for there to be a fence and obliged the fence is painted white. All things considered, it is only obliged the fence is painted white. Proposals addressing CTDs thus derive 'all things considered' oughts from prima facie oughts. For example, by using defeasible reasoning [209] to exclude *prima facie* oughts that are violated, in favour of secondary oughts. Other significant approaches are preference–based formalisations [228, 230] which interpret ought as ideal and CTDs as representing sub–ideal circumstances, then separating contradictory oughts into separate worlds of ideality. For example, in the ideal world it is forbidden for there to be a fence, in a sub–ideal world the fence ought to be painted white and both do not hold in the same ideality (hence no contradiction). In all of these approaches the semantics of classical implication are replaced with non–classical semantics.

Relevant to our dissertation is the fact that, as expected, such a modal representation for norms supports nesting. Nested deontic modalities such as $\mathbf{OO}p$ are grammatical in SDL. Unfortunately, SDL is a strictly non–temporal logic and hence unsuitable on its own for the purposes of this dissertation. Furthermore, SDL lacks constitutive rules; with material implication as the only conditional of which context is not a part.

2.4.2. ANDERSON'S REDUCTION

Anderson proposed a logic that replaces SDL's deontic operators in favour of evaluative norms [9, 10]. In Anderson's proposal SDL is reduced to alethic modal[2] logic by replacing $\mathbf{O}p$ with a formula stating that it is necessary ($\Box$) that going against the norm ($\neg p$) materially implies a violation (V): $\Box(\neg p \rightarrow V)$. In Anderson's reduction, norms are rules that ascribe violations.

Grossi [105] developed Anderson's idea further by proposing a logic for evaluative norms expressed as *constitutive* rules. In Grossi's proposal a logic of context and ascriptions is proposed. In his proposal implications operating on a context i are introduced, denoted as $\Rightarrow_i^{cl}$. Evaluative norms stating that p is a duty are represented as a formula ascribing a violation V in a context i: $\neg p \Rightarrow_i^{cl} V$. In Grossi's proposal, Anderson's reduction is realised in a logic of constitutive rules. Hence, Grossi realises the context–sensitive ascription of abstract institutional concepts, including violation, in a deontic logic.

Aldewereld et al. [4] build on these proposals with a formalism for reasoning about evaluative norms at different abstraction levels. They combine constitutive rules to ascribe abstract institutional facts from more concrete brute or institutional facts. Constitutive rules define norms by ascribing evaluative statements (compliant and violation in their case). To give their example, the rule "transferring money with a credit card counts–as payment" ascribes the abstract institutional fact of payment from the more concrete fact of using a credit card. A norm is then "paying counts–as fulfilment'. The norm is abstract in that it states someone should make a payment but it does not concretely define, on its own, what payment exactly is. By providing a semantics for counts–as statements that includes a form of transitivity, norms can also be concretised. For example, through transitivity, using a credit–card counts–as fulfilment. In their proposal, constitutive rules are contextual and hence transitivity only holds between constitutive rules if their contexts are compatible. To summarise, Aldewereld et al. take the principles behind Grossi's reduction to constitutive rules in order to reason about concretisation of abstract norms.

Developments in the reductionist approach to norms are relevant to this dissertation. First, they offer a way to reason about abstract norms, in this case by concretising abstract norms. Hence, it seems that the same approach can be taken to multi–level governance. Perhaps, abstract regulations governing other regulations at higher–levels of governance can be reified to determine their concrete meaning and therefore whether concrete norms violate the abstract norms. Conversely, perhaps the same reasoning can be reversed to take concrete norms and abstract them to determine if they are compliant with more abstract norms. However, a major stumbling block is that by reducing norms to evaluative constitutive rules it is no longer straightforward to represent and reason about regulation governing regulations, as we argued previously. Consequently, this dissertation does not take a reductionist approach to norms.

[2]If this is Greek to you, alethic modalities are modalities denoting truth, in comparison to deontic modalities, which denote normativity.

2.4.3. TEMPORAL DEONTIC LOGICS

The previously described approaches are situated in a static setting with no consideration for time. Hence, they deal with non–temporal norms. For example the norm 'you should not murder' has no temporal element. However, in the real world duties are often temporal and contain deadlines. For example 'you should submit your paper before the submission deadline'. Thus, there are deontic logics that deal explicitly with time, by which we mean a temporal ordering of states containing formulae, including temporal obligations.

Several temporal logics independent from deontic logic already exist. The prominent ones are Linear Temporal Logic (LTL) [199], Computational Tree Logic (CTL) [69] and CTL* [68], which combines both. There are also logics of actions as transitions between states. Dynamic logic [116] being a prominent action logic, which combines modal logic operators of necessity and possibility with actions to express and reason about statements such as 'it is possible that action a will cause p to hold'. A common approach to temporalising deontic logic is to combine a deontic logic and one of the aforementioned temporal or action logics.

To name a few. Broersen et al. [33] combine SDL and CTL. They introduce dyadic deontic modalities ($O(\rho \leq \delta)$) representing a propositional formula ρ should hold before or at the same time as a formula δ representing the deadline. In Broersen et al.'s formalisation if δ occurs before ρ then a proposition denoting violation holds. Conversely, ρ holding before or at the same time as δ causes a proposition denoting norm fulfilment to hold. An obligation holds from one state to the next until it is fulfilled or violated, that is, a norm persists by default.

F. Dignum et al. [60] combine SDL and dynamic logic. Obligations can take the form $O(\alpha < \rho < \delta)$ representing an obligation conditional on α that requires action ρ to be performed before δ. States also have time indices. The special index *now* denotes the time of the current state. In F. Dignum et al's proposal instantaneous norms that must be fulfilled immediately are expressed as $O(\alpha < \rho < now + 1)$ representing that α causes an obligation for ρ to be performed before the next state.

F. Dignum and Kuiper [59] combine SDL with a logic of 'dense' time in which actions are not instantaneous. Rather, actions are performed over a time period. An obligation $O(\alpha)$ represents that an action α should be continuously performed until it is done. It is interesting to note that violations persist from one state to the next until the norm that has been violated is 'repaired' (e.g. by performing some punishing action, such as paying a fine).

To summarise, temporal deontic logics allow expressing that there should be an ordering in which propositions hold in states or actions are performed. A temporal obligation holding in a state typically states that at the present time an α should be done/hold before a δ. Obligations persist from one state to another if not discharged. Hence, if α and δ do not occur, then α should still be performed/hold before δ. Instantaneous norms then just state that an α should be done/hold before the next state. In this dissertation we adopt similar notions in Chapter 3, Chapter 4 and Chapter 5 – a deontic statement persists until discharged, and either represents one thing should be done before another or that something must be done instantaneously.

2.4.4. SEINSOLLEN AND TUNSOLLEN: OUGHT–TO–BE AND OUGHT–TO–DO

There is a difference between ought–to–be statements about what should hold (seinsollen) and ought–to–do statements about what should be done (tunsollen) [35]. In SDL [234] the two are not distinguished; SDL abstracts away from whether a proposition denotes an action or a state–of–affairs. Subsequent developments have focussed on one or the other, or both.

- **Ought–to–be** – logics of this type are characterised by obligations that hold in states and are about propositions that hold in states. Examples include [33, 53, 98]. It is interesting to note that ought–to–be can represent two types of obligation explored in [98]. The first is an achievement obligation: a state of affairs should be achieved before a deadline. The second is a maintenance obligation: a state of affairs should be maintained up until a deadline.

- **Ought–to–do** – logics of this type are characterised by obligations that hold in states and are about events or actions that occur when transitioning between states. Examples include [33, 58, 60, 122]. It is interesting to note that Broersen et al. [33], Horty and Belnap [122], and V. Dignum et al. [58] use a stit (see–to–it–that) operator (or a version thereof) to represent the action of an agent bringing about a particular state of affairs. For example $stit\phi$ is an action to bring about ϕ in a state. Consequently, ought–to–do statements are framed as ought–to–bring–about a particular state, abstracting away from labelled actions whilst still remaining ought–to–do statements.

In this dissertation, we look at obligations about agents' actions (or events if we view actions as requiring intention). This is because they seem to be more concrete and are frequently found in the law – you should not *murder*, you should not *steal*, if you do either you should *go* to prison. We also look at obligations about obligations to reason about regulations governing other regulations. Yet, obligations are normative propositions that *hold* in states and so obligations about obligations are seinsollen ought statements about a state of affairs. As Von Wright points out, if obligations are about actions then nested modalities are nonsensical [235, p. 91]. Where does this leave us, do we look at both ought–to–do (obligations about actions) and ought–to–be (obligations about obligations which hold in states)?

One approach would be only to look at ought–to–do. Here, the idea is to follow Wansing's suggestion [237] that nested obligations are obligations to see–to–it–that obligations hold. The idea being that a nested obligation is an obligation to perform a locutionary act to establish an obligation, for example, if we are an authority with legal power, by saying "I command you to do X" we can see–to–it–that there is an obligation to do X (see Searle's account of how to go from is to ought, or obligations from locutionary promises [212]). The problem is, this means imposing obligations requires agency. However, in this dissertation we set out to look at institutions with regulations that govern other regulations – an institution is not a type of agent, it belongs to a class with instantiations that include the written law, promises, contracts etcetera.

Hence, in this dissertation we combine ought–to–do for obligations that govern agents' actions and ought–to–be for norms governing other norms. The combination of both was also looked at by D'Altan et al. [51]. Chapter 3, Chapter 4 and 5 combine both types of ought. Other aspects, of ought–to–be statements such as maintenance and achievement obligations are not considered in this dissertation.

2.5. PRACTICAL FORMAL GOVERNANCE

In this section a number of frameworks for practical formal governance are proposed. There are many links between deontic logics and the following research described and hence the two cannot be easily disentangled. Rather, the following frameworks are characterised firstly by drawing from a wider informal sphere such as legal and organisational theory, whilst deontic logics come from a philosophical tradition. Secondly, by being pragmatic, such as by having a low computational complexity (e.g. by talking about obligations over single propositions rather than arbitrary propositional formulae) or providing clear links to implementation. It is important to note that this dissertation falls firmly in practical formal governance, since it provides a formal account that is characterised in such a way that the decision procedure is obvious and/or is coupled with a computational framework (i.e. an implementation).

2.5.1. ORGANISATIONAL FRAMEWORKS

Whilst this dissertation focuses on institutional reasoning a similar coordination mechanism, multi–agent organisations, is also widely studied for formalisation. Like institutions, organisations comprise a normative dimension. In contrast with institutions, organisations also comprise constructs such as roles agents play and interaction patterns to achieve various organisational objectives. If an institution is legislation, contracts and promises, then an organisation is a university or a corporation within which institutions can play a part in the form of contracts and other bodies of rules and regulations. Consequently, formal reasoning between the two areas has much overlap (e.g. norms), institutions tending to focus on legal concepts (e.g. legal empowerment to affect an institution) whilst organisations focus on cooperation concepts and processes to achieve goals (e.g. interaction patterns).

HarmonIA [233] is an organisational framework that focuses on conceptualising 'electronic organisations' and methodologies for their design. In HarmonIA, organisations comprise different levels of abstraction. These are an abstract level, concrete level, rule level and a procedure level. At the abstract level, the organisation defines abstract norms such as statutes. At the concrete level the organisation refines the abstract level's norms to concrete policies. At the rule level further refinements are made to the concrete norms as rules for agents to follow in order to comply with the concrete norms. The procedure level comprises the actual computational implementation of the rules. In HarmonIA norms are modal and their refinement/concretisation is based on the counts–as relation. However, a semantics for norm refinement is not investigated. Instead the *approach* is given [233, p.94] and tools to automate refinement/concretisation are left for future work [233, p.168]. HarmonIA's focus is on the conceptualisation and design methodology for organisations in MAS at different abstraction levels, but not the formal reasoning.

MOISE+ [124] provides a conceptualisation for designing organisations comprising roles, dependencies between roles and norms. In MOISE+ norms have a modal representation. In contrast to much of the work in formal philosophy and elsewhere, MOISE+ does not provide a formal semantics for norms or other organisational concepts. Rather, it focuses on programming organisations at an agent level and an organisation–control level. At the agent level, the J–MOISE+ (sub–)framework provides a way to program agents to enact roles in a MOISE+ organisation. At the organisation level, the S–MOISE+ [125] (sub–)framework provides an interface for agents to join a MOISE+ organisation. Through the S–MOISE+ interface organisational constraints are regimented on agents as *hard* constraints such that agents are forced to comply. MOISE+ is an agent–organisation framework focussed on programming agents and organisations, there are many aspects relevant to institutions that are not considered (e.g. constitutive rules, abstraction and rule change).

OperA [61] is a framework providing an organisation design methodology and formal organisational reasoning. Conceptually, an organisation in OperA is analogous to a human organisation. An organisation comprises a social structure defining organisation objectives, agent roles and relations between roles; an interaction structure describing scenes for agents to play roles in and landmark objectives for the organisation to achieve; a normative structure describing the norms agents must abide by when fulfilling roles and interacting in scenes; and a communicative structure describing the communicative acts agents can make. An organisational logic formalises OperA organisations. The logic combines deontic logic and temporal logic to define normative reasoning for the norms agents adopt when enacting roles and reasoning about the responsibilities and capabilities endowed on agents for enacting those roles. The logic also determines when organisational landmarks are achieved. The formal theory provided by OperA is implemented with the Operetta tool [5] for checking organisational properties. In OperA organisations are the formal counterpart to human organisations and a formal semantics provides organisational reasoning, but OperA lacks many aspects found in the legal/institutional sphere (again, constitutive rules, abstraction and rule change).

OperA+ [133] by Jiang builds on the OperA framework with additional organisation design concepts and methodology, and novel computational mechanisms for organisational reasoning. Enhancing OperA's organisation design methodology, OperA+ proposes multiple levels of abstraction. The most abstract level comprises organisational objectives. From here, the design process enters a contextual level where the organisation design is concretised by decomposing the organisation objectives into sub–domains or situations. Jiang uses a top–level objective of train maintenance as an example where in this case the sub–domains are planned and unplanned maintenance. Such sub–domains are then concretised further at a solution level, by instantiating the sub–domains as social structures and normative structures. Like in OperA, the social structures comprise roles agents play and the normative structure comprises norms. Unlike OperA, OperA+ also includes constitutive rules for establishing institutional actions from brute facts. Opera+ also introduces an operational level, comprising groups of agents enacting roles represented as group preferences for particular actions. OperA+ organisations are operationalised using Coloured Petri Nets (CPNs), a graph–based formalism for describing distributed systems and computing their states over time. Norms are represented as CPN graphs, akin to an evaluative norm representation. Since norms have an evaluative representation, it is unclear how to extend OperA+'s representation and reasoning to norms governing other norms. Other aspects missing from OperA+ but focussed on in the dissertation you are reading now include a semantic definition of links between concrete and abstract norms and governing rule change.

2.5.2. INSTITUTIONAL FRAMEWORKS

Institutional frameworks in MAS can be understood as analogous to their informal counterparts (e.g. legislation). Typical elements are constitutive rules, norms, and the ability for the institution to evolve from one institutional state to the next according to the change of brute facts and the institution's constitutive rules. Research on institutional reasoning falls largely into two spheres, legal reasoning (e.g. for legislation) and open multi–agent systems (e.g. artificial markets). We describe three frameworks in what is by no means an exhaustive overview, instead we focus on the frameworks that are closest to what this dissertation is trying to achieve.

OCeAN [78, 81, 82] (**Ontology CommitmEnts Authorizations Norms**) is a high–level institution specification language and an operationalisation in the discrete event calculus. In OCeAN an ontology specifies institutional actions, events and roles agents play (hence there is an organisational flavour to OCeAN). OCeAN adopts constitutive rules ("A counts–as B in a context C") in order to define an agent communication language. Here, the communicative act 'A' counts–as an institutional action in a context where the communicative agent is playing a role that authorises that agent to realise the institutional action. For example, an auctioneer telling everyone the auction is open counts–as opening the auction, where playing the role of auctioneer represents the agent's authorisation. OCeAN also contains a normative component, adopting the usual deontic modalities in a modal representation. OCeAN's focus is on ontologies, social commitments and organisational concepts such as roles.

TMDL [98, 103] (Temporal Modal Defeasible Logic) is a non–monotonic logic for reasoning about institutions over time. The institutional concepts captured are the deontic modalities, constitutive rules, normative power (the ability for an agent to establish a norm through a communicative act), and positive and negative rights. The logic is temporal in the sense that propositions (including modal statements) hold at points in time when they are initiated by an action according to a rule, and persist in holding (inertia) until they are terminated. The logic's modus operandi is *defeasibility*, the idea that a proposition is defeasibly proven but a proof to the contrary *defeats* the proposition and makes it disproven. In TMDL there are three types of rule in horn–clause form. Firstly, strict rules $\phi_0, ..., \phi_n \rightarrow \psi$ where the consequent is strictly provable when the antecedent is provable and cannot be disproven if the antecedent is proven. Secondly, defeasible rules $\phi_0, ..., \phi_n \Rightarrow \psi$ where proving the antecedent means the consequent is defeasibly proven, but proving contrary propositions from the consequents of other rules defeats ψ or the rule's defeasible premises and makes ψ disproven. Thirdly, defeater rules $\phi_0, ..., \phi_n \rightsquigarrow \psi$, which do not make their conclusion proven, but rather defeat contrary premises and conclusions of other rules that are defeasibly proven. To determine which rules can defeat other rules, a superiority relation between rules is specified.

Defeasible logic, such as TMDL, defines a proof procedure that can be implemented in Prolog [13]. The idea is that a derivation asserts the conclusion of a rule, then all the ways to attack the derivation and its conclusion are found, all the attacks of the attacks are found, and so on. This process continues until no more attacks can be applied and it is determined whether there remain any undefended attacks of the original assertion, in which case it is not proven, and otherwise it is proven. TMDL, through its proof procedure, makes it possible to reason about exceptions as found in the law and also exceptions to exceptions etcetera.

InstAL [42, 43] (INSTitution Action Language) is a framework for temporal institutional reasoning. An institution specification in InstAL comprises events that can occur and fluents that can hold in the institution, as well as constitutive rules. The events can be observable corresponding to a brute fact, or institutional such as 'payment' or denoting a norm is violated. Fluents represent institutional facts about a domain, the deontic statements of obligation and permission in a modal–form, and institutional empowerment denoting an event is *empowered* to affect the institution. In InstAL, anything not permitted is prohibited and hence a prohibitive society is reasoned about. The constitutive rules ascribe institutional events from observable events (corresponding to the notion of a brute fact changing) and other institutional events. The constitutive rules also ascribe the effects of institutional events on the institution's state, causing institutional fluents to be initiated or terminated from one state to the next. Regulations are represented as constitutive rules where the consequents are the initiation or termination of obligation, permission and empowerment fluents.

InstAL's semantics are defined with a formal framework based on set theory and mathematical functions. Complementing the formal framework is a computational framework that executes institution specifications in response to a trace of observable events. The computational framework uses Answer–Set Programming (ASP), a non–monotonic logic programming language [91, 92] based on the stable model semantics [93]. The output of the framework is a state–transition system. Each state contains fluents representing institutional facts and normative positions (obligations and permissions). Transitions between states are the events produced by observable events and the transitive closure of constitutive institutional event generation rules. Fluents persist by default from one state to the next unless terminated, capturing the common–sense law of inertia using Event–Calculus–like constructs. InstAL is complemented with a framework for finding rule changes to meet certain properties, in this case resolving inconsistencies between norms, based on Inductive Logic Programming in ASP [154, 157].

2.6. KNOWLEDGE GAPS AND APPROACH

In this section we compare the frameworks described previously, depicted in table 4.7, in terms of their support for PARAGon's objective of institutional design and enactment governance reasoning. The comparison is split between foundational representation and reasoning requirements for our formalisation aims, and the necessary conceptual building blocks (shown in **bold**) we require to formalise institutional design and enactment governance. We also denote those frameworks with a corresponding implementation, although the degree to which there exists an implementation varies from one framework to another (see the previous discussion).

The *foundational* representation and reasoning constructs for this dissertation are modal norms, since they can be extended to nested modal norms and therefore regulations governing regulations; a temporal setting for real–world relevance and the reasoning about rule–change over time; and constitutive rules for reasoning about institutions operating at different levels of abstraction. The conceptual *building blocks* we require are: rule–modifying constitutive rules for ascribing which rules an agent can modify and when; temporal rule modifications; regulations governing regulations for reasoning about multi–level

	Modal Norms	Temporal Setting	Constitutive Rules	**Regulations Governing Regulations**	**Abstract Norms**	**Automatic Rule Change for Compliance**	**Rule-modifying Constitutive Rules**	**Temporal Rule Modifications**	Implementation
Standard Deontic Logic [234]	✓			✓					
Anderson's Reduction [9, 10] and Subsequent Developments [4, 105]			✓		✓				
Temporal Deontic Logics [33, 59, 60]	✓	✓							
HarmonIA [233]	~	~	~		~				
MOISE+ [124]	✓	✓							✓
OperA [61]	✓	✓							✓
OperA+ [133]		✓	✓						✓
OCeAN [78, 81, 82]	✓	✓	✓						✓
TMDL [98, 103]	✓	✓	✓						✓
InstAL [42, 43, 154, 157]	✓	✓	✓			~			✓

Table 2.1: Comparison between the state–of–the–art knowledge provided by different frameworks. A tick denotes the framework provides the necessary representation and reasoning constructs. A ~ denotes the framework only provides a conceptualisation sans semantics in the case of HarmonIA or technologies that can feasibly be adjusted for our purposes in the case of InstAL. Knowledge in a regular font denotes foundational concepts that need formalising, knowledge in **bold** represents the building block concepts that need formalising to reason about institutional design and enactment governance.

governance; norm abstraction or concretisation based on constitutive rules for interpreting institutions in multi–level governance; and automatic rule change for compliance in order to help institution designers in multi–level governance enact institutions whilst avoiding punishment for non–compliance. As we can see, the foundational components are adequately supported by some frameworks, however the conceptual building blocks are not and hence it is those building blocks which the PARAGon framework addresses.

Organisation frameworks support some necessary foundational institutional reasoning constructs, such as norms and potentially constitutive rules. However, they lack the semantics for the governance of institutional design and change. In the case of the HarmonIA framework, whilst significant attention is paid to norm concretisation it is done by way of example and not with a general semantics. The same is true for the other aspects of HarmonIA. What HarmonIA does offer, is an indication of on what basis norm abstraction should be reasoned about. Namely, constitutive rules. In fact, that is the same intuition this dissertation takes, albeit we provide a formal semantics for reasoning about abstracting rather than concretising norms using constitutive rules. MOISE+, OperA and OperA+ all offer some foundational aspects that could potentially be built on by this dissertation. However, the institutional reasoning frameworks, focussing on institutions as we do, offer further institutional reasoning foundations suitable for our aims.

As for the institutional reasoning frameworks – OCeAN, TMDL and InstAL all share institutional concepts such as norms, constitutive rules and institutional empowerment. Some differences are that OCeAN also comprises organisation concepts, TMDL incorporates defeasible reasoning and InstAL has been used as a basis for Inductive Logic Programming based norm revision. In all of these frameworks norms are specified at a single level of abstraction, constitutive rules establish an institutional reality but not rule changes, and detecting and rectifying compliant institution *designs* is not captured. Hence, all frameworks provide necessary formal foundations, but do not close the knowledge gap this dissertation addresses.

The question arises as to the suitability of these frameworks for our purposes. In this dissertation InstAL is used as a basis for the institutional reasoning we contribute. The main reason is that in InstAL, due to having an ASP computational framework, it is already possible to apply existing ASP–based Inductive Logic Programming techniques in order to resolve problems with institutions. In the case of [157] the problems resolved are normative conflicts. In this dissertation we seek to resolve non–compliance in multi–level governance and hence Inductive Logic Programming techniques are used and developed towards our aims.

The following summarises how the PARAGon framework fills the gap in the knowledge. Determining compliance according to institutional design governance in multi–level governance requires: regulations governing regulations, and norm abstraction or concretisation based on constitutive rules, and is addressed with a formal framework in Chapter 3 and a computational framework in Chapter 4. Automatically resolving non–compliant institution designs to support designers in avoiding punishment requires automatic rule change for compliance and is addressed with a computational framework in Chapter 5. Determining when rule changes are legally made requires rule–modifying constitutive rules and temporal rule modifications, and is addressed with a formal framework in Chapter 6.

2.7. DISCUSSION

In this chapter we began with an overview of the informal concepts behind multi–agent systems, institutions, governance and governing governance. Two aspects stand out. First, there are many parallels between the need to govern agents (due to their autonomy and liability to deviate from the ideal), and governing institution designers (for the same reason, since institution designers are also agents). Second, the informal work looked at in the area of institutions is, in fact, fairly formal (e.g. in the case of Searle's counts–as rules). On the other hand, work on governing governance appears distinctly informal, where we largely drew on work in political science and examples from real–world legislation.

These aspects led us to identifying the main notions that the PARAGon framework needs to formalise. First, we elicited the foundational notions to formalise (institutions, norms). Then, we identified the building blocks required to formalise governing governance. For example, we elicited the need to rely on modal norms over evaluative norms to reason about regulations governing regulations. Likewise, we argued that it is clear constitutive rules play an important role in governing of governance both for providing links between concrete and abstract concepts and governing institutional enactment of new rules or changes to existing rules. What was unclear at this point was to what extent existing approaches provided the foundations or the building blocks required.

Hence, we provided a comparison between existing formal approaches. We paid particular attention to where the gaps in the existing knowledge lie that need to be filled. Based on this analysis we selected the InstAL framework from which we can use various foundational formal components on which to build the PARAGon framework this dissertation contributes.

3

Formalising Compliance in Multi–level Governance

For a large class of cases–though not for all–in which we employ the word "meaning" it can be defined thus: the meaning of a word is its use in the language[239, Sect. 43]

Ludwig Wittgenstein

This chapter is based on the following papers:
King, T. C., Li, T., De Vos, M., Dignum, V., Jonker, C. M., Padget, J., & van Riemsdijk, M. B. (2016, July). Automated Multi–level Governance Compliance Checking. *Journal of Autonomous Agents and Multiagent Systems (JAAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems. (**In Submission**)
Which extends the following paper:
King, T. C., Li, T., De Vos, M., Dignum, V., Jonker, C. M., Padget, J., & Riemsdijk, M. B. Van. (2015). A Framework for Institutions Governing Institutions. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015) (pp. 473–481). Istanbul, Turkey: International Foundation for Autonomous Agents and Multiagent Systems. [144]

In this chapter we look at *soft constraints* placed on institution designers, which prescribe the institution designs they *should* enact. The contributions of this chapter are:

- A formal representation language for institutions governing other institution designs in multi–level governance.

- A formal semantics for determining when institutions are compliant in multi–level governance, accounting for concrete regulations at lower–levels of governance, governed by increasingly abstract regulations at higher–levels of governance.

Institutions (e.g. legislation) govern societies towards ideal and coordinated behaviour with rules and regulations coupled with the means to detect compliance and issue rewards and punishments. Increasingly, institutional reasoning is formalised and computerised with automated normative and institutional reasoning frameworks (see [12] for a review). Such frameworks can support governing bodies in punishing agents and support agents in understanding their legal duties.

However, typical institutions are not written in a vacuum. Rather, institutions are constrained and regulated by higher–level institutions. This is what is called *multi–level governance* [121]. In multi–level governance, institution designers design institutions comprising rules and regulations, but whose design is also subject to regulation. In 2006 the European Union issued the Data Retention Directive [71] for harmonising member states' data retention regulations, in 2009 the UK implemented the directive with the Data Retention Regulations [226] in order to avoid fines. Yet, in 2014 the European Court of Justice ruled [70] that the directive was non–compliant with the Charter of Fundamental Rights [72], and annulled the directive. Legislating in multi–level governance exposes institution designers to risks of punishment and wasted legislating time and burdens a judiciary with checking compliance.

So far, institutional reasoning frameworks have focussed on single–levelled societal governance. Typically, automated institutional reasoning deals with regulations operating at the level of institutions governing agents and/or corporations. For example, the UK's Data Retention Regulations [226] obliges communications providers to store communications metadata. There lacks reasoning 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 chapter we look at how institutions themselves are regulated by higher–level institutions.

In particular, we look at increasingly abstract regulations at higher–levels of governance that 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 the least ambiguous. 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 European Charter of Fundamental Rights [72] contains vague regulations requiring that people's private and family life is respected. The EU's data retention directive [71] contains a more concrete regulation requiring communications service providers (e.g. ISPs and telephonic companies) to store people's communications metadata (e.g. a phonecall's time and place) within a fixed time frame. The directive's communications metadata regulation must be shown to be compliant with the Charter of Fundamental Right's more abstract right to a private and family life, or else the directive will be annulled. In fact, the directive was annulled due to violating privacy [70]. At the same time, the directive itself governs the design of institutions – member state's 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 a compliant institution differently, allowing the data retention period to be between 6 and 24 months. The UK's data retention regulations [226] are more concrete and must be shown to ensure communications metadata is stored within the required time frame, no shorter and no longer. In fact, they do just that, concretely requiring communications metadata is stored for 13 months which complies with the abstract requirement of the directive to store data between 6 and 24 months. The UK's institution must be shown to be compliant with the more abstract regulation of the directive in order to avoid fines. In turn, the data retention directive must be shown to be compliant with the more abstract charter of fundamental rights in order to avoid annulment.

To this end, this chapter contributes two main formal building blocks that together contribute a system for detecting non–compliance in multi–level governance. These novel elements are:

- A way to represent and reason about regulations governing regulations – specifically, combining norms about agents' actions with norms that oblige and prohibit the imposition of other norms in different social contexts. In this chapter we look explicitly at norms, rules that state what should occur and when. In particular, we look at norms that state what the effects of other norms should be.

- Abstraction based on constitutive rules – specifically, taking concrete norms at lower–levels of governance and defining a semantics that re–interprets those concrete norms by *abstracting* them to the same level of abstraction the higher–levels of governance govern them at. The semantics define an abstracting relation between concrete norms and their abstract definitions based on context–sensitive links provided by constitutive rules between concrete and abstract concepts. For example, the European Data Retention Directive [71] requires member states to store communications metadata. The semantics determine that storing communications metadata without someone's consent is, abstractly, unfair data processing.

This chapter is written to be as self–contained as possible. Thus, we start by describing the individual components of institutions and the approach we take to reasoning about them in Section 3.1. The representation we use is given in Section 3.2. The formal reasoning is presented in Section 3.4. We describe related work in Section 3.5. Conclusions are presented in Section 3.6.

3.1. 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 regulatory rules, higher–order norms, connections between institutions and reasoning about regulation abstraction. The InstAL (Institution Action Language) framework [42, 43] provides constitutive rules and modal regulatory rules. 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 the InstAL reasoning from being about prohibitive societies (where anything not permitted is forbidden) to permissive societies (everything is permitted unless explicitly prohibited), adding explicit prohibitions and removing explicit permissions. Whilst in InstAL brute facts are observable events or properties which hold in a state (fluents), we only consider brute facts to be observable events. For simplicity, all properties which hold in a state are institutional and not brute in our framework. Based on InstAL [42, 43], an institution in our framework specifies six elements.

Firstly, following InstAL, events. 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.

Secondly, following InstAL, fluents which 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 they have consented, which is removed if they revoke their consent. Some fluents represent the deontic positions that hold. The deontic positions denote institutional *empowerments*, and *obligations and prohibitions*. Empowerments represent that an institutional event is *empowered* to affect the institution, meaning an institutional event *can* occur in the institution. For example, "storing communications metadata is empowered to occur". Empowerments represent hard constraints on the events that can (in general) occur (for further discussion see [42, 43]). 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 positions (normative fluents) express that something should be done before a deadline. For example, an obligation to pay a fine within one month.

Thirdly, following InstAL, constitutive rules which cause institutional events to occur when observable or other 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 institutional events causing further institutional events to occur is "storing personal data counts–as unfair data processing in the context that a user has not consented".

Fourthly, following InstAL, constitutive rules which initiate and terminate fluents in holding due to institutional events. For example, "a user consenting to storing their data counts–as initiating the fluent stating the user has consented". Constitutive rules establishing what we call normative fluents are norms. For example, "a user using a communications device counts–as initiating 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.

Modal norms, which are regulative rules, are constitutive rules which initiate and terminate fluents due to institutional events. For example "a user using electronic communications initiates an obligation for the communications provider to store their communications metadata". Thus, norms are not a distinctive element of institutions, but rather a type of constitutive rule.

Fifthly, *extending* InstAL, constitutive rules which derive fluents from other fluents in a particular context. For example, "an obligation to store personal data non–consensually derives (counts–as) unfair data processing in all social contexts". 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 constitutive rules which initiate and terminate fluents the special meaning 'B' does not hold until terminated, rather, it holds when 'A' holds in the context 'C'. That is the 'Bs' in these types of rule are *non–inertial* fluents, since the Bs do not persist from over time by default until terminated (i.e. they do not possess inertia). Unlike the previous rules, constitutive rules which derive fluents from other fluents are not present in the InstAL framework.

Sixthly, following InstAL, a set of initial inertial fluents which hold in the institution's first state and continue to hold until terminated. The set of initial inertial fluents can be the empty set. 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 3.1 and described as follows.

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.

But, how are concrete regulations imposed at lower levels of governance determined to be (non–) compliant with abstract regulations at higher levels of governance? The approach we take is to firstly, link each institutional level such that concrete normative fluents

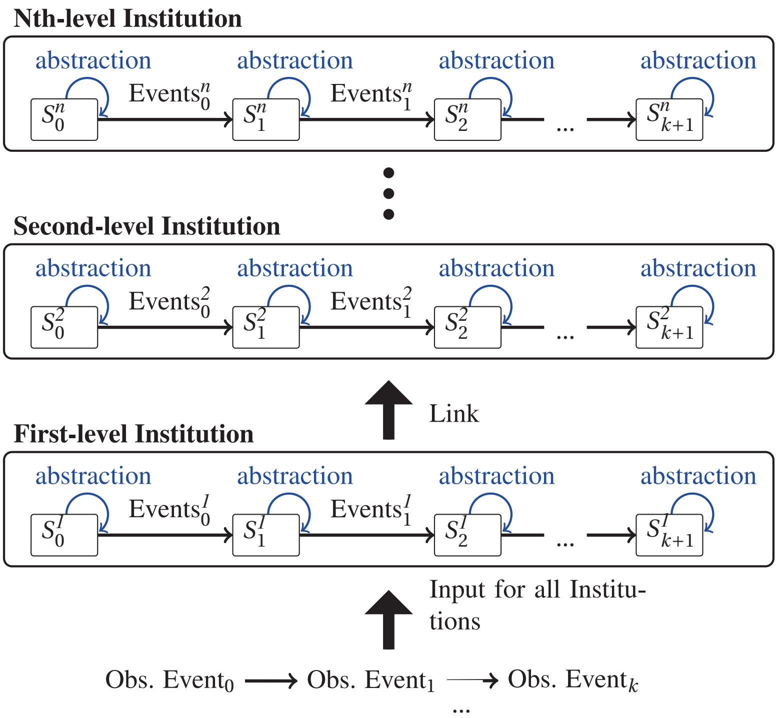

Figure 3.1: Multi–level Governance reasoning overview

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. 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*.

In order to abstract concrete normative fluents, we use constitutive rules on which to base the abstraction. Recall that constitutive rules establish links between concrete and abstract concepts, from which we can determine links between concrete and abstract norms. For example, the 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. 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.

Given that normative fluents are abstracted, they can be determined for compliance with abstract higher–order normative fluents that govern them. For example, whether the abstract obligation to oblige storing data non–consensually is in itself prohibited, or perhaps has an even more abstract meaning (e.g. unfair data processing) which is prohibited/obliged. So, concrete normative fluents from lower levels are re–interpreted as more abstract normative fluents at higher levels to determine whether in their abstract incarnation they cause non–

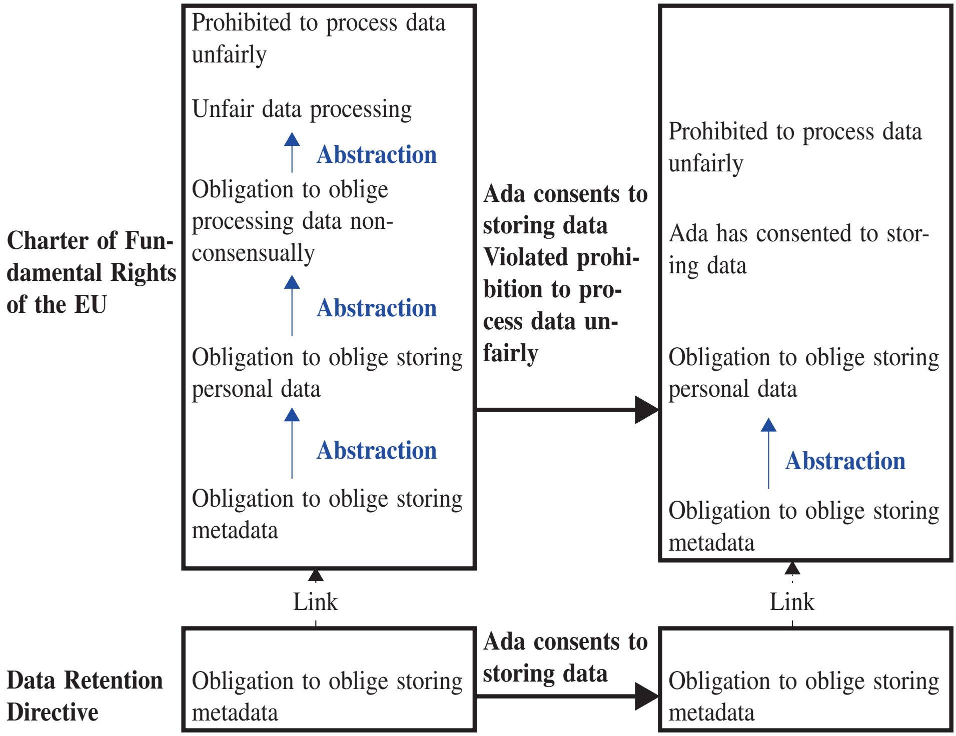

Figure 3.2: An example of abstracting normative fluents at different levels of governance based on the context. Normative fluents oblige/prohibit an aim a occurs before or at the same time as a deadline d. We use $<$ to denote one thing occurring before another and $\leq$ to denote one thing occurring before or at the same time as another.

compliance.

An example is depicted in Figure 3.2 based on the running case study. In the EU–DRD's first state there is an obligation to oblige storing communications' metadata, which is passed up to the EU–CFR. As described, concrete normative fluents are abstracted based on whether the prescribed event *counts–as* a more abstract event in a context entailed by the state. Storing metadata counts–as storing personal data. Storing personal data counts–as non–consensual data processing in the context where an agent has not consented. In the EU–CFR's first state the obligation to oblige storing metadata is abstracted to an obligation to oblige storing personal data. Then, to an obligation to oblige processing data without consent. An obligation to oblige processing data non–consensually is abstracted further to 'unfair data processing'. Unfair data processing is prohibited and thus a norm violation event occurs in the transition to the EU–CFR's 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 flu-

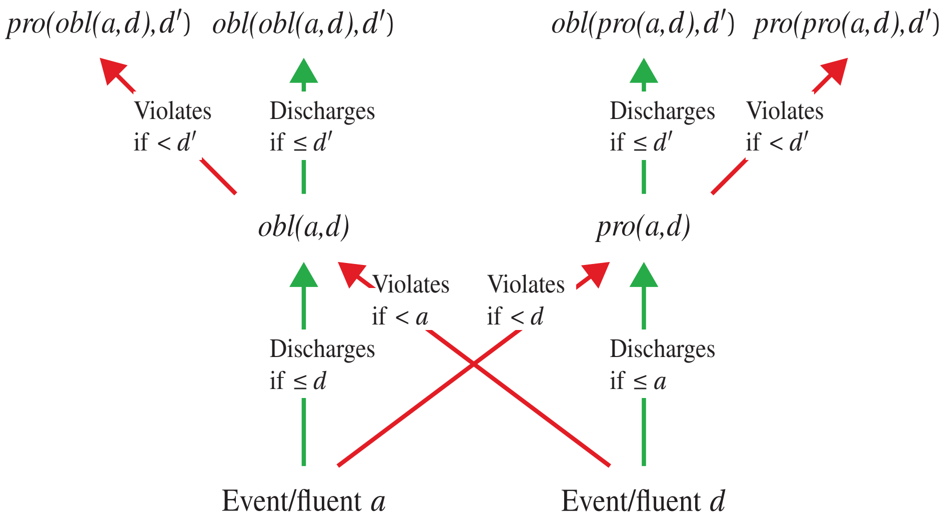

Figure 3.3: Discharge and violation (higher–order) normative fluent conditions. $< X$ denotes the event/fluent holding or occurring strictly before X causes a violation. $\leq X$ denotes the same, but the condition is non–strictly before.

ents' 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, whether concrete normative fluents imposed by lower level institutions have an abstract interpretation in higher level institutions. Moreover, whether these abstract interpretations of concrete norms violate abstract higher–order norms in higher level institutions. Such non–compliance can be determined by inspecting the sequence of events in higher level institutions for higher–order norm compliance events. The semantics provide a mechanism acting as a kind of legal monitor for multi–level governance.

3.2. FORMAL REPRESENTATION

In this section we present the syntax for representing multi–level governance. We begin with representing normative fluents. These are fluents which represent temporal obligations and prohibitions, meaning they have an aim which should be achieved before a deadline. In contrast, InstAL considers at temporal obligations and non–temporal permissions, where anything not permitted is prohibited (a prohibitive society). We look at temporal obligations and temporal prohibitions, where anything not prohibited is permitted (a permissive society).

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*. 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 [60], 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 the normative fluents lasts indefinitely. For example *pro(aim, never)* means it is always prohibited for the aim to occur or in other words 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 3.3. 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 a norm (norm discharge – $disch(n)$) or violation ($viol(n)$) 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).

A grammar to specify normative fluents is formalised as:

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. They state 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 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 institution 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 [145], 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 chapter'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 [43]. 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 which 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. $now, never \not\in \mathcal{E}$, meaning the institution cannot define when the *brute* 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 fluents which 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 which 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 holds in a state if all 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 ($\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}_{inert}} \times 2^{\mathcal{F}_{inert}}$), 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 – ($\mathcal{G} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{E}_{inst}}$), described as an event generation function. These rules are only descriptive. They specify that when an exogenous or institutional event occurs conditional on a context holding in a state another institutional event occurs.

Fluent derivation rules – ($\mathcal{D} : \mathcal{X} \times \mathcal{F} \rightarrow 2^{\mathcal{F}_{ninert}}$), 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.*
- $\mathcal{F} = \mathcal{F}_{dom} \cup \mathcal{F}_{norm}$ *is the set of fluents.*
- $\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}_{inert}} \times 2^{\mathcal{F}_{inert}}$ *is the state consequence function.*
- $\mathcal{G} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{E}_{inst}}$ *is the event generation function.*
- $\mathcal{D} : \mathcal{X} \times \mathcal{F} \rightarrow 2^{\mathcal{F}_{ninert}}$ *is the fluent derivation function.*
- $\Delta \subseteq \mathcal{F}_{inert}$ *is a set of inertial fluents which hold in the institution's initial state and persist from one state to the next if they are not terminated.*

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 [72, 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 EU–CFR institution includes:

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

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

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

According to the European Court of Justice's (ECJ) judgement [70] on the EU–DRD, with respect to the EU–CFR, storing communications metadata counts–as storing personal data. If an agent's, Ada's, metadata is stored in any context (the empty set) then the event of storing the agent's, 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.

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

$$\begin{aligned}&\mathcal{G}^{cfr}(\{\neg \mathit{consentedDataProcessing(ada, isp)}\}, \mathit{storeData(isp, ada, personal)}) \ni\\ &\quad \mathit{nonConsensualDataProcessing(ada)}\end{aligned}$$

The EU–DRD [71, 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.

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

According to the ECJ's interpretation of the EU–DRD [70] 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 EU–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.

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

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 which 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, such 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* $\forall \langle i, j \rangle \in R : i < j$*.*

According to these definitions, we can represent the three main aspects of multi–level governance we focus on in this chapter. Firstly, regulations which 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 a multi–level governance institution in two ways. Firstly with constitutive rules which state a concrete concept counts–as a more abstract concept in a particular context. These abstracting constitutive rules are represented as the event generation function – a concrete event counts–as a more abstract event in a particular social context; the state consequence function – a concrete event counts–as initiating/terminating a fluent that abstractly describes the social reality in a particular context; and the fluent derivation function – a concrete fluent counts–as a more abstract fluent in a particular social context. The second method of abstraction is implicit abstraction of concrete normative fluents regulating concrete concepts to more abstract normative fluents regulating abstract concepts. Normative fluent abstraction is defined semantically, based on constitutive rules between concrete and abstract concepts, and requires no explicit representation.

3.3. CASE STUDY FORMALISATION

To demonstrate the representation in full, our case study and its formalisation in the formal framework are subsequently summarised. We look at three legal institutions. These institutions operate at different levels of abstraction. Higher–levels interpret the regulatory effects of lower–levels in terms of more abstract normative fluents and concepts based on constitutive rules. Providing the constitutive rules to reinterpret and abstract regulations are the interpretations provided by courts. For example, the lower–level Data Retention Directive's interpretation by the European Court of Justice (ECJ) with respect to the higher–level Charter of Fundamental Rights. We view constitutive rules provided by court interpretations of a lower–level institution as being a part of the higher–level institution which governs the lower–level. For example, the Charter of Fundamental Rights comprises its rules and regulations set out in writing and also the interpretations of the Data Retention Directive's concepts provided by the ECJ.

Formally, the multi–level governance institution is the tuple $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ where the institutions are $\mathcal{T} = \langle \mathcal{I}^{drr}, \mathcal{I}^{drd}, \mathcal{I}^{cfr} \rangle$ and the governance relation is $R = \{\langle drr, drd \rangle, \langle drd, cfr \rangle\}$. Each of the institutions are formalised in tables 3.1, 3.3 and 3.4 where for brevity we leave out empowerment fluents and upper–case terms represent variables acting as shorthand for their ground instantiations. For clarity, we use labels rather than indexes for each institution. Each institution is described and formalised as follows.

The EU Charter of Fundamental Rights [72] (EU–CFR, Formalised in Table 3.1), a third–level institution which through the most abstract regulations governs the Data Retention Directive. The specific fragments of legislation we look at are:

- Article 7 – "*Everyone has the right to respect for his or her private and family life, home and communications.*" (rule CFRC.2, and CFRIS.1)
- Article 8 – "*Everyone has the right to the protection of personal data concerning him or her.*" (rule CFRC.3, and CFRIS.1) and "*Compliance with these rules shall be subject to control by an independent authority.*" (rule CFRC.5, and CFRIS.1)

Partly based on the European Court of Justice's interpretation of the Data Retention Directive [70] we also take the following legal interpretations as holding in the EU–CFR:

- Storing communications content or metadata counts–as storing personal data (rules CFRG.1 and CFRG.2).
- Storing personal data if the person it concerns has not consented counts–as non–consensual data processing. (rule CFRG.3).
- There is unfair data processing whenever there is an obligation for there to be non–consensual data processing (rule CFRD.1).
- Data is unprotected if there is an obligation to store data in the context that the data is not anonymised (rule CFRD.4).
- Privacy is disrespected when there is an obligation to store personal data (rule CFRD.8).

It is interesting to note that the Charter of Fundamental Rights regulates regulations but in an implicit manner, that is, nowhere does it state that the regulations regulate other regulations. That is, not obliging and prohibiting obligations or prohibitions, but rather obliging and prohibiting abstract concepts which can represent obligations and prohibitions. For example, rather than prohibiting obligations for data to be processed unfairly, the EU–CFR prohibits unfair processing itself. In this chapter, we assume that whenever the EU–CFR governs an abstract concept, such as unfair data processing, then it views obliging that abstract concept as counting–as that abstract concept. For example, processing data unfairly has many interpretations, one of which is obliging data is processed unfairly. Following this idea, if the EU–CFR prohibits processing data unfairly, then an obligation to process data unfairly is reduced to the abstract concept of unfair data processing itself, which violates the EU–CFR's prohibition. Through transitivity, it follows that an obligation to oblige unfair data processing is also reduced in the same way resulting in a violation of a prohibition to process data unfairly. Interpreting abstract normative fluents as the abstract concepts they oblige/prohibit is realised with fluent derivation rules:

- Obliging unfair data processing counts–as unfair data processing (rule CFRD.2).
- Obliging data is anonymised counts–as data being anonymised (rule CFRD.3).
- Obliging data is unprotected counts–as data being unprotected (rule CFRD.5).
- Obliging privacy is disrespected counts–as privacy being disrespected (rule CFRD.7).
- Obliging data is processed counts–as data being processed (rule CFRD.9).

Table 3.1: Charter of Fundamental Rights of the European Union Formalisation

$$\mathcal{G}^{cfr}(\emptyset, \mathit{storeData(CommProv0,\ Agent0,\ content)}) \ni \mathit{storeData(CommProv0,\ Agent0,\ personal)} \tag{CFRG.1}$$

$$\mathcal{G}^{cfr}(\emptyset, \mathit{storeData(CommProv0,\ Agent0,\ metadata)}) \ni \mathit{storeData(CommProv0,\ Agent0,\ personal)} \tag{CFRG.2}$$

$$\mathcal{G}^{cfr}(\{\neg \mathit{consentedDataProcessing(Agent0,\ CommSerProv)}\}, \mathit{storeData(CommProv0,\ Agent0,\ personal)}) \ni \mathit{nonConsensualDataProcessing(Agent0)} \tag{CFRG.3}$$

$$\mathcal{G}^{cfr}(\{\neg \mathit{jurisdiction(Location0,\ eu)}\}, \mathit{storeDataAt(CommProv0,\ Agent0,\ Location0)}) \ni \mathit{storeDataOutsideEU} \tag{CFRG.4}$$

$$\mathcal{C}^{cfr\uparrow}(\emptyset, \mathit{consent(CommSerProv0,\ Agent0)} \ni \mathit{consentedDataProcessing(Agent0,\ CommSerProv0)} \tag{CFRC.1}$$

$$\mathcal{C}^{cfr\uparrow}(\emptyset, \mathit{viol(pro(privacyDisrespected(Agent),\ never))}) \ni \mathit{pro(privacyDisrespected(Agent),\ never)} \tag{CFRC.2}$$

$$\mathcal{C}^{cfr\uparrow}(\emptyset, \mathit{viol(pro(dataUnprotected(Agent0,\ personal),\ never))}) \ni \mathit{pro(dataUnprotected(Agent0,\ personal),\ never)} \tag{CFRC.3}$$

$$\mathcal{C}^{cfr\uparrow}(\emptyset, \mathit{viol(pro(unfairDataProcessing(Agent0),\ never))}) \ni \mathit{pro(unfairDataProcessing(Agent0),\ never)} \tag{CFRC.4}$$

$$\mathcal{C}^{cfr\uparrow}(\emptyset, \mathit{viol(pro(uncontrolByIndepAuth,\ never))}) \ni \mathit{pro(uncontrolByIndepAuth,\ never)} \tag{CFRC.5}$$

$$\mathcal{D}^{cfr}(\emptyset, \mathit{obl(nonConsensualDataProcessing(Agent0),\ now)}) \ni \mathit{unfairDataProcessing(Agent0)} \tag{CFRD.1}$$

$$\mathcal{D}^{cfr}(\emptyset, \mathit{obl(unfairDataProcessing(Agent0),\ now)}) \ni \mathit{unfairDataProcessing(Agent0)} \tag{CFRD.2}$$

$$\mathcal{D}^{cfr}(\emptyset, \mathit{obl(dataAnonymised(CommSerProv0,\ Agent0),\ now))}) \ni \mathit{dataAnonymised(CommSerProv0,\ Agent0)} \tag{CFRD.3}$$

$$\mathcal{D}^{cfr}(\{\neg dataAnonymised(CommSerProv0,\ Agent0)\},\ obl(storeData(CommSerProv0,\ Agent0,\ Data0),\ now) \ni obl(dataUnprotected(Agent0,\ CommSerProv0),\ now)) \tag{CFRD.4}$$

$$\mathcal{D}^{cfr}(\emptyset, obl(dataUnprotected(Agent0,\ CommSerProv0),\ now) \ni dataUnprotected(Agent0,\ CommSerProv0) \tag{CFRD.5}$$

$$\mathcal{D}^{cfr}(\emptyset, \{obl(storeData(CommSerProv0,\ Agent0,\ personal),\ now)\}) \ni privacyDisrespected(Agent0) \tag{CFRD.6}$$

$$\mathcal{D}^{cfr}(\emptyset, \{obl(privacyDisrespected(Agent0),\ now)\}) \ni privacyDisrespected(Agent0) \tag{CFRD.7}$$

$$\mathcal{D}^{cfr}(\emptyset, obl(storeData(CommSerProv0,\ Agent0,\ personal),\ now)) \ni dataProcessed \tag{CFRD.8}$$

$$\mathcal{D}^{cfr}(\emptyset, obl(dataProcessed,\ now)) \ni dataProcessed \tag{CFRD.9}$$

$$\mathcal{D}^{cfr}(\{\neg pro(storeDataOutsideEU,\ never)\}, dataProcessed) \ni uncontrolByIndepAuth \tag{CFRD.10}$$

$$\Delta^{cfr} = \{pro(privacyDisrespected(Agent),\ never),\ pro(dataUnprotected(Agent0,\ personal),\ never),\ pro(unfairDataProcessing(Agent0),\ never),\ pro(uncontrolByIndepAuth,\ never), is(charles,\ lawEnforcement)\} \tag{CFRIS.1}$$

The EU Data Retention Directive [71] (EU–DRD) is a second–level institution. It consists of less abstract regulations than the Charter of Fundamental Rights which governs it. On the other hand, it has more abstract regulations than the institution it governs, the UK's implementation of the EU–DRD called the UK's Data Retention Regulations. Whilst both the EU–CFR and the EU–DRD govern regulations, a directive by nature is written specifically for that purpose as a regulation regulating legal institution. Consequently, for the EU–DRD we adopt an explicit form for regulations governing regulations.

The specific fragments of the Data Retention Directive that we formalise are:

– Article 5 – "*Member states shall ensure that the following categories of data are retained under this Directive:" (paraphrased) the data necessary to trace and identify the source, destination, date, time and duration, and type of a communication. Also, the users' communication equipment and its location* (rule DRDC.1, and DRDD.2) and "*No data revealing the content of the communication may be retained pursuant to this Directive*" (DRDD.2).

– "*Member States shall ensure that the categories of data specified in Article 5 are retained for periods of not less than six months and not more than two years from the date of the communication*" (rule DRDC.2).

– Article 8 – "*Member States shall ensure that the data specified in Article 5 are retained in accordance with this Directive in such a way that the data retained and any other necessary information relating to such data can be transmitted upon request to the competent authorities without undue delay*" (rule DRDC.3).
– Article 13 (clause 1) – "[...]*sanctions are fully implemented with respect to the processing of data under this Directive.*" (rule DRDC.4).

We also take the following legal interpretations as holding in the EU–DRD:

– Something happening after one month counts–as undue delay (rule DRDG.1).
– Paying a fine counts–as being punished (rule DRDG.2).
– Data retention is ensured for a period of between six and twenty–four months whenever there is a prohibition on deleting the data before 12 months and an obligation to delete it after 13 months (rules DRDD.1 and DRDD.2).

Table 3.3: EU Data Retention Directive Formalisation

$$\mathcal{G}^{drd}(\emptyset, time(m1)) \ni undue\_delay \tag{DRDG.1}$$

$$\mathcal{G}^{drd}(\emptyset, payFine(CommSerProv0, Agent0)) \ni punish(CommSerProv0) \tag{DRDG.2}$$

$$\begin{aligned}&\mathcal{C}^{drd\uparrow}(\emptyset, useElectronicCommunication(Agent0, CommServProv0)) \ni\\ &obl(obl(storeData(CommServProv0, Agent0, metadata), now), now)\end{aligned} \tag{DRDC.1}$$

$$\begin{aligned}&\mathcal{C}^{drd\uparrow}(\emptyset, storeData(CommSerProv0, Agent0, Data0)) \ni\\ &obl(ensure\_data\_retention\_period(Agent0,\\ &\quad CommSerProv0, Data0, m6, m24), now)\end{aligned} \tag{DRDC.2}$$

$$\begin{aligned}&\mathcal{C}^{drd\uparrow}(\{is(Agent0, lawEnforcement)\},\\ &\quad requestData(Agent0, CommSerProv0, Agent1)) \ni\\ &obl(obl(provideData(CommSerProv0, Agent0, Agent1), undue\_delay), now)\end{aligned} \tag{DRDC.3}$$

$$\begin{aligned}&\mathcal{C}^{drd\uparrow}(\emptyset, viol(obl(provideData(Co, Ag0, Ag1), undue\_delay))) \ni\\ &obl(obl(punish(CommServProv0), time(m6)), now)\end{aligned} \tag{DRDC.4}$$

$$\begin{aligned}&\mathcal{D}^{drd}(\{pro(deleteData(CommSerProv0, Agent0, Data0), time(m12))\},\\ &obl(deleteData(CommSerProv0, Agent0, Data0), time(m13))) \ni\\ &ensure\_data\_retention\_period(Agent0, CommSerProv0, Data0, m6, m24)\end{aligned} \tag{DRDD.1}$$

$$\begin{aligned}&\mathcal{D}^{drd}(\{obl(deleteData(CommSerProv0, Agent0, Data0), time(m13))\},\\ &pro(deleteData(CommSerProv0, Agent0, Data0), time(m12))) \ni\\ &ensure\_data\_retention\_period(Agent0, CommSerProv0, Data0, m6, m24)\end{aligned} \tag{DRDD.2}$$

$$\begin{aligned}\Delta^{drd} = \{&obl(pro(storeData(CommProv0, Agent0, content), never), now),\\ &is(charles, lawEnforcement)\}\end{aligned} \tag{DRDD.2}$$

The UK's Data Retention Regulations [226] (UKDRR) is a first–level institution which governs communications service providers in the United Kingdom. The fragments of the legislation we formalise are:

– Article 4 – "*It is the duty of a public communications provider to retain the communications data specified in the following provisions of the Schedule to these Regulations*" obliges metadata is stored and prohibits content data is stored (rule DRRC.1, and DRRIS.1)
– Article 5 – "*The data specified in the Schedule to these Regulations must be retained by the public communications provider for a period of 12 months from the date of the communication in question*" (rules DRRC.2 and DRRC.3).
– Article 8 – "*The data retained in pursuance of these Regulations must be retained in such a way that it can be transmitted without undue delay in response to requests*" (rule DRRC.4).
– We assume the regulations are enforced with fines (e.g. rule DRRC.4).

Table 3.4: UK Data Retention Regulations Formalisation

$$\begin{array}{l}\mathcal{C}^{drr\uparrow}(\emptyset, \mathit{useElectronicCommunication(Agent0,\ CommProv0)}) \ni \\ \mathit{obl(storeData(CommProv0,\ Agent0,\ metadata),\ now)}\end{array} \tag{DRRC.1}$$

$$\begin{array}{l}\mathcal{C}^{drr\uparrow}(\emptyset, \mathit{storeData(CommProv0,\ Agent0,\ metadata)})) \ni \\ \mathit{pro(deleteData(CommProv0,\ Agent0,\ metadata),\ time(m12))}\end{array} \tag{DRRC.2}$$

$$\begin{array}{l}\mathcal{C}^{drr\uparrow}(\emptyset, \mathit{storeData(CommProv0,\ Agent0,\ metadata)}) \ni \\ \mathit{obl(deleteData(CommProv0,\ Agent0),\ time(m13))}\end{array} \tag{DRRC.3}$$

$$\begin{array}{l}\mathcal{C}^{drr\uparrow}(\{\mathit{is(Agent0,\ lawEnforcement)}\}, \\ \mathit{requestData(Agent0,\ CommProv0,\ Agent1,\ metadata)}) \ni \\ \mathit{obl(provideData(Agent0,\ CommProv0,\ Agent1,\ metadata),\ time(m1))}\end{array} \tag{DRRC.4}$$

$$\begin{array}{l}\mathcal{C}^{drr\uparrow}(\emptyset, \mathit{viol(obl(provideData(CommProv0,\ Agent0,\ Agent1),\ time(Length0))))}) \ni \\ \mathit{obl(payFine(CommProv0,\ secrOfState),\ time(m6))}\end{array} \tag{DRRC.5}$$

$$\begin{array}{l}\Delta^{drr} = \{\mathit{pro(storeData(CommProv0,\ Agent0,\ content),\ never)}, \\ \qquad \mathit{is(charles,\ lawEnforcement)}\}\end{array} \tag{DRRIS.1}$$

3.4. 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 3.4 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.

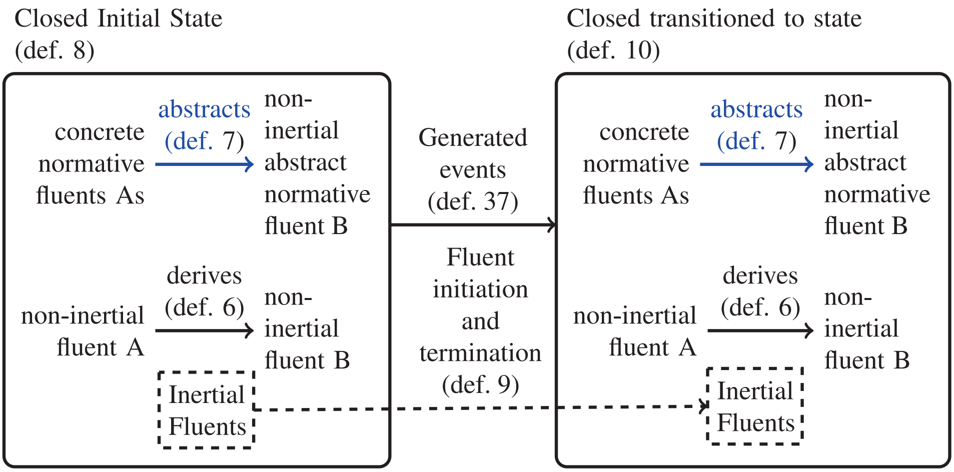

Figure 3.4: An overview of the semantics, depicting the transition from the initial state to the next state and state closure.

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 $j - \langle E^1_j, ..., E^n_j \rangle$.

STATE CONDITIONS

Institutions in a multi–level governance specification contain rules which 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 meaning whilst being syntactically different – this is not unusual, in 'Standard Deontic Logic' [234] forbidden X is defined as obliged not X and likewise for much subsequent work.

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 discharge and violation coincide. Referring again to Figure 3.3, an event/fluent a is obliged to occur/hold before or at the same time as some d by the obligation fluent *obl(a, d)* and is prohibited to occur/hold strictly before d by the prohibition fluent $pro(a, d)$. Given two normative fluents *obl(a, d)* and $pro(a', d')$ where a is equivalent to d' and d is 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.[1]

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:*

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

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 $\langle E^1_j, ..., E^n_j \rangle$. 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:*

$$now \in E' \tag{D37.1}$$

$$E^i \subseteq E' \tag{D37.2}$$

$$\exists X \in \mathcal{X}^i, e \in E', e' \in \mathcal{G}^i(X,e) : S^i \models X \wedge S^i \models pow(e') \Rightarrow e' \in E' \tag{D37.3}$$

$$S^i \models obl(a,d) \wedge (a \in E' \vee S^i \models a) \Rightarrow disch(obl(a,d)) \in E' \tag{D37.4}$$

$$S^i \models obl(a,d) \wedge (d \in E' \vee S^i \models d) \wedge disch(obl(a,d)) \notin E' \Rightarrow viol(obl(a,\ d)) \in E' \tag{D37.5}$$

$$S^i \models pro(a,d) \wedge (d \in E' \vee S^i \models d) \Rightarrow disch(pro(a,d)) \in E' \tag{D37.6}$$

$$S^i \models pro(a,d) \wedge (a \in E' \vee S^i \models a) \wedge disch(pro(a,d)) \notin E' \Rightarrow viol(pro(a,d)) \in E' \tag{D37.7}$$

$$\exists \langle h,i \rangle \in R, e \in \mathcal{E}^h_{norm} \cap \mathcal{E}^i_{norm} \Rightarrow e \in E' \tag{D37.8}$$

In more detail:

- D37.1 – the event of *now* always occurs.

[1] 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.

- D37.2 – events that have already occurred still occur (monotonicity).
- D37.3 – 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.
- D37.4 to D37.7 – 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.
- D37.8 – norm compliance events occurring in lower level institutions linked to this institution, also occur in this institution.

Note that GR^i is increasingly monotonic and a well–defined 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).

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 which 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:*

$$S^i \subseteq S' \tag{D6.1}$$

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

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.

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 $\mathcal{D}(\{\neg B\}, A) \ni B$

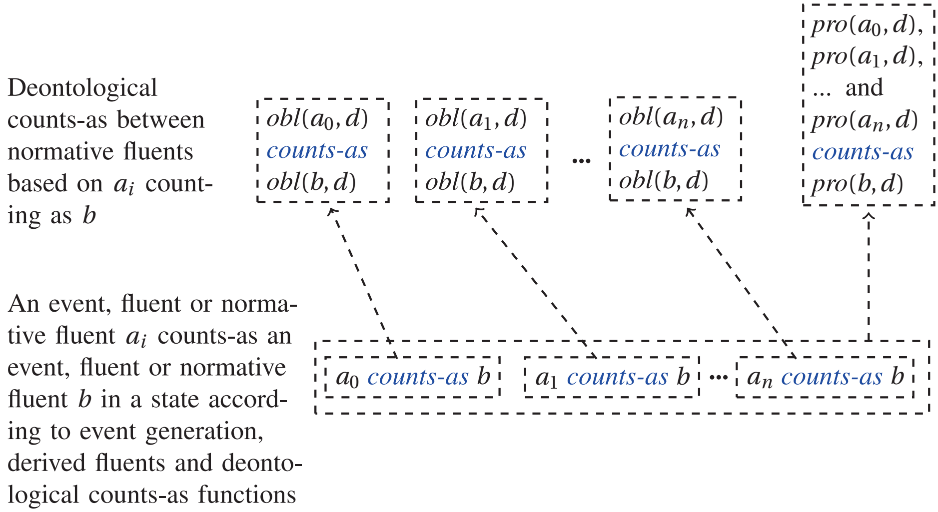

Figure 3.5: Overview for deontological counts–as semantics between concrete and abstract normative fluents, based on counts–as relations between the elements they prescribe holding in a context entailed by a single state.

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". The occurrence of these issues indicates an institution design problem, which should be resolved by the institution designer.

ABSTRACTING NORMATIVE FLUENTS

This section presents a semantics for abstracting concrete normative fluents. The basic idea, depicted in Figure 3.5, 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. In other words, counts–as relations between concrete and abstract normative fluents are based on other counts–as relations between the concrete and abstract concepts they prescribe. 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 types of counts–as rules (referring again to figure 3.5). 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 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 complying with the concrete normative fluents (i.e. discharging or not violating) by causing an event to occur or fluent to hold counts–as a certain institutional event or fluent to hold which *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 3.5, 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 inappropriate to base counts–as between prohibitions on, 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* defining all ways an abstract institutional concept can occur/hold when more concrete concepts occur/hold. Since the institutional events/fluents are *ascribed* by an institution's counts–as relations, if none of $a_0,...,a_n$ occur/hold then b is not ascribed and therefore does not occur/hold. 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).

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 which counts–as an obligation for b to occur before z, which is 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 EU–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:

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

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

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

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.

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

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.

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

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:

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

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 [70] 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
$\mathcal{D}^{cfr}(\emptyset, obl(\textit{nonConsensualDataProcessing(ada)}, \textit{now})) \ni \textit{unfairDataProcessing(ada)}$. 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:

$$\langle\{\textit{obl(obl(nonConsensualDataProcessing(ada)), now), now)}\},$$
$$\textit{obl(unfairDataProcessing(ada), now)}\rangle \in DC^i(S^i)$$

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: 1. according to the fluent derivation function obliging non–consensual data processing counts–as unfair data processing, thus 2. obliging an obligation to process data non–consensually counts–as obliging unfair data processing. 3. The EU–CFR considers whether data is processed unfairly or obliged to be processed unfairly as irrelevant, both are viewed as unfair data processing. 4. Thus, an obligation to process data unfairly counts–as unfair data processing according to the fluent derivation function, $\mathcal{D}^{cfr}(\emptyset, obl(unfairDataProcessing(ada)), now)) \ni \textit{unfairDataProcessing(ada)}$. 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 (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:

$$\langle\{\textit{obl(obl(unfairDataProcessing(ada), now), now)}\},$$
$$\textit{unfairDataProcessing(ada)}\rangle \in DC^i(S^i)$$
$$\langle\{\textit{obl(obl(obl(unfairDataProcessing(ada), now), now), now)}\},$$
$$\textit{unfairDataProcessing(ada)}\rangle \in DC^i(S^i)$$
$$\ldots$$

Following this discussion, we formally define deontological counts–as, based on counts–as relations which 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 $\mathcal{A}^i$ which form the 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 also 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:*

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

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

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

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

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

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:

$$S^i \subseteq S' \qquad \text{(D7.6)}$$

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

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 which 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 $\overline{DC}^i(S) = \perp$ then we have detected an institutional design problem for the institution designer to rectify.

MODELS

In this section we define a *model* which describes how each institution in a multi–level governance institution evolves from one state to the next, driven by observable events which 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 the previous operations according to the following description. An institution starts at an initial state which 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 $INIT^i : \Sigma^i \times 2^{\mathcal{E}^i} \to 2^{\mathcal{F}^i}$ and $TERM^i : \Sigma^i \times 2^{\mathcal{E}^i} \to 2^{\mathcal{F}^i}$:*

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

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

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

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 the set of terminated inertial fluents includes any inertial fluents terminated according to the institution's consequence function. Condition D9.2.2 states any discharged or violated inertial (concrete) normative fluents are also terminated, meaning discharged/violated normative fluents do not persist by default[2].

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. It is formally defined as and subsequently described:

Definition 10. ***State Transitions*** *The state transition operation $TR^i : \Sigma^i \times 2^{\mathcal{E}^i} \to \Sigma^i$ 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.

[2]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 still usually 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 [98].

- 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.

- 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 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 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 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. $\langle S_j^1, ..., S_j^n \rangle$. The tuple M is a model of $\mathcal{ML}$ w.r.t. ctr if and only if:*

$$\forall i \in [1, n] : S_0^i \textit{ is the initial state of each institution } \mathcal{I}^i \textit{ w.r.t. } \langle S_0^1, ..., S_0^n \rangle \quad \text{(D13.1)}$$

$$\forall i \in [1, n], \forall j \in [0, k] : E_j^i = GR^i(S_j^i, \{se_j^i\}) \quad \text{(D13.2)}$$

$$\forall i \in [1, n], \forall j \in [0, k] : S_{j+1}^i = TR^i(S_j^i, E_j^i) \quad \text{(D13.3)}$$

- 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.

COMPLIANCE MONITORING

A multi–level governance institution model monitors the compliance of other institutions' 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 which 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 $\forall i \in [n] : M^i = \langle S^i, E^i \rangle, S^i = \langle S_0^i, ..., S_{k+1}^i \rangle, E^i = \langle E_0^i, ..., E_k^i \rangle$. 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:*

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

Non–compliance is found if the set of violation events is not the empty set. For an institution governing a society this implies 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 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.

3.5. RELATED WORK

In this chapter we presented a formal framework for reasoning about multi–level governance. Specifically, determining compliance where higher–level institutions which impose abstract regulations governing lower–level institutions which impose concrete regulations. The purpose of this chapter is to contribute a rigorous formalisation, with particular attention paid to the philosophical aspects. Closely related works contribute formalisms for reasoning about hierarchical governance and abstract regulations.

3.5.1. HIERARCHICAL GOVERNANCE

There appears to be little on hierarchical governance and the regulation of regulations. In [159] Lopez and Luck propose a framework for reasoning about norms governing agents, created from a top–down governance perspective. Their framework, based on the Z specification language, gives a precise specification language of a normative system/institution rather than like ours a specification language and operationalisation (semantics). Like our framework, theirs offers similar expressivity with temporal norms, rewards, punishments, etcetera. In particular Lopez and Luck formalize what they call *legislative norms* which are special norms filling the role of, viewing norms as being dynamic and subject to creation, deletion and modification, governing the act of norm changes. This still presents a substantial difference to the method of hierarchical governance and regulation governing regulations we propose, since we use higher–order norms that govern the *outcome* of other norms from which (non–)compliance is determined (typically pre–runtime). Lopez and Luck's legislative norms on the other hand govern the changes to the norms (rules) themselves.

Boella and van der Torre [26] offer a conceptual formalisation of *hierarchical* normative systems in Input/Output Logic (a logic aimed at studying conditional norms [165]). Like our formalisation, they formalise governance hierarchies, but their formalised concepts have nuanced differences with ours. One of their foci is the role of permissions in hierarchical normative systems. They look at permissions from two perspectives: firstly permissions in static normative systems where authorities do not change norms, secondly permissions in dynamic normative systems where authorities are liable to change norms. In the first case of static normative systems, 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. Conversely, a higher-level authority issuing an obligation is essentially preventing a lower-level authority from issuing a permission that derogates the obligation. In the second case of dynamic normative systems, permissions issued by higher-level authorities act to *enable* lower-level authorities to *change* the normative system in some way, such as permitting a norm to be implemented. The same goes for obligations and prohibitions, a higher-level authority may prohibit a lower-level from implementing a permission that derogates other norms. In contrast, we do not look at derogation, similar concepts would require a kind of defeasibility, which as discussed in the relation to defeasible logic is left for future work. In terms of norms that enable/disable changes to a *dynamic* normative system, we do not look at dynamics of norms in this chapter, but later in Chapter 6 we formalise counts-as conditionals that regulate norm change in a *temporal setting*.

Subsequent work by Boella and van der Torre [29] formalises higher-order norms in the context of security policies. A security governance hierarchy involves global policy makers obliging/permitting/prohibiting local policy makers to/from obliging/permitting/prohibiting individual agents to/from sharing information and knowledge. In the formalisation agents impose norms by ascribing violations to the actions of other named agents. For example, if agent A prohibits agent B from doing x then agent A ascribes, according to an internal rule, a violation on B for doing x. Hence, the general approach is to adopt Anderson's [9] reduction of deontic logic to alethic modal logic, where the modalities are replaced by rules with violation as the necessary consequent of prohibited actions/states of affairs. In order to impose higher-order norms, agents ascribe nested violations. So for example agent C prohibiting agent A from prohibiting B to do x, is reduced to an ascription by agent C to the action of agent A ascribing a violation to B doing x (a violation violates a norm). The formalisation is in contrast to ours. A higher-order norm according to Boella and van der Torre's reduction to ascribed violations, is violated only when the norm it prescribes is violated. Consequently, if there are no violations of first-order norms then there are no violations of higher-order norms, owing to one focus being on the enforcement of norms which requires violations being ascribed to agents. In contrast, higher-order norms in our formalisation are discharged/violated by the mere imposition of obligations and prohibitions by lower-level institutions, although we show examples where violations themselves are obliged/prohibited, moreover we specifically address the different level of abstractions norms operate at.

Aside from these different approaches and foci, Boella and van der Torre's formalisation in [29] provides an argument for von Wright's "transmission of will" principle requiring norms of all orders are enforced. As we discussed briefly in the background section (specifically 2.2.3), it is shown in formalised scenarios that von Wright's 'transmission of will' [236] does not merely involve an obligation to oblige being fulfilled, but also the lower-order obligation being enforced. Moreover, higher-order norms should be enforced, or else lower authority agents acting in their own self interests may (not) impose certain norms that contravene higher-order norms. Central to demonstrating these two enforcement requirements is the use of violations being ascribed to agents' actions (including the action of ascribing violations). In our formalisation, it is grammatical for higher-order norms to be enforced but in our context of institutions governing institutions higher-order norm enforcement is meaningless. That is, in our formalisation violation occurs as a part of a pre-runtime check (e.g. to check whether national legislation implements an EU directive before a particular deadline) and hence it makes no sense to enforce higher-order norms when violations occur. Rather we assume rewards/punishments for (non-)compliant institution designs are imposed by some kind of external process or logic after a pre-runtime check has determined (non-)compliance of the institution design as a whole. In comparison, since Boella and van der Torre are dealing with agents imposing norms on other agents, enforcement of higher-order norms is meaningful. As for transmission of will requiring obliged lower-order norms to be enforced, we do not have a general formalisation of this requirement. However, we do formalise a specific component the EU Data Retention Directive, which that when obliged obligations are violated then it is obliged there is an obligation for an offender to be punished. In summary, our work agrees with Boella and van der Torre's that both higher and lower-order norms should be enforced, but in the former case we assume an external process handles enforcement and in the latter case we formalise specific scenarios but provide no generalised reasoning for the transmission of will.

3.5.2. ABSTRACTING NORMS

As we discussed in the background section, there has already been a reduction of Standard Deontic Logic [234] to a logic of counts–as conditionals representing evaluative norms [9] (as studied in [106, 112]). For example, 'B counts–as a violation in a context C'. Following this idea, Aldewereld et al. [4] propose implemented reasoning for concretising abstract norms. This is 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, applies to what differentiates it from our work. 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 mechanism for reasoning about abstract institutional concepts is also proposed by Grossi et al. [108]. Unlike in our work, Grossi et al. do not propose abstraction of norms themselves (in fact, they propose concretising concepts), since normative reasoning is not a part of their proposal. 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 not support nesting of deontic modalities and therefore higher–order norms. They also offer an alternative path to normative reasoning. Namely, assigning different descriptive concepts a 'role' of ideal thereby designating their normativity together with axioms for normative reasoning. The idea being, norms can then be nested and normative statements concretised. However, this part of the proposal is not formalised. Furthermore, our work still differs in that we are interested in abstracting rather than concretising norms in a temporal setting.

In comparison a series of papers by Fornara and Colombetti [79], Fornara [77] and Fornara, Okouya and Colombetti [80] 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 using an external process which 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. [49] 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 primitives 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.

To summarise, work proposing ways to reason about abstract and concrete norms or using techniques that can be extended to do so is quite different from our own. 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. Rather, 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.

3.6. DISCUSSION

In this chapter, we answered the question "*How can we formalise compliance in multi–level governance?*" with a formal framework. The general answer being that a lower–level institution is compliant in multi–level governance when the effects of its regulations (obligations and prohibitions) are compliant with higher–order regulations at higher–levels of governance. In particular, taking into account the fact that lower–level institutions operate at a different levels of abstraction to higher–levels. Accommodating for this fact, compliance is determined based on the abstract meaning of concrete regulations. We adopted the usual notion of *counts–as* between concrete and abstract institutional concepts. Based on the *counts–as* ontological primitive we semantically defined the abstraction of norms, first and higher–order, with the notion of *deontological counts–as*. Under the semantics, concrete regulations from lower–level institutions are abstracted to the same level of abstraction in higher–level institutions. In their abstract form, the effects of concrete regulations (obligations and prohibitions) are determined for compliance with abstract regulations in higher–level institutions. It is important to note that the abstract meaning of concrete regulations is context–sensitive. Therefore, whether concrete regulations are compliant depends on the contexts they are applied in. To summarise, institutions are compliant in multi–level governance if in different social contexts the abstract meaning of their concrete regulations is compliant with abstract higher–order norms at higher–levels of governance.

The strength of the semantics is in formalising the abstraction of concrete norm effects (obligations and prohibitions), potentially abstracting higher–order concrete norm effects. By paying particular attention to the norm abstraction semantics, with lengthy discussion and the application to our case study, we argued the semantics correctly abstract concrete normative fluents. However, in general the semantics proposed in this section are not without their weaknesses. In particular, as we discussed multiple times, many of the operations are undefined for cases where an inconsistent institution design causes contradiction. Arguably, the fault is with the institution design and not the semantics. On the other hand, inconsistent information is a fact of life. In this chapter, we do not propose an inconsistency tolerant semantics. But, it is important for future work to address this weakness. Particular attention should be paid to resolving inconsistencies arising under the novel semantics provided by deontic abstraction (whereas current work in the literature deals with inconsistency occurring according to a semantics where abstraction is not modelled).

4

Computational Multi–level Governance Compliance Checking

Science is knowledge which we understand so well that we can teach it to a computer; and if we don't fully understand something, it is an art to deal with it.

Donald Knuth

This chapter is based on the following papers:
King, T. C., Li, T., De Vos, M., Dignum, V., Jonker, C. M., Padget, J., & van Riemsdijk, M. B. (2015, May). Automated Multi–level Governance Compliance Checking. *Journal of Autonomous Agents and Multiagent Systems (JAAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems. (**In Submission**)
Which extends the following paper:
King, T. C., Li, T., De Vos, M., Dignum, V., Jonker, C. M., Padget, J., & Riemsdijk, M. B. Van. (2015). A Framework for Institutions Governing Institutions. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015) (pp. 473–481). Istanbul, Turkey: International Foundation for Autonomous Agents and Multiagent Systems. [144]

This chapter makes the following contributions:

- A computational framework for representing institutions in multi–level governance and automatically detecting compliance in multi–level governance.

The previous chapter gave a formal definition for when institutions are compliant in multi–level governance. Special attention was paid to the increasing abstraction of regulations at higher–levels of governance that govern the relatively concrete regulations of lower–levels of governance. Hence, one objective of this dissertation was achieved. Namely, to formalise the informal notions of compliance in multi–level governance.

In this chapter the focus moves from formalising compliance to automated compliance checking with a computational framework that corresponds to the formal framework. The approach we take is to represent institutions and their semantics as Answer–Set Programming (ASP) programs. By using ASP we can produce models of a multi–level governance institution for a trace of events. Thus we can automatically determine compliance by inspecting the generated models for non–compliant lower–level institutions. This follows the approach of InstAL [42, 43], which provides a representation in Answer–Set Programming for individual institutions governing MAS. However, we need to provide a novel representation for institutions governing other institutions in multi–level governance institutions. Thus, we need to provide a corresponding Answer–Set Program for reasoning about abstract norms governing more concrete norms.

To this end, this chapter contributes the following novel elements for practical reasoning about compliance in multi–level governance:

- Regulations governing regulations computational reasoning. By taking a formal representation of multi–level governance institutions and producing, through a transformation, a set of answer–set programming rules. These rules capture norms governing norms' representation and reasoning. The previous chapter contributed a representation and semantics for norms governing norms. This chapter contributes the implementation, such that in our computational framework when a norm at a higher–level of governance is violated this indicates that the regulatory effects (imposed obligations and prohibitions) of lower–levels are non–compliant.

- Norm abstraction reasoning. By taking a formal representation of multi–level governance institutions and producing through a transformation a set of answer–set programming rules. These rules capture the semantics for abstracting concrete norms based on whether the concrete prescribed concepts count–as more abstract concepts.

- A computational framework based on answer–set programming, including the two aforementioned representations, which *corresponds* to the formal framework. That is, firstly an answer–set program such that for every answer–set program it produces there is an equivalent model in the formal framework for the same multi–level governance institution (soundness). Secondly, an answer–set program such that for every model in the formal framework there is an equivalent answer–set produced by the answer–set program for the same multi–level governance institution (completeness).

In addition to these contributions, the following application is contributed:

- An implementation of the aforementioned transformations as a parser and compiler, extending the InstAL implementation. The implementation provides a high–level representation language for representing multi–level governance institutions and outputs an answer–set programming for checking compliance in those institutions. In this chapter the contributions are applied to an extensive formalisation of EU law for assessing compliance in our running case study.

In the rest of this chapter we begin by introducing Answer–Set Programming (ASP) in 4.1. We describe the overall approach we take to representing multi–level governance institutions in answer–set programming in 4.2. We give the representation in ASP in 4.3. We demonstrate the proposed computational framework we have implemented as a compiler which takes a high–level description of a multi–level governance institution. The output is an ASP program that operationalises a multi–level governance institution. We give an overview of the implementation and results of executing the resulting ASP program for our running case study in 4.4. To show that the computational framework provides a practical implementation of the formal framework we provide *soundness* and *completeness* properties between the two frameworks in section 4.5. The properties are proven in appendices. We finish with conclusions in section 4.7.

4.1. PRELIMINARIES – ANSWER SET PROGRAMMING

Answer–Set Programming is a non–monotonic logic–programming language [17, 93], 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) 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 [93].

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 lower–level institutions 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. [67, 91]). We briefly reintroduce the main definitions to give context for what follows, focussing on the syntax of the CLINGO [91] 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`. [1] 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 [17], the following program declares that a bird is an animal that lays eggs which 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).
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 [92]:

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).` $\}$,

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

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 [92]:

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 concise ways to represent useful constructs present in CLINGO [91]. Namely, *constraints*, *choice rules* and *conditional literals*. Constraints are a special type of rule of the form $:-\mathtt{b_1,...,b_n}$. 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 $\{\mathtt{a_1,...,a_n}\}:-\mathtt{b_1,...,b_n}$., meaning that any atom in $\mathtt{a_1,...,a_n}$ can arbitrarily be picked for inclusion in an answer–set if $\mathtt{b_1,...,b_n}$. is true. Aggregates are present in the body of rules and are of the form $\mathtt{l\{b_1;...;b_n\}u}$ where $\mathtt{l}$ and $\mathtt{u}$ are positive integers meaning that at least $\mathtt{l}$ and at most $\mathtt{u}$ elements of $\mathtt{b_1,...,b_n}$ must be true for the aggregate to be true. Omitting $\mathtt{l}$ or $\mathtt{u}$ removes the respective constraint. Finally, conditional literals can be contained in the body of a rule and are of the form $\mathtt{b_1 : b_2,...,b_n}$. 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 in a strict sense, however an aggregate of the form $\mathtt{l\{b_1 : b_2,...,b_n\}u}$ means that $\mathtt{b_1}$ is counted as being true when restricted to the domain of $\mathtt{b_2,...,b_n}$. Without variables this simply means that $\mathtt{b_1}$ is counted as true when $\mathtt{b_2,...,b_n}$ is true and hence operates similarly to conditional literals.

4.2. APPROACH

In this section we describe the overall approach we take to reasoning about multi–level governance institutions using answer–set programming. The general idea is to take a formal representation of a multi–level governance institutions. Then, to produce a corresponding representation of that formal description as a set of answer–set programming rules. Moreover, to also provide a corresponding set of answer–set programming rules which capture multi–level governance institution's formal semantics.

The ASP representation for multi–level governance institutions comprises facts representing the signature of each institution's formal representation (events and fluents) and their initial states. Furthermore, the ASP representation comprises rules, which represent each individual institution's rules described by their functions (event generation, consequence and fluent derivation).

Each ASP rule, representing an institution rule, comprises an antecedent and a consequent. The antecedent corresponds to the parameters the functions take. For the event generation and state consequence functions, the corresponding ASP 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 ASP rules’ antecedents comprise conditions on the state modelling the fluent A and 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. This 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 ASP as more general rules. For example, stating that if an inertial fluent is initiated then it holds until it is terminated. Finally, composite traces are mapped to a corresponding ASP representation as sets of facts, each stating that an event has been observed at a particular point in time.

The mapping to an ASP program makes use of the same common predicates used previously in work extending InstAL to settings with multiple institutions [154–156]. In turn these bear similarities to constructs used in the Event Calculus [150]. 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 and an event occurring, then an event *occurs*. This means events in function parameters correspond to `occurred/3` whilst state conditions correspond to sets containing positive and negative `holdsat/3` predicates. An empty state condition (the empty set) is always true and replaced with the special atom #`true`.

In more detail, the translation for representing a multi–level governance institution, comprising a set of ASP rules, is split into two parts containing the translation of the individual institutions and the translation of the links between them. For example in the Charter of Fundamental Rights, `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 event generation function is translated to rules. Each rule contains an `occurred/3` atom in the head representing the event that is caused to occur. Each rule's body comprises an `occurred/3` atom representing the *causal event*, and positive and negative `holdsat/3` atoms representing the rule's condition on the state. For example, the following rule states that non–consensual data processing occurs if Ada's personal data has been stored, but she has not consented and 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).
```

A consequence function is translated to ASP 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 Charter of Fundamental Rights institution the fluent stating that Ada has consented to data processing is initiated if she consents. In the Data Retention Directive 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).
```

The fluent derivation function is represented as ASP rules with `holdsat/3` atoms in the head and body. For example, in the `cfr` 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 which governs L comprises rules with occurred and holdsat atoms in the head. The occurred rules state a norm dischargement/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 dischargement/violation events occurring in L and consequently in H are about normative fluents L imposes. Likewise, 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 dischargement/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's Data Retention Regulations, `drr`, then it also holds in the EU's Data Retention Directive 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.

Finally, the semantics are 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).
```

4.3. COMPUTATIONAL FRAMEWORK

In this section we detail the computational framework in full. We provide translations from multi–level governance institution's representation and semantics in the formal framework to an ASP program. The transformation provides an ASP program which is sound and complete with respect to the formal framework, that is, the transformation produces a *corresponding* ASP program. By executing the ASP program we can determine compliance of lower–level institutions with higher–level institutions in a multi–level governance institution.

4.3.1. MULTI–LEVEL GOVERNANCE INSTITUTION ASP REPRESENTATION

We begin with a convenient translation between a state condition as it appears in an event generation or consequence function and its representation in ASP as a set of positive and negative `holdsat/3` atoms. That is, we translate a state condition *Exp* for an institution $\mathcal{I}$ accordingly:

Definition 18. ***Expression Translation*** *Let literals be denoted x_i for $i \in \mathbb{N}$. The translation of an expression Exp for an individual institution $\mathcal{I}$ with the label In is defined as:*

$$EX(Exp,\mathcal{I}) = \begin{cases} \texttt{\#true} & \textit{if } Exp = \emptyset \\ \texttt{holdsat}(p,In,\texttt{I}) & \textit{if } Exp = p \\ \texttt{not holdsat}(p,In,\texttt{I}) & \textit{if } Exp = \neg p \\ EX(x_0,\mathcal{I}),...,EX(x_n,\mathcal{I}) & \textit{if } Exp = \{x_0,...,x_n\} \end{cases}$$

Individual institutions are translated according to a set of rules for translating each individual institution in a multi–level governance institution from their formal representation to a corresponding ASP representation. Formally and subsequently described in more detail:

Definition 19. ***Multi–level Governance Institution ASP Translation*** *Let $\mathcal{ML} = \langle \mathcal{T}, R\rangle$ be a multi–level governance institution s.t. $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n\rangle$. The program Π^{insts} is the multi–level governance institutions program obtained for $\mathcal{ML}$ according to the following translation:* $\forall i \in [1,n](\mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{C}^i, \mathcal{G}^i, \mathcal{D}^i, \Delta^i\rangle)$:

$$\mathcal{I}^i \Leftrightarrow \texttt{inst}(In^i). \tag{D19.1}$$

$$e \in \mathcal{E}^i_{obs} \Leftrightarrow \texttt{evtype(e,}In^i\texttt{,ex).} \tag{D19.2}$$

$$e \in \mathcal{E}^i_{inst} \Leftrightarrow \texttt{evtype(e,}In^i\texttt{,inst).} \tag{D19.3}$$

$$f \in \mathcal{F}^i_{inert} \Leftrightarrow \texttt{ifluent(f,}In^i\texttt{).} \tag{D19.4}$$

$$f \in \mathcal{F}^i_{ninert} \Leftrightarrow \texttt{nifluent(f,}In^i\texttt{).} \tag{D19.5}$$

$$\exists X \in \mathcal{X}^i, e \in \mathcal{E}^i, p \in \mathcal{C}^{i\uparrow}(X,e) \Leftrightarrow \begin{array}{l}\texttt{initiated(}p,In^i\texttt{,I)} :- \\ \texttt{occurred(}e,In^i,I\texttt{),} \\ \texttt{not holdsat(}p,In^i\texttt{,I),} \\ EX(X,\mathcal{I}^i)\texttt{,instant(I).}\end{array} \tag{D19.6}$$

$$\exists X \in \mathcal{X}^i, e \in \mathcal{E}^i, p \in \mathcal{C}^{i\downarrow}(X,e) \Leftrightarrow \begin{array}{l}\texttt{terminated(}p,In^i\texttt{,I)} :- \\ \texttt{occurred(}e,In^i\texttt{,I),}EX(X,\mathcal{I}^i)\texttt{,} \\ \texttt{holdsat(}f,In^i\texttt{,I),instant(I).}\end{array} \tag{D19.7}$$

$$\exists X \in \mathcal{X}^i, e \in \mathcal{E}^i, e' \in \mathcal{G}^i(X,e) \Leftrightarrow \begin{array}{l}\texttt{occurred(}e',In^i\texttt{,I)} :- \\ \texttt{occurred(}e,In^i\texttt{,I),} \\ \texttt{holdsat(pow(}e'\texttt{),}In^i\texttt{,I),} \\ EX(X,\mathcal{I}^i)\texttt{,} \\ \texttt{instant(I).}\end{array} \tag{D19.8}$$

$$\exists X \in \mathcal{X}^i, f \in \mathcal{F}^i, f' \in \mathcal{D}^i(X,f) \Leftrightarrow \begin{array}{l}\texttt{holdsat(}f',In^i\texttt{,I)} :- \\ \texttt{holdsat(}f,In^i\texttt{,I),} EX(X,\mathcal{I}^i)\texttt{,} \\ \texttt{instant(I).}\end{array} \tag{D19.9}$$

$$f \in \Delta^i \Leftrightarrow \texttt{holdsat(}f,In^i\texttt{,I)} :- \texttt{start(I).} \tag{D19.10}$$

$$\exists \langle h,i\rangle \in R, n \in (\mathcal{F}^h_{anorm} \cup \mathcal{F}^h_{cnorm}) \cap \mathcal{F}^i_{ninert} \Leftrightarrow \begin{array}{l}\texttt{holdsat(}n,In^i\texttt{,I)} :- \\ \texttt{holdsat(}n,In^h\texttt{,I).}\end{array} \tag{D19.11}$$

$$\exists \langle h, i \rangle \in R, e \in \mathcal{E}^h_{norm} \cap \mathcal{E}^i_{norm} \Leftrightarrow \quad \texttt{occurred}(e,In^i,\texttt{I}) :- \texttt{occurred}(e,In^h,\texttt{I}). \tag{D19.12}$$

$$\forall f, f' \in \mathcal{F}_{cnorm} \cup \mathcal{F}_{anorm} : f \equiv f' \Leftrightarrow \quad \texttt{holdsat}(f,In^i,\texttt{I}) :- \texttt{holdsat}(f',In^i,\texttt{I}), \texttt{instant}(\texttt{I}). \tag{D19.13}$$

The translation is described as follows for an individual institution. The first translation D19.1 ensures the unique label of the institution is a fact. Translations D19.2 to D19.5 add facts corresponding to the signature of the institution for representing exogenous events, institutional events, inertial fluents and non–inertial fluents. Translations D19.6 and D19.7 add ASP rules corresponding to the consequence function, ensuring fluents are initiated and terminated if a condition on a state holds and an event occurs. Translation D19.8 adds rules for producing events according to the event generation function conditional on a state and another event occurring. Translation D19.9 adds rules for ensuring non–inertial fluents hold if they hold according to the fluent derivation function. The initial set of inertial fluents is added to the institution's initial state with transformation D19.10. If the institution governs a lower–level institution, then any normative fluents which hold in the lower–level institution are 'passed up' (if they are non–inertial fluents in the higher–level institution) D19.11. Likewise for norm dischargement and violation events D19.12. Finally, transformation D19.13 ensures any normative fluent that is equivalent to another normative fluent that holds in a state, also holds in that state. This concludes the transformations providing specific rules for a multi–level governance institution describing when events occur and fluents are initiated and terminated.

4.3.2. DEONTOLOGICAL COUNTS–AS ASP REPRESENTATION

The formal semantics for deontological counts–as, which abstracts concrete norms, is represented as a set of ASP rules for each individual institution. The formal semantics for deontological counts–as is a *general* function defined for any state describing which relatively concrete normative fluents if they hold cause another, more abstract normative fluent to hold. This is based on concepts the concrete fluents prescribe count–as more abstract concepts in a particular state. This is according to the event generation function, fluent derivation function, and counts–as between normative fluents according to deontological counts–as itself. The semantics for deontological counts–as are defined in a general way based on whether concrete concepts *could* cause a more abstract concept to occur/hold in a state. This general definition is based on inspecting the input and output of the functions representing counts–as rules. In other words, the formal semantics are defined by inspecting the left and right hand side of "A counts–as B in a context C" rules for whether the context C holds in a state and therefore deducing whether A *could* cause B to occur/hold.

However, such a general definition is not possible in ASP. This is because counts–as rules are represented as ASP rules. It is not possible to inspect the left and right hand–side of ASP rules and determine whether, for a counts–as rule “A counts–as B in a context C”, if A *could* cause B. We address the problem of having no foreseeable way to represent the formal deontological counts–as semantics as general ASP rules. That is by translating from individual institutions’ formal representation to *specific* rules that provide the same effects as deontological counts–as in a *flattened* form.

Each of these specific rules for deontological counts–as causes a specific abstract normative fluent to hold in the head. They are conditional on specific concrete normative fluents holding in the body depending on the context. Deontological counts–as is context–sensitive, an obligation to store personal data counts–as an obligation to process data non–consensually only if a user has not consented. This means it is only necessary for certain concrete normative fluents to hold in specific contexts for abstraction to take place, obliging personal data is only abstracted if a user has not consented. In different contexts the concrete normative fluents required for abstraction to take place are potentially different or no abstraction takes place. If a user has consented then an obligation to store personal data is not abstracted to non–consensual data processing. So, the body states in which contexts the concrete normative fluents necessarily hold for abstraction to take place and that at least one concrete normative fluent must be eligible for abstraction in order for abstraction to take place.

To give an example for abstracting obliging storing personal data to obliging non–consensual data processing in the Charter of Fundamental Rights interpretation of the Data Retention Directive:

```
holdsat(obl(obl(nonConsensualDataProcessing(ada),now),now),
        cfr, I) :-
   holdsat(obl(obl(storeData(CommServProv0,Agent0,personal),now),
              now), cfr, I):
   not holdsat(consentedDataProcessing(ada,isp),cfr,I);
   1{holdsat(obl(obl(storeData(isp,ada,personal),now),now),cfr,I):
     not holdsat(consentedDataProcessing(ada,isp),cfr,I)},
   holdsat(pow(cfr,nonConsensualDataProcessing(ada)),cfr,I),
   instant(I).
```

The head atom is the more abstract normative fluent obliging an obligation for data to be stored non–consensually. In the body, `holdsat(obl(obl( storeData(CommServProv0,Agent0,personal),now),now),cfr,I): not holdsat(consentedDataProcessing(ada,isp),cfr,I)`, represents that *in the context* that Ada has not consented to data processing the concrete obligation to oblige storing personal data must hold in order to be abstracted to obliging an obligation to store data non–consensually. At least one context in which abstraction can take place must hold in order for abstraction to actually take place according to `1{holdsat(obl(obl(storeData (isp,ada,personal),now),now),cfr,I): not holdsat(consentedDataProcessing(ada,isp),cfr,I)}`. In other words, if Ada has consented to data processing, then no abstraction can take place.

Such specific rules to represent deontological counts–as are formally defined as and subsequently described in detail:

Definition 20. ***Deontological Counts–as Translation*** *Let $\mathcal{ML} = \langle \mathcal{T}, \mathcal{B} \rangle$ be a multi–level governance institution s.t. $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$. The program Π^{abstr} is the deontic abstraction program obtained for $\mathcal{ML}$ according to the following translation:* $\forall i \in [1, n](\mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{C}^i, \mathcal{G}^i, \mathcal{D}^i, \Delta^i \rangle)$:

$$\begin{array}{rcl} obl(b,d) \in \mathcal{F}^i_{anorm}, b \in \mathcal{E}^i & & \texttt{holdsat(obl}(b,d),In^i,\texttt{I}):- \\ a \in \mathcal{E}^i, X \in \mathcal{X}, b \in \mathcal{G}^i(X,a) & \Leftrightarrow & CN, C, \\ CN = \{EX(obl(a,d),\ \mathcal{I}^i){:}EX(X,\ \mathcal{I}^i);\} & & \texttt{holdsat(pow}(b),In^i,\texttt{I}), \\ C = 1\{CN\} & & \texttt{instant(I)}. \end{array} \tag{D20.1}$$

$$\begin{array}{rcl} pro(b,d) \in \mathcal{F}^i_{anorm}, b \in \mathcal{E}^i & & \texttt{holdsat(pro}(b,d),In^i,\texttt{I}):- \\ CN = \{EX(pro(a,d),\mathcal{I}^i){:}EX(X,\ \mathcal{I}^i);\,| & \Leftrightarrow & CN, C, \\ a \in \mathcal{F}^i, X \in \mathcal{X}, b \in \mathcal{G}^i(X,a)\} & & \texttt{holdsat(pow}(b),In^i,\texttt{I}), \\ CN \neq \emptyset,\ C = 1\{CN\} & & \texttt{instant(I)}. \end{array} \tag{D20.2}$$

$$\begin{array}{rcl} obl(b,d) \in \mathcal{F}^i_{anorm}, b \in \mathcal{E}^i & & \texttt{holdsat(obl}(b,d),In^i,\texttt{I}):- \\ a \in \mathcal{F}^i, X \in \mathcal{X}, b \in \mathcal{D}^i(X,a) & \Leftrightarrow & CN, C, \\ CN = \{EX(obl(a,d),\ \mathcal{I}^i){:}EX(X,\ \mathcal{I}^i);\} & & C = 1\{CN\} \end{array} \tag{D20.3}$$

$$\begin{array}{rcl} pro(b,d) \in \mathcal{F}^i_{anorm}, b \in \mathcal{E}^i & & \texttt{holdsat(pro}(b,d),In^i,\texttt{I}):- \\ CN = \{EX(pro(a,d),\mathcal{I}^i){:}EX(X,\ \mathcal{I}^i);\,| & \Leftrightarrow & CN, C, \\ a \in \mathcal{F}^i, X \in \mathcal{X}, b \in \mathcal{D}^i(X,a)\}, & & \texttt{instant(I)}. \\ CN \neq \emptyset,\ C = 1\{CN\} & & \end{array} \tag{D20.4}$$

$$\begin{array}{rcl} obl(b,d), b \in \mathcal{F}^i_{anorm}, & & \\ r \in \Pi^{abstr}(H(r) = \texttt{holdsat}(b,In^i,\texttt{I})), & & \\ CN = \{EX(obl(a,d),\ \mathcal{I}^i){:}c;\,| & & \texttt{holdsat(obl}(b,d),In^i,\texttt{I}):- \\ \texttt{holdsat}(a,In^i,\texttt{I}):c; \in B(r)\} & \Leftrightarrow & CN, C, POWS, \\ POWS = EX(\{pow(e)\,| & & \texttt{instant(I)}. \\ \texttt{holdsat(pow}(e),In^i,\texttt{I}) \in B(r)\},\mathcal{I}^i), & & \\ CN \neq \emptyset,\ C = 1\{CN\} & & \end{array} \tag{D20.5}$$

$$\begin{array}{rl} pro(b,d), b \in \mathcal{F}^i_{anorm}, & \\ R = \{r \in \Pi^{abstr} : H(r) = \texttt{holdsat}(b,In^i,\texttt{I})\}, & \texttt{holdsat}(\texttt{pro}(b,d),In^i,\texttt{I}) :- \\ CN = \{EX(pro(a,d), \mathcal{I}^i):c; \mid r \in R, \Leftrightarrow & CN, C, POWS, \\ \texttt{holdsat}(a,In^i,\texttt{I}) :c; \in B(r)\} & \texttt{instant}(\texttt{I}). \\ POWS = EX(\{pow(e) \mid & \\ \texttt{holdsat}(\texttt{pow}(e),In^i,\texttt{I}) \in B(r)\},\mathcal{I}^i), & \\ CN \neq \emptyset,\ C = 1\{CN\} \in B(r) & \end{array} \tag{D20.6}$$

In more detail, the first two translations D20.1 and D20.2 produce ASP rules for abstracting first–order norms. Abstraction is based on rules of the type "A counts–as B in context C" according to the event generation $\mathcal{G}^i$ function. The second two translations D20.3 and D20.4 are, like the first, for producing first–order norm abstraction but this time based on the fluent derivation function $\mathcal{D}^i$. For all four translations D20.1 to D20.4 the resulting rules are of the type "obliging/prohibiting A(s) counts–as obliging/prohibiting B(s) in context(s) C(s)". These new abstracting counts–as rules act as base cases for producing further abstracting counts–as rules for higher–order norms by translations D20.5 and D20.6. That is, translations D20.5 and D20.6 provide rules which abstract concrete higher–order normative fluents to abstract higher–order normative fluents based on whether the prescribed lower–order normative fluents count–as abstract lower–order normative fluents according to deontological counts–as itself.

To give further examples, the rules for abstracting first–order obligations are provided by the translation according to D20.1 and D20.3. The translation is conditional on there being an event/fluent a counting–as a more abstract event/fluent b in a context X. This counts–as relation is according to the event generation and fluent dependency functions. The result is a rule that states if a concrete obligation fluent about an event/fluent a holds in a state and that state entails the context X (in which a counts–as b) then a more abstract obligation fluent prescribing b to occur/hold also holds. In the Charter of Fundamental Rights two rules cause an abstract obligation to store personal data to hold if either metadata or content data are obliged to be stored in any context (denoted with `#true`):

```
holdsat(obl(storeData(commServ,user,personal),now), cfr, I) :-
   holdsat(obl(storeData(commServ,user,content),now), cfr, I):
   #true;
   1{ holdsat(obl(storeData(commServ,user,content),now), cfr, I):
      #true; },
   holdsat(pow(storeData(commServ,user,personal)), cfr, I),
   instant(I).
holdsat(obl(storeData(commServ,user,personal),now), cfr, I) :-
   holdsat(obl(storeData(commServ,user,metadata),now), cfr, I):
   #true;
```

```
   1{ holdsat(obl(storeData(commServ,user,metadata),now), cfr, I):
      #true; },
   holdsat(pow(storeData(commServ,user,personal)), cfr, I),
   instant(I).
```

Rules for abstracting first–order prohibitions are provided by the translations according to D20.2 and D20.4. The translation is conditional on there being events/fluents a which in different contexts X count–as an abstract event/fluent b. This counts–as relation is according to the event generation and fluent dependency functions. The result is a rule that specifies for events/fluents a(s) that count–as an event/fluent b in contexts X entailed by a state, if all the events/fluents a are prohibited (before the same deadline) then a more abstract prohibition on b also holds (with the same deadline). The rule is conditional on at least one context X in which an a counts–as a b holds. The following is an example rule produced in the Charter of Fundamental Rights institution based on storing content data or storing metadata unconditionally counting–as storing personal data:

```
holdsat(pro(storeData(commServ,user,personal),never), cfr, I) :-
   holdsat(pro(storeData(commServ,user,content),never), cfr, I):
   #true;
   holdsat(pro(storeData(commServ,user,metadata),never), cfr, I):
   #true;
   1{ holdsat(pro(storeData(commServ,user,content),never), cfr, I):
   #true;
   holdsat(pro(storeData(commServ,user,metadata),never), cfr, I):
   #true; },
   holdsat(pow(storeData(commServ,user,personal)), cfr, I),
   instant(I).
```

Higher–order obligation abstraction rules are provided by the translation according to D20.5. In more detail, the translation is conditional on there being a rule that states concrete normative fluents a(s) count–as an abstract normative fluent b in contexts X. The result is a rule that is similar, albeit abstracting higher–order obligations. The new rule states that obliging a(s) before the same deadline counts–as an abstract obligation b before the same deadline in contexts X, so long as one of the contexts for an a to count–as a b holds. To give an example for the Charter of Fundamental Rights as before, albeit this time a rule that abstracts obligations on prohibitions to store content and metadata to an obligation to prohibit storing personal data:

```
holdsat(obl(pro(storeData(commServ,user,personal),never), now),
        cfr, I) :-
   holdsat(obl(pro(storeData(commServ,user,content),never), now),
           cfr, I): #true;
   holdsat(obl(pro(storeData(commServ,user,metadata),never), now),
           cfr, I): #true;
   1{holdsat(obl(pro(storeData(commServ,user,content),never), now),
              cfr, I): #true;
   holdsat(obl(pro(storeData(commServ,user,metadata),never), now),
           cfr, I): #true;},
   holdsat(pow(storeData(commServ,user,personal)), cfr, I),
```

```
    instant(I).
```

Higher–order prohibition abstraction rules are provided by the translation according to D20.6. In more detail, the translation is conditional on there being rules expressing that concrete normative fluents *a*s count–as an abstract normative fluent *b* in contexts X. The result is a rule that states prohibiting normative fluents *a*s before the same deadline in the contexts X counts–as prohibiting normative fluent *b* before the same deadline, so long as one of the contexts for an *a* to count–as a *b* holds. To give an example for the Charter of Fundamental rights, a rule causes an abstract prohibition on prohibiting storing personal data to hold if prohibiting prohibitions to store content and metadata hold (in all contexts, since content and metadata count–as personal data unconditional on a state).

```
holdsat(pro(pro(storeData(commServ,user,personal),never), never),
        cfr, I) :-
    holdsat(pro(pro(storeData(commServ,user,content),never), never),
            cfr, I): #true;
    holdsat(pro(pro(storeData(commServ,user,metadata),never),
            never), cfr, I): #true;
    1{holdsat(pro(pro(storeData(commServ,user,content),never),
    never), cfr, I): #true;
    holdsat(pro(pro(storeData(commServ,user,metadata),never),
            never), cfr, I): #true;},
        holdsat(pow(storeData(commServ,user,personal)), cfr, I),
        instant(I).
```

This concludes our presentation of the ASP representation for deontological counts–as, as a set of specific rules which *flatten* the general formal deontological counts–as semantics.

4.3.3. Multi–level Governance Semantics ASP Representation

Now we present the final component of the computational framework, the general semantics of multi–level governance represented as ASP rules. These semantics comprise two ASP programs, $\Pi^{trace(n)}$ representing a composite event trace and $\Pi^{base(n)}$ containing *general rules* to produce a multi–level governance model for a trace of composite events represented in ASP. We begin with the translation from the formal representation of a composite trace, to a corresponding ASP representation, comprising facts representing an event that is observed at each point in time:

Definition 21. ***Composite Trace Translation*** *Let $ctr = \langle e_0, ..., e_k \rangle$ be a composite trace. The program $\Pi^{trace(k)}$ is the ASP trace program for ctr comprising the following set of rules:*

$$\{\texttt{compObserved}(e_i, i). \mid i \in [n]\} \tag{D21.1}$$

We now present the ASP program $\Pi^{base(k)}$ comprising general rules for producing sequences of states transitioned between by sets of events for a composite trace, that is, a multi–level governance institution model. Formally and subsequently described:

Definition 22. ***Multi–level Governance Semantics ASP Translation*** *The program $\Pi^{base(k)}$*

is the base multi–level governance ASP program:

$$\begin{aligned}&\texttt{holdsat(P,In,J)}:-\texttt{holdsat(P,In,I)},\texttt{not terminated(P,In,I)},\texttt{next(I,J)},\\&\qquad\qquad\texttt{ifluent(P,In)}.\end{aligned} \tag{D22.1}$$

$$\texttt{holdsat(P,In,J)}:-\texttt{initiated(P,In,I)},\texttt{next(I,J)},\texttt{ifluent(P,In)}. \tag{D22.2}$$

$$\texttt{occurred(now,In,I)}:-\texttt{instant(I)},\texttt{inst(In)}. \tag{D22.3}$$

$$\texttt{occurred(E,In,I)}:-\texttt{evtype(E,In,ex)},\texttt{observed(E,In,I)}. \tag{D22.4}$$

$$\texttt{occurred(null,In,I)}:-\texttt{not evtype(E,In,ex)},\texttt{observed(E,In,I)}. \tag{D22.5}$$

$$\begin{aligned}&\{\texttt{compObserved(E,J)}\}:-\texttt{evtype(E,In,ex)},\texttt{instant(J)},\\&\qquad\qquad\texttt{not final(J)},\texttt{inst(In)}.\end{aligned} \tag{D22.6}$$

$$\begin{aligned}&:-\texttt{compObserved(E,J)},\texttt{compObserved(F,J)},\texttt{instant(J)},\texttt{evtype(E,InX,ex)},\\&\quad\texttt{evtype(F,InY,ex)},\texttt{E!=F},\texttt{inst(InX;InY)}.\end{aligned} \tag{D22.7}$$

$$\texttt{obs(I)}:-\texttt{compObserved(E,I)},\texttt{evtype(E,In,ex)}. \tag{D22.8}$$

$$:-\texttt{not obs(I)},\texttt{not final(I)},\texttt{instant(I)}. \tag{D22.9}$$

$$\texttt{observed(E,In,I)}:-\texttt{compObserved(E,I)},\texttt{inst(In)}. \tag{D22.10}$$

$$\begin{aligned}&\texttt{occurred(disch(obl(A,D)),In,I)}:-\texttt{holdsat(obl(A,D),In,I)},\\&\qquad\qquad 1\{\texttt{occurred(A,In,I)},\\&\qquad\qquad \texttt{holdsat(A,In,I)}\}.\end{aligned} \tag{D22.11}$$

$$\begin{aligned}&\texttt{occurred(viol(obl(A,D)),In,I)}:-\texttt{holdsat(obl(A,D),In,I)},\\&\qquad\qquad \texttt{not occurred(disch(obl(A,D)),In,I)},\\&\qquad\qquad 1\{\texttt{occurred(D,In,I)},\texttt{holdsat(D,In,I)}\}.\end{aligned} \tag{D22.12}$$

$$\begin{aligned}&\texttt{occurred(disch(pro(A,D)),In,I)}:-\texttt{holdsat(pro(A,D),In,I)},\\&\qquad\qquad 1\{\texttt{occurred(D,In,I)},\\&\qquad\qquad \texttt{holdsat(D,In,I)}\}.\end{aligned} \tag{D22.13}$$

$$\begin{aligned}&\texttt{occurred(viol(pro(A,D)),In,I)}:-\texttt{holdsat(pro(A,D)),In,I)},\\&\qquad\qquad 1\{\texttt{occurred(A,In,I)},\texttt{holdsat(A,In,I)}\},\\&\qquad\qquad \texttt{not occurred(disch(pro(A,D)),In,I)}.\end{aligned} \tag{D22.14}$$

$$\begin{aligned}&\texttt{terminated(N,In,I)}:-1\{\texttt{occurred(viol(N),In,I)},\\&\qquad\qquad \texttt{occurred(disch(N),In,I)}\},\\&\qquad\qquad \texttt{ifluent(N,In)},\texttt{holdsat(N,In,I)}.\end{aligned} \tag{D22.15}$$

$$\begin{aligned}&\texttt{final}(k).\ \texttt{start(0)}.\ \texttt{instant(0..T)}:-\texttt{final(T)}.\\&\texttt{next(T,T+1)}:-\texttt{instant(T)}.\end{aligned} \tag{D22.16}$$

The common–sense law of inertia is captured with the first rule, D22.1. This states an inertial fluent held in the previous state and it has not yet been terminated, then it continues to hold. Initiated fluents hold in the state after they are initiated due to rule D22.2. The event of '*now*' occurs in every time instant according to D22.3. Collectively, D22.4 and D22.5 ensure there is a *synchronised* trace for each institution for a given composite trace. If an event is observed and an institution recognises it as an exogenous event, then that event occurs in the institution according to D22.4. If there is no event recognised by an institution that is observed, then the null event occurs in that institution D22.5. Rule D22.6 gives an arbitrary choice to include an observed event in the composite trace, which, together with rules D22.7 to D22.10 ensures there is a complete composite trace of length *k* and that no two observable events occur at the same time. Rules D22.11 to D22.14 ensure relevant compliance events occur when normative fluents that hold are discharged/violated. Rule D22.15 ensures discharged and violated normative fluents are terminated. Finally, facts and rules in D22.16 ensure the model and composite trace is of length *k*, that there are facts representing time instants between zero and *k*, and that the time instants are strictly ordered according to `next/2`.

This concludes our presentation of the ASP programs which comprise the computational framework, thus, allowing multi–level governance institutions and their semantics to be executed and consequently compliance checks performed.

4.4. EXECUTED 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 which 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:

```
observed(exUseElectronicCommunication(ada, isp), 0).
observed(exStoreData(isp, ada, metadata), 1).
observed(exRequestData(charles, isp, ada), 2).
observed(exConsent(ada, isp), 3).
observed(exUseElectronicCommunication(ada, isp), 4).
observed(exStoreData(isp, ada, metadata), 5).
```

First the agent Ada uses electronic communications provided by the service provider ISP. Then, the service provider, ISP, stores Ada's communications metadata. An agent, Charles, requests data from ISP concerning Ada. Ada consents to her data being stored (after it is stored). Ada uses ISP's electronic communications again. Finally, ISP stores Ada's metadata again. In the first half of this observable event trace Ada's data is stored without her consent, in the latter half Ada's data is stored after she has given consent.

The resulting multi–level governance institution model is depicted in Figure 4.1, for brevity edited to just containing those fluents that are relevant to the discharge and violation

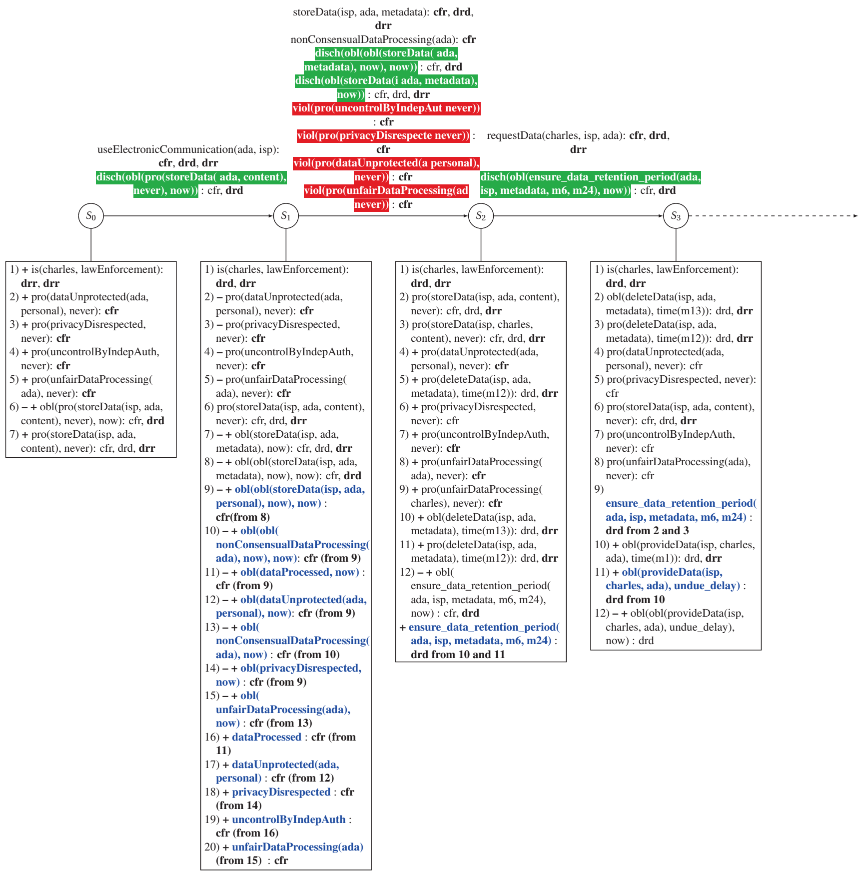

Figure 4.1: Case study execution. The originating institutions for a fluent are in **bold**, '+' indicates an initiated fluent, '–' indicates a terminated fluent. Non–inertial fluents are in **bold** denoting they are derived from other fluents according to the fluent derivation and deontological counts–as (norm abstraction) operations. Norm **discharge** and **violation** events are highlighted.

of norms. 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` in the UK–DRR and EU–DRD. 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

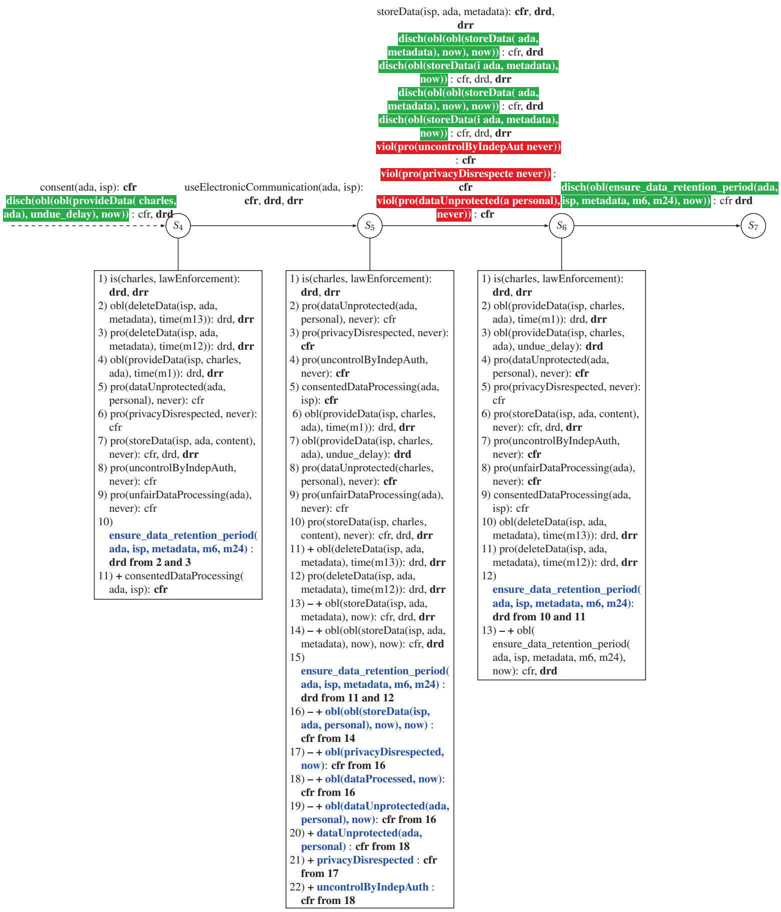

Figure 4.1 (cont.)

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. The transition to state S_2 includes the event of `ada`'s communications' metadata being stored by `isp`, driven by the corresponding exogenous event, and consequently the discharge of the obligation from the UK–DRR for `isp` to store Ada's communications metadata.

- State S_2 – The EU–DRD, which governs the UK–DRR, obliges data retention to be ensured for between 6 and 24 months. In comparison, the UK–DRR, both obliges that Ada's communications metadata is deleted before 13 months and prohibits deletion before 12 months. Hence, the UK–DRR requires that metadata is stored for between 12 and 13 months, which is abstractly interpreted in the EU–DRD as ensuring data is retained between 6 and 24 months (`ensure_data_retention_period(ada, isp, metadata, m6, m24)`) discharging the EU–DRD's obligation for data to be stored within this timeframe. In the transition to state S_3 `charles` requests Ada's data from `isp`.

- State S_3 – Since `charles` is playing the role of `lawEnforcement`, this causes the EU–DRD to oblige that `isp` is obliged to provide the data before any `undue_delay`. Meanwhile, the UK–DRR obliges `isp` to provide `charles` with the data within one month (`m1`). According to the EU–DRD anything occurring before one month *counts–as* the event of `undue_delay`. Thus the EU–DRD interprets the obligation to provide data within one month as the abstract obligation to provide data before any undue delay. This causes discharge of the obligation to oblige data is provided before any undue delay.

- States S_4 to S_6 follow largely the same pattern. In the transition to the next state S_4 `ada` consents to her data being stored, which has no affect on the UK–DRR or the EU–DRD. Transitioning to state S_5 `ada` uses electronic communications, then `ada`'s data is stored, causing the same obligations and prohibitions to be imposed by the UK–DRR and EU–DRD as when these events occurred previously.

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 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 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).

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 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. A full compliance check examines all possible social contexts by checking all possible traces of events that result in unique answer–sets (or in our formal framework, models).

4.5. 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 the appendices). We begin by packaging, for convenience, the aforementioned ASP programs into a single ASP program $\Pi^{\mathcal{ML}(k)}$.

Definition 23. ***Multi–level Governance ASP 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 institutions 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 ASP program for $\mathcal{ML}$ and a composite trace ctr is:*

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

The first property we present gives the set of of events that are *maximally* in the set of events according to the event generation operation.

Lemma 1. *If GR^i is the event generation operation for an institution $\mathcal{I}^i$ with respect to a* *tuple of events E and a tuple of events S, and* $E' = GR^i(S^i, E^i)$ *w.r.t. S and E then* $\forall e' \in E'$:

$$e' = now \qquad \textbf{\textit{or}} \qquad (T1.1)$$

$$e' \in E^i \qquad \textbf{\textit{or}} \qquad (T1.2)$$

$$\exists X, e : X \in \mathcal{X} \land e \in E \land S^i \models X \land e' \in \mathcal{G}(X, e) \land S^i \models pow(e') \qquad \textbf{\textit{or}} \qquad (T1.3)$$

$$e' = disch(obl(a,d)) s.t. \exists obl(a,d) : S^i \models obl(a,d) \land (a \in E' \lor S^i \models a) \qquad \textbf{\textit{or}} \qquad (T1.4)$$

$$e' = viol(obl(a,\ d)) s.t. \exists obl(a,d) : S^i \models obl(a,d) \land (d \in E' \lor S^i \models d) \land$$
$$disch(obl(a,d)) \notin E' \qquad \textbf{\textit{or}} \qquad (T1.5)$$

$$e' = disch(pro(a,d)) s.t. \exists pro(a,d) : S^i \models pro(a,d) \land (d \in E' \lor S^i \models d) \qquad \textbf{\textit{or}} \qquad (T1.6)$$

$$e' = viol(pro(a,d)) s.t. \exists pro(a,d) : S^i \models pro(a,d) \land (a \in E' \lor S^i \models a) \land$$
$$disch(pro(a,d)) \notin E' \qquad \textbf{\textit{or}} \qquad (T1.7)$$

$$\exists \langle h, i \rangle : \langle h, i \rangle \in R, e \in \mathcal{E}^h_{norm} \cap \mathcal{E}^i_{norm} \qquad (T1.8)$$

Proof. Immediately follows from Definition 37. □

The second property we present gives the set of fluents that are maximally in the closure of deontological counts–as.

Lemma 2. *If* DC^ω *is the closure of the deontological counts–as function for an institution* $\mathcal{I} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \mathcal{D}, \Delta \rangle$, *a state* $S \in \Sigma$ *for* $\mathcal{I}$ *and* $S' = DC^\omega(S)$ *then* $\forall f' \in S'$:

$$f' \in S \quad \textbf{\textit{or}} \qquad (T2.1)$$

$$\exists \langle N', f' \rangle \in DC(S') : N' \subseteq S' \qquad (T2.2)$$

Proof. Immediately follows from Definition 7. □

We now give the soundness property for the deontic abstraction representation in ASP 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 ASP 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 deontic counts–as function $DC^{i\omega}$ applied.

Lemma 3. *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 ASP 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 $S^i_j = DC^{i\omega}(S^i_j)$.

Proof. See the appendices. □

The next property we are interested in is soundness for the translation to an ASP program as a whole. Specifically, the property states any answer–set for a multi–level governance ASP 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 ASP 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:*

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

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

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

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

Proof. See the appendices. □

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 ASP 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 ASP program for*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Proof. See the appendices. □

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

4.6. RELATED WORK

In this chapter we presented a computational framework for reasoning about multi–level governance. That is, we focussed on the practical side. Whilst there are many practical institutional reasoning frameworks, we find none which contribute automated reasoning for determining compliance in multi–level governance. We discuss the related work in more detail as follows.

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. [63, 159, 160]) 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. [26, 58, 110, 111, 165, 229]). 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. [42, 43, 98, 103, 130, 154]). 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. [42, 43]. Li et al. have made developments on InstAL for detecting conflicts between norms [156], in particular in interacting institutions [155] and cooperating institutions [154]. In the work of Li et al. institutions are linked with special *bridge* institutions such that events occurring in one can cause events to occur in another and likewise for fluents being initiated or terminated. Such bridge institutions have a similar role to our links between different levelled institutions. The difference is that bridge also operate as a king of institution, evolving from one state to the next such that the bridges (links) between institutions can change over time. In contrast, we use institutional links which remain static over time. Further developments on InstAL were realised by Pieters et al. [194, 195] for reasoning about institutions as a means to police and enforce security policies. In their work, Pieters et al. [194, 195] extend InstAL with rules for non–inertial fluents which (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 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. Our work is also related to our previous work which was loosely based on InstAL [144, 145] for reasoning about multi–tier institutions and higher–order norms.

The main differences between these developments and this chapter is we have extended InstAL for representation and reasoning about multi–level governance. In more detail, there are differences in reasoning about permissive societies (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 chapter: combining higher–order normative reasoning and norm abstraction. We summarise all of these differences in Table 4.7.

Also relevant are practical approaches for reasoning about normative social commitments (e.g. contracts, promises) [113, 243], using the Event Calculus [150]. These bear similarity to our own approach since we use EC–like constructs (initiation and termination of fluents driven by events) and reasoning. Commitments have been formalised with 'life-cycle' 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 these approaches but do not coincide with our notion of higher–order norms. In our case, a higher–order norm represents a statement such as 'the *outcome* of your rules should not be an obligation to do X in context C', where X can be a concrete or abstract notion. On the other hand, nested commitments represent 'you have promised to me that you will not make a *commitment* towards person X to do Y in context C', here the nesting simply represents that a rule of a specific form should not be created, regardless of the rule's effects and the abstract meaning

	InstAL [42, 43]	Li et al. [154–156]	Pieters et al. [194, 195]	King et al. [144, 145]	This chapter
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 4.7: Comparison between closely related developments on InstAL.

of those effects. Consequently, nested commitments, come from a fundamentally different perspective and are not aligned with our formalisation of regulations which govern other regulations nor do they capture abstraction.

Another practical institutional reasoning approach is temporal defeasible deontic logic. Defeasible logic is a non–monotonic logic designed to be implemented in Prolog [13, 183]. There are three rules types in defeasible logic, *strict* rules ($\rightarrow$) whose conclusion is true so long as the antecedent is true, *defeasible* rules ($\Rightarrow$) whose conclusion is true unless the rule is rebutted by another rule, and *defeating* rules ($\leadsto$) whose conclusion is never true but if the antecedent is true rebuts other rules with a contradictory conclusion. Defeasible logic 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 which rebut them by applying defeating rules, and then recursively counter–attacking all rebuttals with further arguments, terminating by constraints on non–repeatability of arguments (e.g. [200]). Defeasible *temporal deontic logics* proposed by Governatori et al. [98, 103] 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.

4.7. DISCUSSION

In this chapter, we answered the question "*how can institutional design compliance in multi–level governance be computationally verified?*" with a computational framework. In doing so, we provided a practical way to reason with the semantics proposed in the previous chapter. We assessed our proposal along two 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 showing the two correspond. 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 in order to specify multi–level governance and the means to automatically detect (non–)compliance. To summarise, this chapter provides both a practical way to determine compliance in multi–level governance and also a way to assess the previous chapter's proposed semantics with a real–world case study and its execution.

The main weakness of this chapter's contributions is that it is heavily dependent on Answer–Set Programming. In one sense this is not a problem, since the previous chapter proposed a semantics which 'stand on their own'. In another sense, this chapter aims to provide practical reasoning, whilst certain limitations of answer–set programming have practical implications. Specifically, due to answer–set programming's limitations, this chapter was unable to provide general rules for all of the semantics proposed in the previous section. In particular, the deontological abstraction semantics, characterised as a function in the previous section, are represented as a flattened function in the form of ASP rules in this section. What this means is that any time an institution design is changed and we wish to check its compliance, then the high–level description of an institution needs to be newly compiled in its entirety to an Answer–Set Program. If deontological abstraction's semantics had a general set of ASP rules, then only the changed rules in the changed institution would need a new corresponding ASP representation. To address this weakness, future work should investigate different representation results. Possible ways to achieve this include a corresponding representation in Prolog (e.g. as for defeasible logic [13]) or an embedding in a higher–order logic theorem prover where the higher–order logic could act as a meta–language in which to represent the semantics from the previous section (e.g. as in [18]).

5

Explanatory Rectifications for Non–compliant Institutions

Plurality is never to be posited without necessity.

William of Ockham

Whenever possible, substitute constructions out of known entities for inferences to unknown entities.

Bertrand Russell

This chapter is based on the following paper:
King, T. C., Li, T., Vos, M. De, Jonker, C. M., Padget, J., & Riemsdijk, M. B. Van. (2016). Revising Institutions Governed by Institutions for Compliant Regulations. Coordination, Organizations, Institutions, and Normes in Agent Systems XI: COIN 2015 International Workshops, COIN@ AAMAS, Istanbul, Turkey, May 4, 2015, COIN@ IJCAI, Buenos Aires, Argentina, July 26, 2015, Revised Selected Papers., 9628, 191 – 208. [145]

This chapter makes the following contributions:

- An application of Inductive Logic Programming (ILP) in ASP, demonstrated with an implementation, to provide a mechanism that *successfully* revises rules of a non–compliant institution to ensure it is compliant.
- A revision mechanism that ensures institution revisions are *minimal* in their regulatory effects through adding, deleting and modifying rules, thereby keeping as closely as possible to the designer's original intentions.

In chapters 3 and 4 we presented a formal and computational framework for determining compliance in multi–level governance. The contributions can be used by a judiciary to determine punishments for non–compliance or by a legislature to avoid enacting non–compliant legislation. However, at this point in the dissertation it is unclear how a legislature should address non–compliance by changing an institutional design. In this chapter we return to the issue of rule change, but this time asking the question, how can we mechanically find ways in which legislation can be modified to ensure it is compliant?

On the one hand, an institution designer wishes to avoid punishment for non–compliant institution designs. On the other hand, an institution designer designed the non–compliant institution with the aim of meeting particular societal outcomes. At the same time, it is important not to just rectify non–compliance but also find the simplest explanation for non–compliant institution designs in order to support a legislator in avoiding making the same mistakes.

It is an arduous task to determine non–compliance, but even more arduous to rectify it in a way that keeps as closely as possible to the legislature's original intentions. Firstly because different rule modifications, namely addition, deletion and changes to the antecedent, must be tried to determine if they result in compliance in various contexts and therefore find a possible explanation for non–compliance. Secondly, they must be compared to alternative revisions that result in compliance to see if one makes fewer changes to the rules than another, in order to find the *simplest* and most general explanation for non–compliance. Thirdly, because different revisions must also be optimised for keeping as closely as possible to the designer's original intentions by still producing the same legal effects (e.g. obligations and prohibitions) that do not result in the institution design being non–compliant. Clearly, this is a mechanical task ripe for computation.

The idea is to test different changes to an institution design in order to rectify non–compliance and therefore find possible explanations for non–compliance. Then, to take the *simplest* explanation, which minimally changes an institution's outcomes and rules, and apply it as a remedy. In this chapter we address the burden on the institution designer in revising to be compliant with a computational mechanism which also minimise the changes to the outcomes of the institution design.

This chapter builds on the previous chapters' formal and computational frameworks. However, this chapter deals with a simplified version of the formal and computational framework. We discuss this simpler framework, its application to a new case study and the approach we take in 5.1. Then, we present the computational mechanism for successfully and minimally revising an institution for compliance in section 5.2. We finish this chapter by comparing to other mechanisms for revising institutions and sets of rules in 5.3 and conclude with discussion on limitations and wider implications in section 5.4.

5.1. APPROACH

In this chapter, we look at revising rules for compliance. As we saw in the previous chapter, individual institutions which are governed by other institutions (in that case, within multi–level governance), can be represented as ASP rules. Building on that idea, the approach we take to revising an institution to be compliant is to revise its representation as ASP rules, using ASP itself to try various rule modifications.

However, as we discussed at the end of the preceding chapter, there is a limitation of the approach taken due to two contributing factors. Firstly, because whenever a constitutive rule is changed in an institution, then the abstract meaning of concrete normative fluents, which is derived from the constitutive rules, also changes. Secondly, because normative fluent abstraction is represented in ASP as specific ASP rules (rather than a general semantics as in our formal framework). Taking these two factors together, this means whenever a constitutive rule is changed in ASP then potentially other ASP rules must also be changed representing the abstraction of a normative fluent. Consequently, if we use the framework presented in the preceding section which includes ASP rules for abstracting normative fluents, then the process of changing rules involves firstly, changing the rules, secondly determining all other normative fluent abstraction rules which need to be changed or added. This is a complex process and consequently we look at a simpler problem where we change rules in institutions governed by institutions, where each governance level *operates at the same level of abstraction*.

To address this simpler problem, we first re–introduce a simpler version of our framework. In this version of the framework, we define *multi–tier institutions*, first presented in [144], which defines a tiering of institutions, each institutional tier governing the tier below, where norms are not abstracted. We overview the simpler framework in subsection 5.1.1, then introduce the formal representation in subsection 5.1.2, then we provide the operational semantics in ASP in subsection 5.1.4. Finally, we describe the main technique we use, Inductive Logic Programming (ILP) theory revision, in subsection 5.1.6.

5.1.1. MULTI–TIER INSTITUTIONS

We begin by providing an overview of individual and multi–tier institutions, schematically depicted in figure 5.1. To recap the main concepts, an individual legal institution acts as a mechanism to guide the behaviour of the system it governs. Institutions define a set of constitutive and regulative rules which respectively establish an institutional description and prescription of reality (see Searle's counts–as relation [215]). Constitutive rules *describe* the system governed through creating institutional facts that can represent events caused by other events (e.g. entering a location which is private counts–as entering a private location), or they can represent changes to the institutional state (e.g. entering a private location causes an agent to be at a private location). Regulative rules *prescribe* what properties should hold/events should occur in a system by creating obligations and prohibitions in states (e.g. when requested an agent is obliged to share their location). An institution's regulative rules regulate over a social interpretation of reality constructed from brute facts by constitutive rules.

Conceptually, a multi–tier institution extends the notion of an individual institution governing an MAS to institutions governing institutions in a tiered structure. Each institutional tier governs the tier below. The first–tier imposes norms on what occurs and holds in an

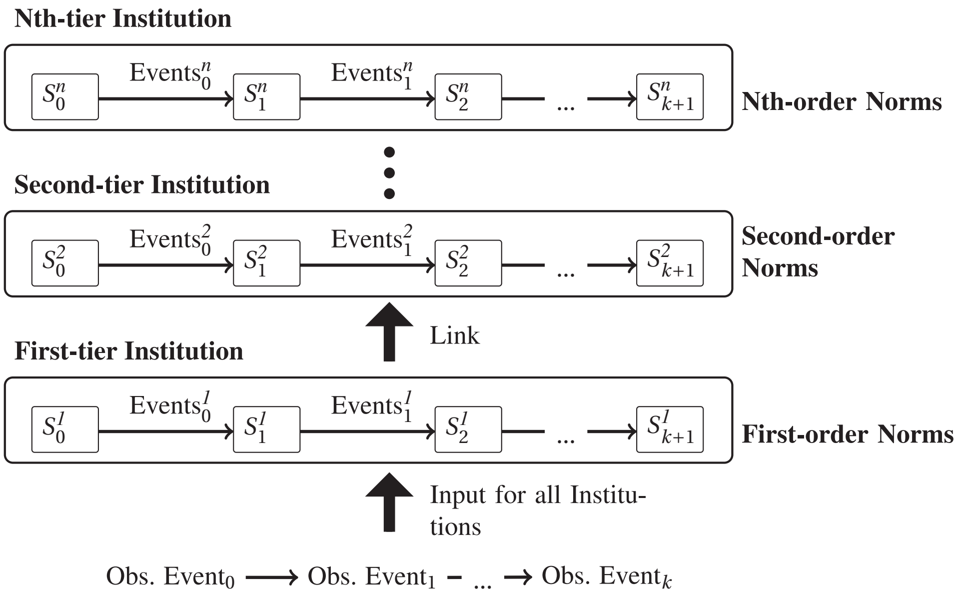

Figure 5.1: Schematic view of a multi–tier institution model

MAS (first–order norms), the second–tier norms on the norms imposed by the first (norms about first–order norms, i.e. second–order norms), and so on. When a lower–tier imposes a normative fluent which is non–compliant with a higher–tier, events denoting a higher–order norm violation occur.

The operational semantics of a multi–tier institution, first presented in [144], allow such non–compliance to be determined by checking a multi–tier institution model for norm violations. Depicted in Figure 5.1, the model describes how each ith–tier institution evolves over time, as an event–state sequence, in response to the evolution of the tier below. The first–tier evolves in response to a trace of observable events that *could* occur in an MAS (i.e. produced for a pre–runtime check). Each tier above the first evolves in response to the event–state sequence of the institution they govern (i.e. the tier below). States contain domain fluents describing the MAS and normative fluents prescribing the events that should occur and fluents that should hold in the tier below (including other normative fluents). Each state transition is caused by events occurring in the institution from the previous state, which are in turn driven by the events and states from the tier below. If a normative fluent in a state is violated by an event or fluent in the tier below (including another normative fluent) a norm violation event occurs in the transition to the next state. Thus, model–checking can be used to compliance–monitor one institution with another by checking for higher–order norm violation events.

Differing from previous chapters 3 and 4 on multi–level governance institutions, in this chapter *multi–tier* institutions restrict each ith tier in regulating the tier below ith–order norms. Moreover, each institutional state does not include the abstraction of any concrete normative fluents.

5.1.2. FORMAL REPRESENTATION

In this section we provide the formal representation of institutions and multi–tier institutions from [144] as follows. We start with the representation for 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 24. ***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:*

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

Institutions in this chapter share many common characteristics of institutions presented earlier. Formally, an institution is defined as and described subsequently:

Definition 25. ***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.*

In more detail, 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 ($\mathcal{E}_{inst}$). The set of institutional events ($\mathcal{E}_{inst}$) 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. Each institution starts at an initial state (Δ). An institution evolves from one state to the next in response to observable 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. As previously, 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.

The approach we take to representing a multi–tier institution is to restrict each ith–tier institution 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 $i$thorder 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. $i$thorder 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. For monitoring, we will later define using ASP rules how each tier is linked, such that each ith–tier contains the normative events and normative fluents produced/imposed by the tier below. Formally, a multi–tier institution is defined as:

Definition 26. ***Multi–tier Institution*** *Let P be a set of propositions denoting the domain, a multi–tier institution is a tuple $\mathcal{T} = \langle \mathcal{I}^1|_{P^1}, ..., \mathcal{I}^n|_{P^n}\rangle$ where:*

- *Each element of $\mathcal{T}$ is an individual institution 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 $i$thorder norms.*

This concludes the representation for individual and multi–tier institutions.

5.1.3. CASE STUDY

Now, we introduce our case study and demonstrate the multi–tier institution representation by formalising it. Our case study is in the context of a system for crowdsourcing audio data from users using specialised cellphone apps, called a soundsensing system [162]. Our case study is for a tier–1 soundsensing institution (SS) which governs users contributing audio data using cellphone apps, and for the tier–2 governmental institution (GI) which regulates the regulations of the soundsensing system away from placing unacceptable limits on user's rights. The case study is formalised in table 5.1 and table 5.3. We follow the convention that upper–case symbols stand for variables. For brevity we leave out the set of events and fluents for each institution. Both institutions consist of rules describing the domain (e.g. an agent entering a new location causes the agent to be at that location) and consider the location 'street b' to be private and the agent 'Bertrand' to be a child (see the initial states).

The soundsensing institution is described as follows and formalised in Table 5.1:

Soundsensing Tier–1 Institution

- Entering a new location causes an agent to leave the previous location (SSIG.1).
- In general when a norm is violated a generic norm violation event occurs (SSIG.2, SSIG.3 and SSIG.4).
- Due to a designer error, leaving a location also causes a generic norm violation event (SSIG.5).
- A user entering a location which is private counts–as entering a private location (SSIG.6).
- Likewise, leaving a location which is private counts–as leaving a private location (SSIG.7).
- Entering a new location causes a user to be at that location (SSICI.1).
- Likewise, leaving a location means the user is no longer at their previous location (SSICT.1).
- When a user turns 14 years of age, they are no longer considered a child (SSICT.2) as defined in real–world regulations pertaining to the retention of children's data[227].
- Users are obliged to provide their location on request to give the collected data location context (SSICI.2).
- If a user violates a norm they are obliged to pay a fine (SSICI.3).
- Users are forbidden from turning their microphone off to ensure data is collected continuously (SSIIS).

In turn, the soundsensing institution is governed by a tier–2 governmental institution designed to meet different aims (e.g. maintaining agents' rights). It is formalised in table 5.3 and described as follows:

Governmental Tier–2 Institution

- Entering a new location causes an agent to leave the previous location (GGI.1).
- When a first–order norm is violated a generic norm violation event occurs (GGI.2 and GGI.3).
- Entering a location causes an agent to be at that location (CIGI.1).
- Likewise, leaving a location causes an agent to no longer be at that location (CTGI.1).
- If someone turns 14 years of age then they are no longer a child (CTGI.2), again, based on real–world regulations pertaining to the retention of children's data[227].
- It is obliged that fines are only imposed on users after they violate a norm. In the initial state (GIIS) there is an obligation for a norm violation to occur before an obligation to pay a fine is imposed. Naturally, the obligation for a norm violation to occur first does not hold once a fine is imposed. If there is an obligation to pay a fine and that obligation is discharged (i.e. the fine is paid), then the obligation for a norm to be violated before a fine is imposed is reinstated (CIGI.2).
- When a user is in an area that forbids audio recording (i.e. a private area), it is forbidden to forbid them from turning their microphone off (CIGI.3).
- It is forbidden to oblige children (users under the age of 14) to share their location (GIIS). Similar regulations can be found in the United States Government's Child Privacy and Protection Act [227].

This concludes the formalisation of our case study. In the next section we will introduce the computational framework we use in this chapter and present the results of using the computational framework to check the compliance of the soundsensing institution with the governmental institution.

Table 5.1: **Soundsensing Institution Formalisation**

$$\mathcal{G}^{SS}(\{at(Loc0,Ag0)\},enter(Loc1,Ag0)) \ni leave(Loc0,Ag0) \tag{SSIG.1}$$

$$\mathcal{G}^{SS}(\emptyset, viol(obl(share\_location(Ag0),leave(Ag0, Loc0)))) \ni norm\_violation(Ag0) \tag{SSIG.2}$$

$$\mathcal{G}^{SS}(\emptyset, viol(obl(share\_location(Ag0),leave(Ag0, Loc0)))) \ni norm\_violation(Ag0) \tag{SSIG.3}$$

$$\mathcal{G}^{SS}(\emptyset, viol(pro(microphone\_off(Ag0),leave\_soundsensing(Ag0)))) \ni norm\_violation(Ag0) \tag{SSIG.4}$$

$$\mathcal{G}^{SS}(\emptyset, \textit{enter(Ag0, Loc0)}) \ni \textit{norm\_violation(Ag0)} \tag{SSIG.5}$$

$$\mathcal{G}^{SS}(\{\textit{private(Loc0)}\}, \textit{enter(Ag0, Loc0)}) \ni \textit{enter\_private(Ag0)} \tag{SSIG.6}$$

$$\mathcal{G}^{SS}(\{\textit{private(Loc0)}\}, \textit{leave(Ag0, Loc0)}) \ni \textit{leave\_private(Ag0)} \tag{SSIG.7}$$

$$\mathcal{C}^{SS\uparrow}(\emptyset, \textit{enter(Loc0, Ag0)}) \ni \textit{at(Ag0, Loc0)} \tag{SSICI.1}$$

$$\begin{aligned}&\mathcal{C}^{SS\uparrow}(\emptyset, \textit{request\_location(Ag0)}) \ni \\ &\qquad \textit{obl(share\_location(Ag0), leave(Ag0, Loc0))}\end{aligned} \tag{SSICI.2}$$

$$\begin{aligned}&\mathcal{C}^{SS\uparrow}(\emptyset, \textit{norm\_violation(Ag0)}) \ni \\ &\qquad \textit{obl(pay\_fine(Ag0), leave\_soundsensing(Ag0))}\end{aligned} \tag{SSICI.3}$$

$$\mathcal{C}^{SS\downarrow}(\emptyset, \textit{leave(Loc0, Ag0)}) \ni at(Loc0, Ag0) \tag{SSICT.1}$$

$$\mathcal{C}^{SS\downarrow}(\{\textit{child(Ag0)}\}, \textit{birthday(Ag0, 14)}) \ni \textit{child(Ag0)} \tag{SSICT.2}$$

$$\begin{aligned}\Delta^{SS} = \{&\textit{private(street\_b)}, \textit{at(ada, street\_b)}, \\ &\textit{at(bertrand, street\_c)}, \textit{child(bertrand)}, \\ &\textit{pro(microphone\_off(Ag0),leave\_soundsensing(Ag0))}\}\end{aligned} \tag{SSIIS}$$

Table 5.3: **Governmental Institution Formalisation**

$$\mathcal{G}^{GI}(\{at(\textit{Loc0,Ag0})\}, enter(\textit{Loc1,Ag0})) \ni leave(\textit{Loc0,Ag0}) \tag{GGI.1}$$

$$\begin{aligned}&\mathcal{G}^{GI}(\emptyset, \textit{viol(obl(share\_location(Ag0),leave(Ag0, Loc0))))} \ni \\ &\qquad \textit{norm\_violation(Ag0)}\end{aligned} \tag{GGI.2}$$

$$\begin{aligned}&\mathcal{G}^{GI}(\emptyset, \textit{viol(pro(microphone\_off(Ag0),leave\_soundsensing(Ag0))))} \ni \\ &\qquad \textit{norm\_violation(Ag0)}\end{aligned} \tag{GGI.3}$$

$$\mathcal{C}^{GI\uparrow}(\emptyset, \textit{enter(Loc0, Ag0)}) \ni \textit{at(Ag0, Loc0)} \tag{CIGI.1}$$

$$\begin{aligned}&\mathcal{C}^{GI\uparrow}(\emptyset, \textit{disch(obl(pay\_fine(Ag0),leave\_soundsensing(Ag0))))} \ni \\ &\qquad \textit{obl(norm\_violation(Ag0),} \\ &\qquad\qquad \textit{obl(pay\_fine(Ag0),leave\_soundsensing(Ag0)))}\end{aligned} \tag{CIGI.2}$$

$$\begin{aligned}&\mathcal{C}^{GI\uparrow}(\{\textit{private(Loc0)}\}, \textit{enter(Loc0)}) \ni \\ &\qquad \textit{pro(pro(microphone\_off(Ag0),leave\_soundsensing(Ag0)),} \\ &\qquad\qquad \textit{leave(Loc0))}\end{aligned} \tag{CIGI.3}$$

$$\mathcal{C}^{GI\downarrow}(\emptyset, \textit{leave(Loc0, Ag0)}) \ni at(Loc0, Ag0) \tag{CTGI.1}$$

$$\mathcal{C}^{GI\downarrow}(\{\textit{child(Ag0)}\}, \textit{birthday(Ag0, 14)}) \ni \textit{child(Ag0)} \tag{CTGI.2}$$

$$\begin{aligned}\Delta^{GI} = \{&\textit{obl(norm\_violation(Ag0),} \\ &\qquad \textit{obl(pay\_fine(Ag0),leave\_soundsensing(Ag0))),} \\ &\textit{pro(obl(share\_location(bertrand),leave(Ag0, Loc0)),} \\ &\qquad \textit{birthday(bertrand, 14))}\} \cup \Delta^{SS}\end{aligned} \tag{GIIS}$$

5.1.4. MULTI–TIER INSTITUTION OPERATIONALISATION IN ASP

The formal representation for multi–tier institutions is complemented by a corresponding ASP operationalisation, much like in chapter 4, for automatic compliance checking of lower–tier institutions with higher–tier institutions. The ASP program, comprises rules using the familiar `initiated`(*p, In,* `I`), `terminated`(*p, In,* `I`), `occurred`(*e, In,* `I`), `holdsat`(*f, In,* `I`), *start*(*I*) and *instant*(*I*) literals outlined previously, as well as the set of state formula literals *EX(X, In, I)*.

The ASP program comprises several components which we refer to later in this chapter. Firstly, programs presented in the preceding chapter:

- A program representing a trace of observable events used as input for producing multi–tier models, the timeline program $\Pi^{trace(k)}$ given previously in definition 21.
- an implementation of the operational semantics, the reasoning program $\Pi^{base(k)}$ given previously in definition 22.

Secondly, two new types of ASP program:

- Programs $\Pi^{\mathcal{I}^i}$ for each individual institution $\mathcal{I}^i$ in the multi–tier institution $\mathcal{M}$ according to the translation given in the subsequent definition 27.
- A program Π^{mtreas}, given in subsequent definition 28. It represents the filtering functions linking the individual institutions in a multi–tier institution, thereby ensuring normative fluents and norm compliance events are passed from lower–tiers to higher–tiers.

The new types of ASP program are defined as follows. First, the programs $\Pi^{\mathcal{I}^i}$ for representing each individual institution in a multi–tier institution in ASP. Defined and described subsequently as:

Definition 27. ***Multi–tier Institution ASP Translation*** *Let* $\mathcal{T} = \langle \mathcal{I}^1|_{P^1}, ..., \mathcal{I}^n|_{P^n} \rangle$ *be a multi–tier institution. The programs* $\Pi^{\mathcal{I}^i}$ *are defined for each individual institution* $\mathcal{I}^i$ *according to the following translation:* $\forall i \in [1, n]$, $(\mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{C}^i, \mathcal{G}^i, \Delta^i \rangle)$*:*

$$\mathcal{I}^i \Leftrightarrow \quad \texttt{tier}(In,i).\texttt{inst}(In). \in \Pi^{\mathcal{I}^i} \tag{IT.1}$$

$$e \in \mathcal{E}^i_{obs} \Leftrightarrow \quad \texttt{evtype(e,}In\texttt{,ex)}. \in \Pi^{\mathcal{I}^i} \tag{IT.2}$$

$$e \in \mathcal{E}^i_{instact} \Leftrightarrow \quad \texttt{evtype(e,}In\texttt{,in)}. \in \Pi^{\mathcal{I}^i} \tag{IT.3}$$

$$f \in \mathcal{F}^i \Leftrightarrow \quad \texttt{ifluent(f,}In\texttt{)}. \in \Pi^{\mathcal{I}^i} \tag{IT.4}$$

$$\mathcal{C}^{i\uparrow}(X, e) = P \Leftrightarrow \quad \forall p \in P : \texttt{initiated}(p,In,\texttt{I}) :- \texttt{occurred}(e,In,\texttt{I}), \; EX(X,\ In,\ I). \in \Pi^{\mathcal{I}^i} \tag{IT.5}$$

$$\mathcal{C}^{i\downarrow}(X,e) = P \Leftrightarrow \quad \forall p \in P : \texttt{terminated}(p, In, \texttt{I}) :- \texttt{occurred}(e, In, I), EX(X, In, I). \in \Pi^{\mathcal{I}^i} \tag{IT.6}$$

$$\mathcal{G}^{i}(X,e) = E \Leftrightarrow \quad \forall e' \in E : \texttt{occurred}(e', In, \texttt{I}) :- \texttt{occurred}(e, In, \texttt{I}), EX(X, In, I). \in \Pi^{\mathcal{I}^i} \tag{IT.7}$$

$$f \in \Delta^i \Leftrightarrow \quad \texttt{holdsat}(f, In, \texttt{I}) :- \texttt{start}(\texttt{I}). \in \Pi^{\mathcal{I}^i} \tag{IT.8}$$

In more detail, the tier each institution operates at is declared as a fact (IT.1). Events and fluents are declared as facts as usual (IT.2 to IT.4). Fluents are initiated (IT.5) and terminated (IT.6) as usual according to the state consequence function. Likewise, events occur as usual according to the event generation function (IT.7). Finally, the initial state is represented with `holdsat/3` atoms (IT.8).

The links between tiers, for 'passing up' normative fluents and norm compliance events from lower–tiers to higher–tiers for compliance checking are defined as the following ASP program:

Definition 28. ***Multi–tier Links ASP Program*** *The program Π^{mtreas} is the following ASP program:*

$$\texttt{occurred(disch(obl(A,D)),HIn,I)} :- \texttt{occurred(disch(obl(A,D)),LIn,I)}, \texttt{tier(LIn,I}_1\texttt{),tier(HIn,I}_2\texttt{)}, \texttt{I}_2 == \texttt{I}_1 + 1. \tag{MTS.1}$$

$$\texttt{occurred(disch(pro(A,D)),HIn,I)} :- \texttt{occurred(disch(pro(A,D)),LIn,I)}, \texttt{tier(LIn,I}_1\texttt{),tier(HIn,I}_2\texttt{)}, \texttt{I}_2 == \texttt{I}_1 + 1. \tag{MTS.2}$$

$$\texttt{occurred(viol(obl(A,D)),HIn,I)} :- \texttt{occurred(viol(obl(A,D)),LIn,I)}, \texttt{tier(LIn,I}_1\texttt{),tier(HIn,I}_2\texttt{)}, \texttt{I}_2 == \texttt{I}_1 + 1. \tag{MTS.3}$$

$$\texttt{occurred(viol(pro(A,D)),HIn,I)} :- \texttt{occurred(viol(pro(A,D)),LIn,I)}, \texttt{tier(LIn,I}_1\texttt{),tier(HIn,I}_2\texttt{)}, \texttt{I}_2 == \texttt{I}_1 + 1. \tag{MTS.4}$$

$$\texttt{holdsat(obl(A,D),HIn,I)} :- \texttt{holdsat(obl(A,D),LIn,I)}, \texttt{tier(LIn,I}_1\texttt{),tier(HIn,I}_2\texttt{)}, \texttt{I}_2 == \texttt{I}_1 + 1, \texttt{start(I)}. \tag{MTS.5}$$

$$\texttt{holdsat(pro(A,D),HIn,I)} :- \texttt{holdsat(pro(A,D),LIn,I)}, \texttt{tier(LIn,I}_1\texttt{),tier(HIn,I}_2\texttt{)}, \texttt{I}_2 == \texttt{I}_1 + 1, \texttt{start(I)}. \tag{MTS.6}$$

$$\begin{aligned}\mathtt{initiated(obl(A,D),HIn,I)} :- &\ \mathtt{initiated(obl(A,D),LIn,I)},\\ &\ \mathtt{tier(LIn,I_1),tier(HIn,I_2)},\\ &\ \mathtt{I_2 == I_1 + 1, start(I)}.\end{aligned} \tag{MTS.7}$$

$$\begin{aligned}\mathtt{initiated(pro(A,D),HIn,I)} :- &\ \mathtt{initiated(pro(A,D),LIn,I)},\\ &\ \mathtt{tier(LIn,I_1),tier(HIn,I_2)},\\ &\ \mathtt{I_2 == I_1 + 1, start(I)}.\end{aligned} \tag{MTS.8}$$

$$\begin{aligned}\mathtt{terminated(obl(A,D),HIn,I)} :- &\ \mathtt{terminated(obl(A,D),LIn,I)},\\ &\ \mathtt{tier(LIn,I_1),tier(HIn,I_2)},\\ &\ \mathtt{I_2 == I_1 + 1, start(I)}.\end{aligned} \tag{MTS.9}$$

$$\begin{aligned}\mathtt{terminated(obl(A,D),HIn,I)} :- &\ \mathtt{terminated(obl(A,D),LIn,I)},\\ &\ \mathtt{tier(LIn,I_1),tier(HIn,I_2)},\\ &\ \mathtt{I_2 == I_1 + 1, start(I)}.\end{aligned} \tag{MTS.10}$$

This concludes our short introduction of multi–tier institution operationalisation in ASP.

5.1.5. EXECUTED CASE STUDY

In this section we execute our case study, comprising the soundsensing institution which is governed by a governmental institution, formalised as a multi–tier institution. That is by executing the multi–tier institutions' corresponding ASP programs, $\Pi^{\mathcal{I}^{SS}}$, $\Pi^{\mathcal{I}^{GI}}$, together with the program linking them Π^{mtreas}, the base semantics program $\Pi^{base(k)}$ and an observable trace program $\Pi^{trace(k)}$, in the last two cases for a trace of observable events of length k. In our case, the trace program comprises an event sequence with three elements, represented as the following ASP facts:

```
compObserved(ex_enter(ada, street_b), 0).
compObserved(ex_request_location(ada), 1).
compObserved(ex_request_location(bertrand), 2).
```

In this short trace, first Ada enters a new location, `street_b`, then Ada's location is requested and finally Bertrand's location is requested. For this short trace, three cases of non–compliance of the soundsensing institution with the governmental institution are discovered, as depicted in figure 5.2.

Firstly, when Ada enters a new location due to an erroneous rule in the soundsensing institution (a designer error), a generic norm violation event occurs in the soundsensing institution (the first state transition). Due to the generic norm violation event occurring, the soundsensing institution imposes an obligation for Ada to pay a fine in state S_1. However, the governmental institution only recognises actual norm violation events as causing a generic norm violation event and obliges an actual norm is violated before an obligation to pay a fine is imposed. Consequently, the governmental institution's higher–order norm for there to be a norm violation before an obligation to pay a fine is imposed is violated in the transition from S_1 to S_2.

Secondly, in the soundsensing institution agents are unconditionally prohibited from turning their microphone off, however the governmental institution prohibits such a prohibition when an agent enters a private location. When Ada enters a private location (`street_b`)

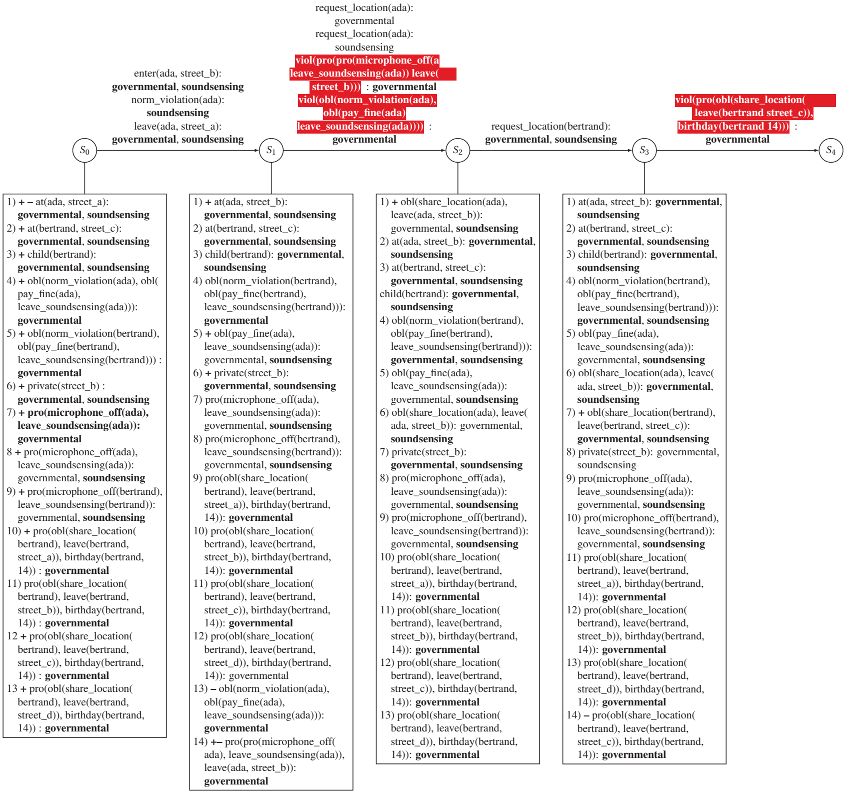

Figure 5.2: Results for a trace, the originating institutions for an event or a fluent are in **bold**, newly initiated fluents are denoted with a **+** and terminated fluents which will cease to hold in the next state are denoted with **–**. Violated second–order norms in the second–tier governmental institution are **highlighted** .

there is still a prohibition on her from turning her microphone off. Consequently, the prohibition on prohibiting Ada from turning her microphone off when she is in a private location is violated during the transition from state S_1 to S_2.

Thirdly, when an agent is requested to provide their location the soundsensing institution obliges them to do so, but this is forbidden by the governmental institution if the agent is a child, such as Bertrand. In state S_3 the soundsensing institution obliges Bertrand, a child, to share his location. Consequently, in the final state transition the governmental institution's prohibition on obliging Bertrand to share his location is violated. This concludes our presentation of non–compliance for the case study in this chapter. In the final part of this section we will discuss the approach we take to automatically resolving non–compliance, by using Inductive Logic Programming.

5.1.6. INDUCTIVE LOGIC PROGRAMMING: A BRIEF OVERVIEW

We view the problem of revising lower–tier institutions to be compliant with higher–tier institutions as a theory revision (TR) problem which can be solved using Inductive Logic Programming (ILP). ILP [180] is a machine learning technique concerned with the induction of logic theories that generalise (positive and negative) examples with respect to a prior background knowledge. In non–trivial problems it is crucial to define the search space accurately. This is done by a *language bias*, that can be expressed using the notion of mode declarations [180], describing the structure of the elements in the target theory. In the case presented here, we want to find the ASP rules that contain certain elements in the head and body. So we will have head and body mode declarations.

An *ILP theory revision task* is a tuple $\langle P, B, M\rangle$ where P is a set of conjunctions of literals, called *properties*, B is a normal program, called the *background theory*, M is a set of mode declarations describing the form that rules in the revised theory can take and $s(M)$ is the set of rules adhering to M. A theory H, called a *hypothesis*, is an inductive solution for the task $\langle P, B, M\rangle$, if (i) $H \subseteq s(M)$, and (ii) P is true in all the answer sets of $B \cup H$.

Our approach to revising a lower–tier institution in a multi–tier institution to be compliant is based on the introduction of new rules, and deleting and revising existing ones. As discussed in [45], non–monotonic inductive logic programming can be used to revise an existing theory. The key concept is that of *minimal revision*. In general, a TR system is biased towards the computation of theories that are similar to a given revisable theory. The difference between two programs T and T' is denoted as $c(T, T')$.

The theory T', called a *revised theory*, is a *TR solution* for the task $\langle P, B, T, M\rangle$ with distance $c(T, T')$, iff (i) $T' \subseteq s(M)$, (ii) P is true in all the answer sets of $B \cup T'$, (iii) if a theory S exists that satisfies conditions (i) and (ii) then $c(T, S) \geq c(T, T')$, (i.e. minimal revision).

5.2. REVISING INSTITUTIONS FOR COMPLIANCE

In this section we give the details of this chapter's main contribution: a system for revising a lower–tier institution to be compliant with a higher–tier institution in a multi–tier institution. In particular, we are interested in revising the institution which the system user (an institutional designer) has the power to effect change. We call this institution to be revised a *mutable* institution. We are interested in revising a mutable institution to meet two properties:

- **Success** meaning that a formerly non–compliant institution for an event–trace is compliant for the same event trace after being revised. This means when normative fluents are obliged to be imposed they are, and conversely any prohibited normative fluents are not imposed.

- **Minimality** is a requirement for any revision to minimise the change in *consequences* of the new institution compared to the old one. That is, following changes to the institution the institution's states are as close as possible to the states prior to the change(s) for a trace of events. To give an example, the soundsensing institution prohibits agents to turn their cellphone microphone off, whilst the governmental institution prohibits such a prohibition in areas deemed 'private'. In this case, an institution revision can be successful by removing the soundsensing institution's prohibition altogether, but only successful *and* minimal by removing the prohibition in just those cases where an agent is at a private location.

We instantiate the problem of revising a mutable institution as an ILP theory revision task in Section 5.1.6. Then, we take a computational approach to solving the ILP theory revision task by performing *inductive* search in ASP [47]. Inductive search is achieved by transforming the mutable institution represented in ASP to an ASP representation encoding the space of ILP theory revisions and enabling different revisions to be tried. We describe our computational approach using ASP in Section 5.2.2, and the implementation and revision results for our case study in Section 5.2.3.

5.2.1. REVISING INSTITUTIONS TO BE COMPLIANT IS AN ILP THEORY REVISION TASK INSTANCE

In this section, we define the revision for compliance task as an ILP revision task according to the revision for compliance requirements outlined previously. We begin by formally defining the search space of possible revisions with *mode declarations*. Mode declarations define the literals that can appear in the head and body of rules. In the case of revising a mutable institution in a multi–tier institution, the mode declarations describe the valid rules for: generating non–normative institutional events, initiating and terminating domain fluents, and given the mutable institution is the ith–tier, initiating and terminating $i$thorder normative fluents (i.e. restricted to only initiating/terminating a normative fluent f if it is not in the language of norms $\mathcal{N}|_{Pr^{i-1}}$ of the tier $i-1$ below).

Definition 29. ***Mode Declarations.*** *Let $\mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{G}^i, \mathcal{C}^i, \Delta^i \rangle$ be a mutable institution for which In is a unique label. The mode declarations for $\mathcal{I}^i$ are a pair $M = \langle M^h, M^b \rangle$ where M^h is the set of head mode declarations and M^b the set of body mode declarations, defined as:*

$$\begin{aligned}
M^h &= \{\texttt{initiated}(f, In, \texttt{I}), \texttt{terminated}(f, In, \texttt{I}) : f \in \mathcal{F} \setminus \mathcal{N}|_{Pr^{i-1}}\} \cup \\
&\quad \{\texttt{occurred}(e, In, \texttt{I}) : e \in \mathcal{E}^i_{instact}\} \\
M^b &= \{\texttt{holdsat}(f, In, \texttt{I}), \neg\texttt{holdsat}(f, In, \texttt{I}) : f \in \mathcal{F}^i\} \cup \\
&\quad \{\texttt{occurred}(e, In, \texttt{I}) : e \in \mathcal{E}^i\}
\end{aligned}$$

The set of *compatible rules* with the head and body mode declarations are also required to contain one and only one event in the body. The compatible rules are defined as:

Definition 30. ***Compatible Rules.*** *Let $M = \langle M^h, M^b \rangle$ be the mode declarations for a mutable institution $\mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{G}^i, \mathcal{C}^i, \Delta^i \rangle$. An ASP rule $\texttt{l}_0 : - \texttt{l}_1, ..., \texttt{l}_\texttt{n}$. where $n \in \mathbb{N}$ is compatible with M iff $l_0 \in M^h$, $\forall i \in [1, n] : l_i \in M^b$ and $|\{l_1, ..., l_n\} \cap \{\texttt{occurred}(e, In, \texttt{I}) : e \in \mathcal{E}^i\}| = 1$. The set of all compatible rules with M is $s(M)$.*

Having described the search space of revisions, a theory revision task *TR* needs to be instantiated with the properties P that a solution must meet. These properties are typically positive examples (formulae that are true following a revision) and negative examples (formulae that are false following a revision). In our case we are only interested in supplying negative examples, stating that non–compliance is eradicated in a solution to *TR*. The negative examples in P are represented as ASP integrity constraints requiring a revised mutable institution is compliant with all higher–order norms it can violate – including those it does not violate before revision – ensuring revision does not cause further non–compliance.

Definition 31. ***Compliance Properties***. *Let $\mathcal{I}^i$ be a mutable institution and $\mathcal{I}^{i+1} = \langle \mathcal{E}^{i+1}, \mathcal{F}^{i+1}, \mathcal{C}^{i+1}, \mathcal{G}^{i+1}, \Delta^{i+1} \rangle$ be the institution with unique name In^{i+1} governing $\mathcal{I}^i$ where $i \in \mathbb{N}$. The compliance properties for $\mathcal{I}^i$ is the set of constraints:*

$$P = \{: -\,\texttt{occurred(viol(}n\texttt{)},In^{i+1}\texttt{,I)},\texttt{instant(I).} : n \in \mathcal{F}^{i+1}_{norm}\}$$

We can now instantiate an ILP theory revision task, as a *compliance* theory revision task in a multi–tier institution according to the previous definitions:

Definition 32. ***Compliance Theory Revision Task***. *Let $\mathcal{I}^i$ be a mutable institution in the multi–tier institution $\mathcal{M}$. An ILP theory revision task $TR = \langle P, B, T, M \rangle$ is a compliance theory revision task for $\mathcal{I}^i$ iff: (a) P is a set of compliance properties for $\mathcal{I}^i$, (b) B is the normal program comprising (i) a multi–tier reasoning program Π^{mtreas}, (ii) the timeline program $\Pi^{trace(k)}$ and (iii) the institution representation program $\Pi^{\mathcal{I}^j}$ for each institution $\mathcal{I}^j$ in $\mathcal{M}$ apart from the mutable institution $\mathcal{I}^i$, (c) T is the institution representation program $\Pi^{\mathcal{I}^i}$ for the mutable institution $\mathcal{I}^i$, and (iv) M is the set of mode declarations for $\mathcal{I}^i$.*

As outlined previously, we require solutions to theory revision to minimise the revision cost in order to remain as close to an institution designer's original intentions as possible. More precisely, the requirement is that the changed, mutable, institution's model for a composite trace contains as many similarities between states compared to before the changes were made (i.e. minimising the changes to consequences). We derive the cost of revision from the changes in consequences rather than the number of rule changes – as used in [154] – since due to non–monotonicity, as the changes in consequences between two versions of a mutable institution increases, the number of rule changes does *not* necessarily monotonically increase. The changes in consequences are the number of added and deleted fluents in the answer set for $B \cup T$ compared to the answer–set $B \cup T'$ for some revised institution T' (i.e. the symmetric set difference between the answer–sets for $B \cup T$ and for $B \cup T'$).

Definition 33. ***Theory Revision Cost*** *Let $TR = \langle P, B, T, M \rangle$ be a compliance theory revision task for a mutable institution $\mathcal{I}$ with unique label In, T' be a solution to TR, ans be the answer–set for $B \cup T$ and ans' be the answer–set for $B \cup T'$ and $\oplus$ be the set symmetric difference operation. The cost $c(T, T')$ is defined as:*

$$c(T, T') = \left| \left\{ \ f = \texttt{holdsat}(p, In, i) : i \in \mathbb{N},\ f \in ans \oplus ans' \ \right\} \right|$$

Thus, a solution to an ILP theory revision task, instantiated as revision for compliance, $TR = \langle P, B, T, M \rangle$ is a search problem. The search space consists of all possible revised versions of a mutable institution theory (normal program) and a solution T' to the theory revision task is one which ensures the mutable institution is now compliant for a particular trace (meets the properties in P) and minimal such that there is no revised theory S which is 'more minimal' ($c(T, S) \geq c(T, T')$). This requires searching all possible solutions and comparing them. In order to perform this search we need a way to try different changes to T and find optimal solutions. In the next subsection we show how we can use ASP to perform this search.

5.2.2. SOLVING ILP INSTITUTION REVISION IN ASP

Based on [154] we use inductive search in ASP to solve an ILP theory revision task $TR = \langle P, B, T, M \rangle$ instantiated as institutional revision for compliance. The approach we take is to transform the theory to be revised T (a mutable institution) into an ASP program where different changes to the theory can be tried (body literal and rule addition and deletion) that fit into the space of possibilities $s(M)$. This is by redefining certain predicates in rules as *undefined*, rather than being a necessary element of a rule. Such undefined predicates are called *abducibles*.

We call this program, with previously fully–defined predicates now defined as abducibles, the *revision* program Π^{rev}. The background theory B remains unchanged and provides both the unchangeable parts of the multi–tier institution and multi–tier reasoning. The background theory allows the effects of different revisions to be determined. The properties to be met, P, constrain any revisions found by the ASP program Π^{rev} to result in a compliant institution. The cost measure between a revisable T and revised theory T', $c(T, T')$ is encoded as an ASP optimisation statement. Computing the answer–sets for these components as a single ASP program explores the search space, with each answer–set representing an outcome (revised theory) that meets the properties P and with those that minimise the difference (changes in consequences) ranked highest and presented to the user for selection. The advantage of this approach is that the representation and reasoning for the non–revisable portions of the multi–tier institution are encoded as the same ASP programs for the computational and revision framework requiring no re–implementation.

In order to go from a revisable theory T representing a mutable institution to a revision program Π^{rev}, we need to alter T in some way such that adding new rules and changing existing rules can be tried by the new program with each answer–set corresponding to different revised theories. The approach we take, as in [47, 154], is to introduce *abducible* predicates which represent the different revision operations and are selected by the program for inclusion in answer–sets. If an abducible is selected for answer–set inclusion then the effect is to perform the revision operation the abducible represents. The abducibles have the form $\texttt{rev}(In, \texttt{i}, \texttt{details}(...))$ conveying to the user the revision operation described in $\texttt{details}(...)$ (e.g. a rule deletion operation) is carried out on a rule with label i in institution In. To give a simple example the rule $\texttt{l}_0 :- \texttt{l}_1$. cannot be selected for deletion by an ASP program, but we can modify it to become $\texttt{l}_0 :- \texttt{l}_1, \texttt{not rev}(In, \texttt{i}, \texttt{details}(\texttt{rDel}))$. meaning if the abducible $rev(In, i, \texttt{details}(\texttt{rDel}))$ is included by the program in an answer–set the effect is to delete the rule i by ensuring the body is never true. The selection of revision tuples for inclusion in an answer–set is encoded in the ASP revision program using the ASP *choice* construct of the form $\{rev(In, \texttt{i}, \texttt{details}(...))\}$.

Each type of revision operation (rule and body literal addition and deletion) requires a different abducible and set of rules in the ASP revision program Π^{rev}. In Table 5.6 we describe the details of the different rules for trying revisions and the transformation from a revisable theory T to a revision program Π^{rev} using In to represent an institution's name, i to represent a rule identifier (e.g. an integer) and $B^{+}_{-}(l)$ to represent whether a literal l is

Rules Describing Institution Changes	**Explanation**
$l_0 :- l_1, ..., l_n,$ `rev`$(In,\ i,$`details(rDel)`$)$. $\{$`rev`$(In,\ i,$`details(rDel)`$)\}$.	*Rule deletion:* Existing rules are extended with an abducible `rev`$(In,\ i,$`details(rDel)`$)$, which when included in an answer–set has the effect of deleting the rule with index i.
$l_0 :- l_1, ..., l_{j-1},$ `try`$(i, j, B^+_-(l_j), l_j),$ $l_{j+1}, ..., l_n.$ `try`$(i, j, B^+_-(l_j), l_j) :- l_j,$ `not rev`$(In, i,$`details(bDel,`$j))$. `try`$(i, j, B^+_-(l_j), l_j) :-$ `rev`$(In, i,$`details(bDel,`$j))$. $\{$`rev`$(In,$`i,details(bDel,j)`$)\}$.	*Body literal deletion:* Each body literal l_j of an existing rule is replaced with the literal `try/4` for trying to delete the body literal l_j. When the abducible `rev`$(In, i,$`details(bDel,`$j))$ is included in an answer–set the effect is to make the try literal true and thus effectively delete the literal l_j, otherwise the try literal is only true when l_j is true (effectively keeping l_j).
$l_0 :-$ `rev`$(In,\ i,$`details(rAdd)`$), l_1, ..., l_n.$ $\{$`rev`$(In,\ i,$`details(rAdd)`$)\}$.	*Rule addition*: Including the abducible `rev`$(In,\ i,$`details(rAdd)`$)$ has the effect of including the rule with index i in the program.

$l_0 :- l_1, l_2, ..., l_n,$
`extension`$(i, l_0, l_{n+1}, B^+_-(l_{n+1}))$.
`extension`$(i, l_0, l_{n+1}, B^+_-(l_{n+1})) :-$
`not rev`$(In, i,$`details(bAdd,`
$B^+_-(l_{n+1}), l_{n+1}))$.
`extension`$(i, l_0, l_{n+1}, B^+_-(l_{n+1})) :-$
`rev`$(In, i,$`details(bAdd,`
$B^+_-(l_{n+1}), l_1), l_1.$
$\{$`rev`$(In, i,$`details(bAdd,`
$B^+_-(l_{n+1}), l_{n+1}))\}$.
	Body literal addition: Existing rules are appended with `extension/4` predicates for each body mode literal a rule can be extended with. Including the abducible `rev`$(In, i,$`details(bAdd,`$pos, l_1)$ in an answer–set has the effect of extending the rule with index i with the body literal l_1 (constraining the rule). That is, adding the revision predicate to an answer set makes the extension predicate true only when the literal with the specified variable bindings are true, effectively adding a constraint/body–literal to the rule. Otherwise, the extension predicate is always true (no constraint is tried for addition).

Table 5.6: Explanation of how abducible revision predicates can (re–)define institutional rules for finding revisions of the institution *In*

positive or negative.

Finally, the cost $c(T, T')$ between two theories is encoded as an ASP optimisation constraint causing the ASP program to only present answer–sets that are minimal in the changes to consequences between $T \cup B$ and $T' \cup B$, which we also extend with a secondary preference for revisions that *generalise* the institution (deleting body literals and rules) rather than specialising (adding new body literals and rules). The optimisation statement is given below where $X@n$ represents the priority n of minimising the numerical value X, `difference/1` measures the difference between the states in the answer–set for the institution before and after revision (in terms of added and removed fluents for each state), `rAdd/1` counts the rule additions, `bAdd/1` the body additions, `bDel/1` the body deletions and `rDel/1` the rule deletions.

```
#minimize {D@5: difference(D); RA@4: rAdd(RA); BA@3: bAdd(BA);
           BD@2: bDel(BD); RD@1: rDel(RD)}.
```

5.2.3. IMPLEMENTATION AND RESULTS

A prototype system for revising a lower–tier institution to be compliant with a higher–tier is implemented according to the description in the preceding sections.[1] The implementation is a compiler written in Java which, as depicted in Figure 5.3, takes as input the mutable institution the institution designer has the power to effect change represented in ASP (the mutable institution program $\Pi^{\mathcal{I}^i}$) and outputs a revision program Π^{rev}. The revision program is then put together with compliance properties to be met by revisions, revision cost minimisation optimisations and the background theory to remain unchanged (the non–mutable institutions, the timeline program and the multi–tier institution reasoning). An answer–set solver applied to the composition of these programs then produces minimal revision suggestions for compliance (answer sets). The suggestions are passed to a user who selects and applies a set of revisions, resulting in a compliant institution represented as an ASP program.

In addition to the system presented in this chapter, the ASP compiler also addresses an apparent lack of re–usability of institutions (e.g. using the same institution for different sets of agents) due to their propositional nature. Rather than taking just propositional institutions as input, the compiler also takes *first–order* institution theories containing variables in the head and body of rules, together with bindings and monadic predicates to denote types. To give an example, `agent(ada)` denotes `ada` is of type agent and `agent(X)` denotes the variable `X` is any ground term of type agent. Thus, a designer does not need to write a new propositional institution for the case where a new agent, Charles, joins the institutionalised society with all the norms and domain fluents that are about Charles. Instead, a fact `agent(charles)` can be added stating Charles is of type agent. In turn, the compiler takes these more re–usable first–order institution theories as input and outputs a first–order institution revision program that tries different variable bindings between head and body literals' variables of the same type.

For our case study, we have used our prototype compiler to produce a revision program for sub–sets of the compliance problem. That is, dividing the program up into smaller parts and resolving one case of non–compliance at a time for tractability, and testing all revision suggestions together at the end to confirm they are consistent. Some of the minimal and successful revisions found are given below (we keep to those we find most intuitive).

The first change suggested addresses the issue of non–compliance due to an obligation

[1]The prototype, multi–tier reasoning in ASP and the examples used in this chapter can be found at `https://sourceforge.net/projects/multitierinstitutionlearning/files/`.

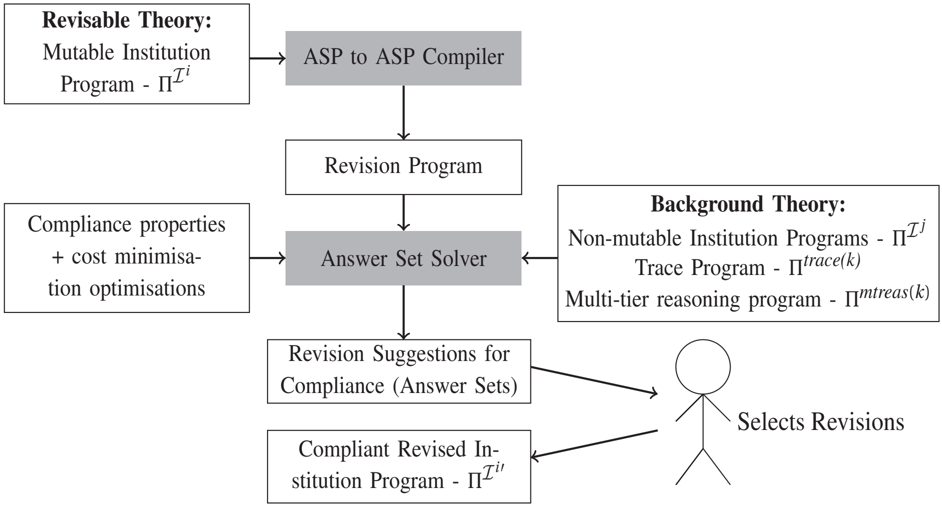

Figure 5.3: Overview of using the implemented compiler and the multi–tier institution framework to resolve non–compliance.

to pay a fine being imposed by the tier–1 soundsensing institution when an agent enters a new area. Non–compliance occurs, because an agent entering a new area triggers a norm violation event in the first tier institution regardless of whether a norm has been violated, whilst the second tier obliges that a norm is genuinely violated before a fine is imposed. The revision suggestion is to delete the rule in the first–tier soundsensing institution causing a norm violation event to occur when an agent enters an area:

```
occurred(norm_violation(Agent0), soundsensing, I) :-
      agent(Agent0), instant(I),
      occurred(enter(Agent0, Location0), soundsensing, I),
      location(Location0).
```

The second issue is that children (people under the age of 14) are obliged to share their location when requested, but this is prohibited by the tier–2 governmental institution. The following suggestion is one of several minimal changes found to ensure the non–compliant obligation is not imposed on children. An additional constraint is placed that an agent, `Agent2`, is not a child and the variable `Agent2` is bound to the variable `Agent0` denoting the agent who would be obliged to share their location. This means that the obligation can not be imposed on a child. The new variable `Agent2` is introduced since the implementation relies on using unique variables for all literals and then systematically trying different optional bindings between the variables (or no bindings). The new rule is:

```
initiated(obl(share_location(Agent0),
              leave(Agent0, Location0)), soundsensing, I) :-
   occurred(request_location(Agent1), soundsensing, I),
   holdsat(at(Agent1, Location0), soundsensing, I),
   not holdsat(child(Agent2), soundsensing, I), Agent0 = Agent1,
```

```
   Agent0 = Agent2, agent(Agent0), agent(Agent1), agent(Agent2),
   location(Location0), instant(I).
```

Finally, the tier–2 governmental institution prohibits a prohibition on an agent to turn their microphone off when they are in a private area. Yet, the tier–1 soundsensing institution always prohibits turning a microphone off until the agent leaves the system (the prohibition exists in the initial state). The revisions found are not to delete the rule initiating a prohibition in the tier–1 soundsensing institution's initial state, but instead, to terminate the prohibition when an agent enters a private area and then initiate it again when they leave. Although the revision adds two rules, it is minimal in the outcome of the tier–1 soundsensing institution since there is still a prohibition on turning the microphone off in all other cases where it is allowed by the tier–2 governmental institution:

```
terminated(pro(microphone_off(Agent0), leave_soundsensing(Agent0)),
   soundsensing, I) :-
   occurred(enter_private(Agent2), soundsensing, I),
   agent(Agent2), agent(Agent0), Agent0=Agent2, instant(I).
initiated(pro(microphone_off(Agent0), leave_soundsensing(Agent0)),
   soundsensing, I) :-
   occurred(leave_private(Agent2), soundsensing, I), agent(Agent2),
   agent(Agent0), Agent0=Agent2, instant(I).
```

5.3. RELATED WORK

There has been much work on norm change in normative systems, however, as far as we are aware we are the first to propose a way to revise institutions to be *compliant* with other institutions in a multi–tier institution.

The most closely related work is by Li et al. [154, 155] who also uses inductive search in ASP to resolve an ILP *theory revision* task. Unlike us, their focus is on resolving norm conflicts between multiple institutions governing a group of agents (e.g. when an agent is prohibited to perform an action by one institution and obliged by another) and later the general case of debugging ASP programs [157]. In comparison, we focus on revising *non–compliance* between lower–tier and higher–tier institutions in a *multi–tier* institution. Our proposal is based on Li et al. and extended to revising an ith–tier institution by adding new rules or modifying/deleting pre–existing rules to impose ith–order norms. We also extend the work to revising with minimal changes in the *consequences* of a revised institution (as opposed to changed rules), finally we look at the creation and deletion of existing rules which in our case study provides more minimal changes in the consequences compared to rule modification.

Vasconcelos et al. [231] have proposed a technique for revising conflicting norms based on first–order unification. Their proposal provides a fine–grained way to revise obligation/permission/prohibition predicates' terms. For example, an obligation to be in an area that overlaps with a prohibited area is revised by changing the obliged/prohibited areas for an agent to be in. In contrast to our work, their focus is on modifying the obligation/permission/prohibition predicates and not with adding/removing/modifying *rules* to meet a particular property (compliance between institutions in our case).

Governatori and Rotolo [102] propose a way to use a defeasible logic to modify legal systems by introducing new norms which derogate, abrogate and annul norms using defeasible rules. Central to their proposal is the idea of a legal system being *versioned* and having two timelines: the versioning timeline and the timeline of the legal system's evolution (i.e. which norms are imposed and when). We only consider the latter timeline, the evolution of an institution (in our case during pre–runtime model checking) and focus on diagnosing *causes* of non–compliance between institutions rather than assuming it is known what the new information (rules) is.

Finally, on the more conceptual and theoretical side, Boella et al.[25] look at how to classify different systems of norm change by investigating a set of rational norm change postulates. Specifically, they look at normative system change to incorporate new *conditional* norms in input/output logics and they investigate the set of consistent postulates for different input/output logics. Again, this work also presupposes which conditional norms should be added to the normative system/institution, thus any system meeting these postulates is quite different from our proposal.

5.4. DISCUSSION

In this chapter we contributed an implemented automated system for revising a lower–tier institution's regulations to be compliant with the regulations of a higher–tier institution it is governed by. The proposal addressed a problem created by pervasive legal artefacts in the social world, where on the one hand institutions are used to govern other institutions in a vertical governance structure we call multi–tier institutions, creating the potential for non–compliant regulations. On the other hand, revising institutions' regulations to be compliant is non–trivial due to their inherent complexity.

Our proposal takes our previous formal and computational framework [144] for determining the compliance of institutions in multi–tier institutions. Then, viewing the problem of revising an institution to be compliant as an instance of an ILP (Inductive Logic Programming) theory revision task, we use inductive search in ASP based on [154] to solve the ILP theory revision task for compliance. Inductive search in ASP is performed by translating, using an implemented compiler, from an ASP representation of an institution that needs to be revised to be complaint where revisions cannot be tried and searched for, to an ASP representation where revisions can be tried and thus revisions for compliance determined. Then, our system goes about finding revisions that are successful in resolving non–compliance and minimal in the changes to the institution's consequences thus keeping the changed institution as close as possible to the institution designer's original intentions.

The system for revising institutions, for tractability, considers a fragment of the search space of revisions: modifying and deleting existing rules and extending a single mutable institution with a limited number of rules. The successful and minimal revisions that do exist (if any) within the space explored are guaranteed to be found. However, there may be more minimal revisions that result in a well–formed institution (according to our representation of institutions) outside of this space, but this space is bigger and takes longer to explore.

We consider this a problem that is important to address. Firstly with formal analysis of the complexity of the full problem. Secondly, by studying the applicability of various heuristics to the full search problem (e.g. genetic algorithms) that cannot guarantee a minimal solution is found (i.e. in the case of genetic algorithms instead converging on local optima) but can help resolve tractability issues. As yet, it is unclear which heuristics are appropriate and how they can be incorporated into ILP revision as abducible search in ASP, presenting an interesting challenge for future work.

6

Formalising Institutional Enactment Validity

One of the major problems encountered in time travel is not that of becoming your own father or mother. There is no problem in becoming your own father or mother that a broad–minded and well–adjusted family can't cope with. [...] The major problem is simply one of grammar.

Douglas Adams, The Restaurant at the End of the Universe

This chapter is based on the following paper:
King, T. C., Dignum, V., & Jonker, C. M. (2016). 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. (pp. 3 – 11). IOS Press. http://doi.org/10.3233/978–1–61499–672–9–3 [143]

Previously, Chapter 3, Chapter 4 and Chapter 5 covered soft constraints for institutional design governance and how to comply with these soft constraints. In this chapter we look at *hard institutional constraints* for institutional enactment governance, which make the social actions of enacting institutional changes possible. This chapter makes the following contributions:

- A novel representation of institutions for which changes can be enacted according to secondary legal rules regulating rule change. In particular, secondary legal rules represented as counts–as rule that make rule change possible and are conditional on an extended notion of social context that includes conditions on past and present states, institution versions and hypothetical effects of rule change.
- A novel semantics for reasoning about rule changes ascribed by rule–modifying counts–as rules.

This chapter looks at institutional governance from the *can* perspective, in terms of which rule changes can be enacted in which contexts and by whom. Specifically, we look at the kinds of rules found in administrative law, which describes the procedures that must be followed for rule change to take place and specifies what *types* of rule change are possible. These are hard institutional constraints on rule enactment, in the sense that they make rule changes (im)possible.

Whilst constitutive counts–as rules ascribe a social reality and how it changes over time, they themselves are also subject to evolving over time when they are modified, such as by a legislative body. Rule–modifying counts–as rules regulate changes to counts–as rules themselves, "A counts–as *modifying a rule* in context C", describing the legislative procedure and the possible modification types. Rule–modifying counts–as rules are also modifiable according to the rule modification regulations. In ([100, 102]) a defeasible logic formalises legal rule change over time but not rule change regulation. In ([28]) counts–as rules that regulate rule modifications are formalised, but not in a temporal setting. However, thus far there has been no attention paid to formalising rule change regulated by counts–as rules in a temporal setting.

This raises the question, in a temporal setting "how can we formalise practical reasoning to determine when institutional rule changes count–as legally valid rule change enactments? should we define when rule changes count–as legal and valid rule changes?" To give some examples:

- Based on ([204, Art. 71]) parliament voting to enact a bill counts–as enacting it within one month. Rule change affects institutional states (e.g. enacting a bill obliging fences to be painted white, or changing what counts–as parliament); rule changes are conditional on the institutional state (e.g. whether a body constituting parliament has voted).
- The UK government voted to *retroactively* require UK residents in a business partnership abroad to pay tax ([192, Sec. 58]), criminalising people in the past. Criminalising retroactive modifications are not possible according to the European Convention of Human Rights ([48, Art. 7]). Rule change affects institutional states (e.g. criminalising people in the past); rule change is conditional on how it would change

institutional states (e.g. if the change would criminalise people in the past then the change is not possible).

- A monarch or parliament can enact and repeal laws. The monarch enacts a law obliging all fences are painted white. The parliament retroactively repeals the power for the monarch to enact laws, reversing the fence–painting law enactment. Retroactive rule change affects past rule–modifying counts–as rules; past rule modifications can be unravelled due to a retroactive modification.

An interdependency exists between the counts–as rules that construct a social reality and rule–modifying counts–as rules. Changing counts–as rules affects the past/present/future institutional context and can change the modifications which happened in the past up until the present; rule modifications are conditional on the past/present/future institutional context and the hypothetical rule change effects. Whether a rule change counts–as a *legal* rule change requires assessing the context in which the change takes place – comprising both the present state of affairs and the potential rule change effects affecting whether a rule change is legal in the first place.

In this chapter we address the question "how can we formally define when legally valid institutional change enactments occur?". The question is addressed by contributing the first formalisation of rule change ascribed by counts–as rules accounting for the aforementioned cases in a temporal setting. We introduce the notion of temporal rule–modifying counts–as rules, "A counts–as modifying a rule in the past/present/future in context C", which ascribe rule changes. To account for institutions dually evolving from one social reality to the next built by counts–as rules, as well as from on counts–as rule set to another, we adopt social reality (state) and institution version timelines. We enhance context from referring to a built social reality, to also a past social reality in different institution versions and the hypothetical effects of rule change. Rule modifications can potentially affect past modifications by changing the social context or contradicting previous modifications, we posit that the most recent modifications are prioritised and can override past modifications.

The rest of this chapter is organised as follows. We begin by comparing what follows with the InstAL framework which we use as a basis for our novel formalisation, and the formalisation used in the previous chapters in 6.1. Then we present the formal representation 6.2. The novel formal semantics are given in 6.3. We apply our framework to real case studies and examples used to test difficult edge cases in 6.4. Finally, we compare with closely related work in 6.4.1 and conclude with wider implications and directions for future work in 6.5.

6.1. COMPARISON WITH INSTAL

In this chapter we extend institutions first formalised in the InstAL (Institution Action Language) framework ([42, 43]) where rules cannot be modified and rule change cannot be regulated, to be modifiable and modification–regulating. In InstAL an institution is a signature comprising events, fluents and regulatory rules. An event is either institutional providing a social interpretation (e.g. a person paying tax) of other institutional events and observable events or is observable and exogenous to the institution corresponding to brute facts (e.g. the event we call a person paying tax). Fluents describe/prescribe the domain (e.g. a person is a business partner who is obliged to pay tax). We view all rules as "A *counts–as* B in context C", stating an event (the 'A') in a social context ('C') causes further institutional events to occur or different fluents to hold in the next state (the 'B's'). An InstAL institution contains rules regulating its evolution from one social reality (state) to the next, but the rules remain static.

We extend institutions comprising counts–as rules to counts–as rules which are modifiable and ascriptively regulate rule changes. Rule modifications activate and deactivate rules in the past/present/future, analogous to enacting regulatory changes. Rule modifications are regulated with counts–as rules, "an event A counts–as a *rule–modifying* event B in context C", which ascribes the actions counting–as past/present/future rule modifications in a particular context.

Importantly, unlike in InstAL, institutions with modifiable rules do not just evolve from one state to the next according to their own counts–as rules. Rather, institutions evolve along two timelines, from one version comprising rules to another version comprising different rules. Each version evolves over time from state to state according to its rules. When a rule change occurs according to rule–modifying counts–as rules in one version, the institution evolves to the next version potentially comprising different (active) counts–as rules at different times. For example, if in one institution version on Wednesday a rule is *retroactively* added on the preceding Monday then that version's past does not change. Instead, the institution evolves to a new version where the rule is active from the Monday potentially causing the version to evolve differently from then onwards. Institutions evolve from version to version where one versions can evolve from one state to another differently according to different rules.

Contexts in counts–as rules are also extended. Contexts can be conditional on the present state but also past institution versions and states. This allows testing potential retroactive rule modification effects. For example, a retroactive rule change is ascribed by a government voting for the change in the context that it does not criminalise people in the past *compared to the previous version's past*. To summarise we extend institutions to evolve along rule version and state timelines according to counts–as rules conditional on contexts comprising past versions and states, and potential rule change effects.

In comparison to Chapter 3, Chapter 4 and Chapter 5 we are only interested in institutions ascribing non–deontic social facts. For example, social actions such as voting in parliament or enacting a rule change, and states of affairs such as someone being a tax payer. In other words, we are interested in the hard institutional constraints that make social actions possible according to counts–as rules. We are not interested in obligations and prohibitions which prescribe the ideal social actions and states of affairs. Consequently, unlike in the previous chapters we provide no specific representation for deontic positions nor a semantics to determine when they are violated. Moreover, we do not consider whether the effects of regulatory rules (i.e. the obligations and prohibitions they impose) are good or bad and hence we do not capture higher–order norms. We use counts–as rules, but unlike in Chapter 3 and Chapter 4 we do not capture links between concrete and abstract regulations according to counts–as rules. Rather, we capture the notion of governance in terms of counts–as rules making institutional change enactments regular and *possible*, hence deontic elements are not necessary.

6.2. REPRESENTATION

We begin with representing institutions which regulate their own temporal rule modifications.

Definition 34. ***Institution*** *An institution is a tuple $\mathcal{I} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \Delta \rangle$. Institutions are distinguished with a superscript (e.g. $\mathcal{I}^{uk} = \langle \mathcal{E}^{uk}, \mathcal{F}^{uk}, \mathcal{C}^{uk}, \mathcal{G}^{uk}, \Delta^{uk} \rangle$). $\Sigma = 2^{\mathcal{F}}$ denotes all states for $\mathcal{I}$.*

Where:

1. $\mathcal{E} = \mathcal{E}_{obs} \cup \mathcal{E}_{inst} \cup \mathcal{E}_{mod}$ is a finite set of *events* comprising:
   - Observable events $\mathcal{E}_{obs}$ and institutional events $\mathcal{E}_{inst}$.
   - Rule modification events $\mathcal{E}_{mod} = \{mod(op, id, t) \mid op \in \{act, deact\}, id \in \mathbb{ID}, t \in \mathbb{N}\}$ – a rule with the identifier *id* (the identifier set being $\mathbb{ID}$) is activated/deactivated (*op*) at a time t.
2. $\mathcal{F} = \mathcal{F}_{dom} \cup \mathcal{F}_{ract}$ is a finite set of *fluents* describing the:
   - Domain $\mathcal{F}_{dom}$.
   - Active rules $\mathcal{F}_{ract} = \{active(id) \mid id \in \mathbb{ID}\}$ identified as *id*.
3. $\mathcal{X}$ is the set of all *contexts* φ expressible in the following grammar for fluents $f \in \mathcal{F}$:

$$\begin{aligned} \varphi ::= &\ \top \mid f \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid P \mid \\ &\ PrS(\varphi) \mid PaS(\varphi) \mid PrV(\phi) \mid PaV(\phi) \\ \phi ::= &\ \varphi \mid NS(\varphi) \end{aligned}$$

   Each expression's informal meaning is the usual for propositional logic symbols. The operators bear truth in the following cases: (a) P if the context is retroactive (i.e. the state in which P operates on is at a time before the version to which it belongs becomes the current version), and (b) if φ is true in: the previous state ($PrS(\varphi)$), all past states ($PaS(\varphi)$), the same state in the previous version ($PrV(\varphi)$), the same state in all past versions ($PaV(\varphi)$), and the next state ($NS(\varphi)$).The next state operator is restricted to past versions, meaning rules are never conditional on the actual future.
4. $\mathcal{G} : \mathcal{X} \times 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}_{inst}}$ – is the *event generation function* where $\mathcal{G}(X, E)$ is an event set caused by the events that occur (E) when the context X holds.
5. $\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}_{dom}} \times 2^{\mathcal{F}_{dom}}$ is the *state consequence function* where for a context $X \in \mathcal{X}$ and an event $e \in \mathcal{E}$ the consequence function's result is notated $\mathcal{C}(X, e) = \langle \mathcal{C}^{\uparrow}(X, e), \mathcal{C}^{\downarrow}(X, e) \rangle$ s.t. the initiated fluent set is $\mathcal{C}^{\uparrow}(X, e)$ and the terminated fluent set is $\mathcal{C}^{\downarrow}(X, e)$
6. $\Delta \subseteq \mathcal{F}$ is the *initial institution state*

For example, the following rule states that if Ada is found guilty (*g(ada)*) then she becomes a criminal (*crim(ada)*). That is, the fluent *crim(ada)* is initiated by the event of being found guilty according to the consequence function ($\mathcal{C}^{\uparrow}$).

$$\mathcal{C}^{\uparrow}(\top, g(ada)) \ni crim(ada)$$

A government rule change (*gmod(act, id, t)*)) that does not retroactively criminalise people counts–as a legal rule change. The condition is in all past retroactive states someone is not a criminal (*crim(ada)*) if in the previous version (prior to rule change) they were not.

$$\mathcal{G}(PaS(P \rightarrow PrV(\neg crim(ada)) \rightarrow \neg crim(ada))),$$
$$\{gmod(act,\ id,\ t)\}) \ni act(id,\ t)$$

In order to reason about modifying specific institutional rules, we tie rule identifiers to the institutional rules they represent. Specifically we map the inputs and single outputs of $\mathcal{G}$ and $\mathcal{C}$ to identifiers (i.e. not the whole set of events or initiated/terminated fluents).

Definition 35. ***Rule Identifier Function*** *A rule identifier function for an event generation function* $\mathcal{G} : \mathcal{X} \times 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}_{inst}}$ *is* $rid^{\mathcal{G}} : \mathcal{X} \times 2^{\mathcal{E}} \times \mathcal{E}_{inst} \rightarrow \mathbb{ID}$. *The rule identifier functions for a consequence function* $\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}_{dom}} \times 2^{\mathcal{F}_{dom}}$ *are* $rid^{\mathcal{C}^{\uparrow}} : \mathcal{X} \times \mathcal{E} \times \mathcal{F}_{dom} \rightarrow \mathbb{ID}$ *and* $rid^{\mathcal{C}^{\downarrow}} : \mathcal{X} \times \mathcal{E} \times \mathcal{F}_{dom} \rightarrow \mathbb{ID}$.

So, the previous rule criminalising Ada has the ID $crim0 = rid^{\uparrow}(\top, g(ada), crim(ada))$. Examples/case studies omit this function.

6.3. SEMANTICS

This section defines institution semantics, following InstAL's method using just sets and functions, with the following considerations.

Observable events cause an institution rule version to transition from state to state by generating transitioning events according to the event generation function $\mathcal{G}$ and initiating and terminating fluents according to the consequence function $\mathcal{C}$. An institution transitions from one version of rules to another when rule modifying events are generated by the event generation function $\mathcal{G}$.

An institutional interpretation represents this dual evolution as a tuple $M = \langle R, V\rangle$ where: 1. $V = \langle V_0, ..., V_j\rangle$ is a tuple of versions each comprising a state and event set sequence up to length k with typical element $V_v = \langle S_v, E_v\rangle$. The state sequence for v is $S_v = \langle S_{v:0}, ..., S_{v:k+1}\rangle$ with typical element $S_{v:i} \in \Sigma$ and the event set sequence (the events transitioning between states) is $E_v = \langle E_{v:0}, ..., E_{v:k}\rangle$ with typical element $E_{v:i} \subseteq \mathcal{E}$. States denoted $S_{v:i}$ and event sets $E_{v:t}$ are denoted with the version v to which they belong and their time instant i. 2. $R : [0, k] \rightarrow [0, j]$ is a function stating which institution version is the *current* version for a given time.

R also represents when rule change events occurring in a version can change that version's rules. Rule modification events only change version rules if the institution has not already evolved to a later version. For example, if on Monday a rule is added, then the institution evolves to a new *current* version where that rule is actually added on Monday. When the version evolves, previous versions become *obsolete* from then onwards (e.g. Monday) meaning their rules are not changeable. If $R(i) \leq v$ then an event occurring in version v at time i can modify rules in v since the version is not yet obsolete.

The semantics are defined with respect to the interpretation $M = \langle R, V\rangle$, an institution $\mathcal{I} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \Delta\rangle$, the set of all institutional interpretations $\mathbb{I}$, and an observable event set trace $et = \langle O_0, ..., O_k\rangle$ with typical element $O_i \subseteq \mathcal{E}_{obs}$.

6.3.1. INSTITUTIONAL CHANGE

Counts–as rules, causing institution state and version change, are conditional on a context being *modelled* by the state in an *interpretation*.

Definition 36. ***Modelling Context*** *For all $X \in \mathcal{X}$ and $f \in \mathcal{F}$, context models $\langle M, S_{v:t}\rangle \models X$ is defined for $\top$, $\vee$ and $\rightarrow$ w.r.t. $\neg$ and $\wedge$ as usual and for the other symbols as:*

$$\langle M, S_{v:t}\rangle \models f \quad \Leftrightarrow \quad f \in S_{v:t} \tag{36.1}$$

$$\langle M, S_{v:t}\rangle \models \neg\psi \quad \Leftrightarrow \quad \langle M, S_{v:t}\rangle \not\models \psi \tag{36.2}$$

$$\langle M, S_{v:t}\rangle \models \psi \wedge \phi \quad \Leftrightarrow \quad \langle M, S_{v:t}\rangle \models \psi \ \textbf{\textit{and}}\ \langle M, S_{v:t}\rangle \models \phi \tag{36.3}$$

$$\langle M, S_{v:t}\rangle \models P \quad \Leftrightarrow \quad R(t) < v \tag{36.4}$$

$$\langle M, S_{v:t}\rangle \models PrS(\psi) \quad \Leftrightarrow \quad \langle M, S_{v:t-1}\rangle \models \psi \tag{36.6}$$

$$\langle M, S_{v:t}\rangle \models PaS(\psi) \quad \Leftrightarrow \quad \forall t' \in [0, t-1] : \langle M, S_{v:t-1}\rangle \models \psi \tag{36.7}$$

$$\langle M, S_{v:t}\rangle \models PrV(\psi) \quad \Leftrightarrow \quad \langle M, S_{v-1:t}\rangle \models \psi \tag{36.8}$$

$$\langle M, S_{v:t}\rangle \models PaV(\psi) \quad \Leftrightarrow \quad \forall v' \in [0, v-1] : \langle M, S_{v'-1:t}\rangle \models \psi \tag{36.9}$$

$$\langle M, S_{v:t}\rangle \models NS(\psi) \quad \Leftrightarrow \quad \langle M, S_{v:t+1}\rangle \models \psi \tag{36.10}$$

Semantics are as usual for modelling a fluent (36.1), weak negation (36.2) and conjunction (36.3). A state is retroactive if at that time the version is not the current version but it will be in the future (36.4) – for example, if on a Wednesday the institution evolves to a new version, then anything occurring on the Monday is retroactive to the new version (i.e. occurring in the version's past). States model formula as expected for a previous state (36.6), all previous states (36.7), the previous version (36.8), all past versions (36.9) and the next state (36.10).

An event 'B' occurs when transitioning to a new state in a version according to a rule – "A counts–as B in context C" ($\mathcal{G}$) – if an event 'A' occurs, the context 'C' is modelled by the state and the counts–as rule itself is active in the version's state. Events occurring in response to observable events E are formalised as an event generation operation.

Definition 37. ***Event Generation Operation*** *The event generation operation $GR : \Sigma \times 2^{\mathcal{E}} \times \mathbb{I} \rightarrow 2^{\mathcal{E}}$ is defined such that $GR(S_{v:t}, E, M) = E'$ iff E' only satisfies the following conditions:*

$$E \subseteq E' \tag{37.1}$$

$$\exists X \in \mathcal{X}, e \subseteq E, e' \in \mathcal{G}(X, e) \cap \mathcal{E}_{inst} : id = rid^{\mathcal{G}}(X, e, e'), \\ \langle M, S_{v:t}\rangle \models X \wedge active(id) \Rightarrow e' \in E' \tag{37.2}$$

$$\exists X \in \mathcal{X}, e \subseteq E, e' \in \mathcal{G}(X, e) \cap \mathcal{E}_{mod} : id = rid^{\mathcal{G}}(X, e, e'), \\ \langle M, S_{v:t}\rangle \models X \wedge active(id), R(t) \neq v \Rightarrow e' \in E' \tag{37.3}$$

$$\exists X \in \mathcal{X}, e \subseteq E, e' \in \mathcal{G}(X, e) \cap \mathcal{E}_{mod} : id = rid^{\mathcal{G}}(X, e, e'), \\ \langle M, S_{v:t}\rangle \models X \wedge active(id), R(t) = v \Rightarrow (e' \in E' \ \textbf{\textit{or}}\ e' \not\in E') \tag{37.4}$$

Any fixed point reached after iterative applications of GR is denoted as $GR^{\omega}(S_{v:t}, E, M)$.

Events that have occurred still occur (37.1). If an active rule states an event e causes an event e' in a context modelled by the state, then e *can* cause e' to occur depending on e''s type. Specifically, whether e' is a type that could cause an inconsistency (e.g. removing rules that ascribe rule modifications, for more on the paradox of rule change see [220]). An event e' always occurs if it is a non–rule–modifying institutional event (37.2) or occurs when the version is obsolete and it cannot modify rules (37.3). Rule modifying events in non–obsolete versions *can* cause rule changes and a potential paradox. So they *optionally* occur in a non–obsolete version where they can cause rule change and/or a paradox (37.4). Hence, *GR* is *multi–valued*.

Iterating the event generation operation until a *fixed point* is reached obtains all events which occur. At least one fixed point is guaranteed.

Lemma 4. *For any set of events* $E \subseteq \mathcal{E}$*, interpretation* M *and state* $S_{v:t} \in \Sigma$ *there exists a fixed point* $GR^{\omega}(S_{v:t}, E, M)$.

Proof. *GR* always has a monotonically increasing value (w.r.t. set inclusion) and a finite domain. □

An institution version transitions between states, driven by event occurrences, according to a state transition operation.

Definition 38. ***State Transition Operation*** *The state transition operation* $TR : \Sigma \times 2^{\mathcal{E}} \times \mathbb{I} \rightarrow 2^{\mathcal{E}}$ *is defined for a state* $S_{v:t}$*, a set of events* $E_{v:t}$ *and an interpretation* M *as:*
$TR(S_{v:t}, E_{v:t}, M) =$

$$\{f \mid f \in S_{v:t} \cap TERM(S_{v:t}, E_{v:t}, M) \textbf{ or} \tag{38.1}$$
$$f \in INIT(S_{v:t}, E_{v:t}, M)\} \tag{38.2}$$

where:
$INIT(S_{v:t}, E_{v:t}, M) =$

$$\{f \mid \exists e \in E_{v:t}, X \in \mathcal{X} : id = rid^{\mathcal{C}^{\uparrow}}(X, e, f),$$
$$f \in \mathcal{C}^{\uparrow}(X, e) \cap \mathcal{F}_{dom}, \langle M, S_{v:t}\rangle \models X \wedge active(id) \quad \textbf{or} \tag{38.3}$$
$$\exists t' \in [0, k], \nexists t'' \in [t', k] : id = rid^{\mathcal{C}^{\uparrow}}(X, e, f),$$
$$R(t') \leq v, R(t'') \leq v, mod(act,\ id,\ t) \in E_{v:t'},$$
$$mod(deact,\ id,\ t) \in E_{v:t''}, f = active(id)\} \tag{38.4}$$

$TERM(S_{v:t}, E_{v:t}, M) =$

$$\{f \mid \exists e \in E_{v:t}, X \in \mathcal{X} : id = rid^{\mathcal{C}^{\downarrow}}(X, e, f),$$
$$f \in \mathcal{C}^{\downarrow}(X, e) \cap \mathcal{F}_{dom}, \langle M, S_{v:t}\rangle \models X \wedge active(id) \textbf{ or} \tag{38.5}$$
$$\exists t' \in [0, k], \nexists t'' \in [t', k] : id = rid^{\mathcal{C}^{\downarrow}}(X, e, f)$$
$$R(t') \leq v, R(t'') \leq v, mod(deact,\ id,\ t) \in E_{v:t'},$$
$$mod(act,\ id,\ t) \in E_{v:t''}, f = active(id)\} \tag{38.6}$$

Transitioning from one state to the next follows common–sense inertia – a fluent holds in a new state if it held in the previous state and was not terminated (38.1) or it was initiated in the previous state (38.2). A domain fluent is initiated/terminated if an event causes it to be according to a rule defined by the state consequence function $\mathcal{C}$ that is active in the current state with a condition (context) that is modelled in the state (38.3 for initiation and 38.5 for termination). A fluent denoting an active rule is initiated/terminated in a state if a rule activating/deactivating event occurs at a time when the version is not obsolete and no contradictory deactivation/activation event occurs at a later time when the version is not obsolete (38.4 for activating rules and 38.6 for deactivating rules). The most recent modifications in a version take precedent if they occur when the version is a non–obsolete version and simultaneous contradictory rule modifications are cancelled.

6.3.2. MODELS

Now we define when an interpretation is an institutional model for an observable event set trace. An institutional interpretation is, broadly speaking, an institutional model for an observable event set trace iff: 1. each version evolves according to the event generation and state transition operations, and 2. the institution evolves from one version to another when rules are modified. However, the event generation operation is multi–valued since rule modifications are *optional*. Thus, there are potentially multiple candidate event sets for transitioning between states and therefore multiple interpretations to select as models.

We want to maximise the rule modification events that are not self–contradicting (e.g. not applying modifications that retroactively remove a rule making retroactive rule removal possible). Interpretations are prioritised, denoted as $<$, based on maximising rule modifications. An interpretation has higher priority over another if at the earliest time in the earliest version in which the interpretation differ it contains a superset of rule modifying events compared to the 'same' set for the lower priority interpretation.

Definition 39. ***Prioritised Interpretation*** *Let* $M^0 = \langle R^0, V^0\rangle \in \mathbb{I}$ *and* $M^1 = \langle R^1, V^1\rangle \in \mathbb{I}$ *be two interpretations for institution* $\mathcal{I}$ *where:* $V^0 = \langle V^0_0, ..., V^0_i\rangle$ *with typical element* $V^0_v = \langle E^0_v, S^0_v\rangle$ *s.t.* $E^0_v = \langle E^0_{v:0}, ..., E^0_{v:k}\rangle$*, and* $V^1 = \langle V^1_0, ..., V^1_j\rangle$ *with typical element* $V^1_v = \langle E^1_v, S^1_v\rangle$ *s.t.* $E^1_v = \langle E^1_{v:0}, ..., E^1_{v:k}\rangle$*. The ordering* $<$ *is a relation between interpretations* M^0 *and* M^1 *such that:*

$$
\begin{aligned}
M^0 < M^1 \quad \Leftrightarrow \qquad & \exists t \in [0,k], \nexists t' \in [0, t\text{-}1]: \\
& v = R^0(t), E^0_{v:t} \cap \mathcal{E}_{mod} \supset E^1_{v:t} \cap \mathcal{E}_{mod} \\
& v' = R^0(t'), E^0_{v':t'} \neq E^1_{v':t'}
\end{aligned}
$$

We operationally characterise a model by constructing a 'correct' interpretation. That is, constructing versions comprising correct state transitions and generated events. We could construct each institution version by starting at an initial state and proceeding from one state to the next according to the event generation and state transition operations. However, this would require knowing which rule modification events happen in each version's past, present and future.

To give an example for an observable event set trace $\langle O_0, ..., O_k\rangle$. An institution starts at an initial state only comprising an active rule enabling a government to make retroactive modifications ($\Delta = S_{0:0} = \{active(gov0)\}$). First, a fence is observably built ($O_0 = \{fb\}$, occurring during the first state transition $fb \in E_{0:0} = GR^{\omega}(S_{0:0}, O_0, M)$). But, there is no active rule that causes the next state to be different ($S_{0:1} = TR(S_{0:0}, E_{0:0}) = S_{0:0} = \{active(gov0)\}$). Then, the government votes to retroactively activate a rule in state zero, stating building a fence initiates an obligation to paint it. Consequently, the second state which has already been determined, $S_{0:1}$, seems wrong since it lacks the fence painting rule and its effects. In fact, the institution should transition to a new rule version V_1. This new version should start at the same initial state $S_{1:0} = \Delta$. But, crucially, transition to the next state ($S_{1:0} = TR(S_{1:0}, E_{1:0})$) with the knowledge that in the future of the new version the fence painting rule will be retroactively added at state zero ($S_{1:0}$) and become active in the second state ($S_{1:1}$). State transitions are defined with respect to an interpretation comprising past/present/future rule modification events which might be unknown when each state and transitioning event set is constructed.

We define an interpretation successor operation which addresses the problem of constructing a 'correct' interpretation without the knowledge of each version's past/present/-future. The successor operation takes as input a preceding interpretation which supplies versions comprising a past/present/future on which each version in the new succeeding interpretation can be constructed according to *TR* and *GR*. That is, a new interpretation is produced using the version timelines of the previous interpretation, taking into account past/present/future rule modifications from the preceding interpretation's version timelines.

A succeeding interpretation might not be the same as the previous interpretation, since the previous interpretation might have been built without knowledge of its own past/present/future. That is, the new interpretation might differ in its temporal evolution (comparable version timelines in each interpretation being different). Consequently, the succeeding interpretation might have new, previously unknown, rule modification events that also need to be accounted for and thus another succeeding interpretation must be produced.

The idea is to iteratively apply the institution successor operation until a succeeding interpretation is produced that is the same as the previous interpretation. That is, until the operation reaches a fixed point, which is guaranteed according to lemma 6 we give later on. Intuitively, the fixed point characterises an interpretation that is built taking into account its own past/present/future modifications in each version (since it was built with respect to an identical preceding interpretation). Formally, the successor interpretation operation is:

Definition 40. ***Successor Interpretation Operation*** *Let* $et = \langle O_0, ..., O_k \rangle$ *be an observable event trace for* $\mathcal{I}$ *of length k. Let* $M' = \langle R', V' \rangle \in \mathbb{I}$ *be an interpretation such that* $V' = \langle V'_0, ..., V'_{j'} \rangle$ *is a tuple of institution versions. The interpretation successor operation* $SUCC : \mathbb{I} \times ET \rightarrow \mathbb{I}$ *is defined for the interpretation M w.r.t.* $\mathcal{I}$ *and et such that* $SUCC(M, et) = M'$ *iff* M' *satisfies the following conditions:*

$$\forall v \in [0, j'] : S'_{v:0} = \Delta \tag{40.1}$$

$$\forall v \in [0, j'], t \in [0, k] : E'_{v:t} = GR^{\omega}(S'_{v:t}, O_t, M) \tag{40.2}$$

$$\forall v \in [0, j'], t \in [0, k] : S'_{v:t+1} = TR(S'_{v:t}, E'_{v:t}, M) \tag{40.3}$$

$$R'(t) = \begin{cases} 0, & t = 0, E'_{0:t} \cap \mathcal{E}_{mod} = \emptyset \\ 1, & t = 0, E'_{0:t} \cap \mathcal{E}_{mod} \neq \emptyset \\ R'(t\text{-}1), & t > 0, E'_{R(t\text{-}1):t} \cap \mathcal{E}'_{mod} = \emptyset \\ R'(t\text{-}1)+1, & t > 0, E'_{R(t\text{-}1):t} \cap \mathcal{E}'_{mod} \neq \emptyset \end{cases} \tag{40.4}$$

$$\textit{Given that } V' = \langle V'_0, ..., V'_{j'} \rangle, R'(k) = j' \tag{40.5}$$

Every institution version starts at the same initial state (40.1). Each state transition (an event set) in a version is produced by the event generation operation applied to the previous state and the observable events occurring at that time (40.2). The next state in a version is the state produced by the state transition operation applied to the previous state and the transitioning events occurring in that version *with respect to* the preceding institutional interpretation (40.3). That is, transitioning from one state to the next takes into account the rule modification events occurring in the past/present/future of the same version in the preceding interpretation. Rule modifications in the latest version cause the current version to evolve/increment to the next version. If no rule modification takes place the version remains the same or the zeroeth version for the zeroeth time instant (40.4). If a rule modification does take place in the latest version, then the current version at that time incremented by one, or is the first version for the zeroeth time point (40.4). The version sequence only goes up until the current version at the last time instant (40.5).

At least one fixed point for the successor interpretation operation, starting at any initial interpretation, is always guaranteed. A fixed point is denoted as $SUCC^{\omega}(M, et)$. To see why, the general idea is that there always exists a series of successive interpretations that monotonically increase which versions and states they agree on.

The following lemma is used to prove that there always exists a series of such interpretations and therefore that there always exists a fixed point. Informally, the lemma is conditional on there being two successors M' and M'' to any interpretation that agree with each other up until a particular time (h) in a version (j). The consequence is that the second interpretation M'' has the same events at time h and state transition at time $h+1$ in version j as if the event and state transitions were produced with respect to M'''s *own* past/present/future timeline.

Lemma 5. *If $\mathcal{I}$ is an institution, M an interpretation and et an observable event trace of length k for $\mathcal{I}$ and there exists interpretations $M' = SUCC(M, et)$ and $M'' = SUCC(M', et)$ where* $\exists h \in [0, k], j \in [0, v'], \forall i \in [0, k]$:

$$\langle V'_0, ..., V'_{j-1} \rangle = \langle V''_0, ..., V''_{j-1} \rangle \tag{A5.1}$$

$$\langle S'_{j:0}, ..., S'_{j:h} \rangle = \langle S''_{j:0}, ..., S''_{j:h} \rangle \tag{A5.2}$$

$$\begin{matrix} mod(op,\ id,\ h) \in E'_{v:i}, \\ R'(i) \leq j \end{matrix} \Leftrightarrow \begin{matrix} mod(op,\ id,\ h) \in E''_{j:i}, \\ R''(i) \leq j \end{matrix} \tag{A5.3}$$

then $E''_{j:h} = GR^{\omega}(S''_{j:h}, O_h, M'')$ *and* $S''_{j:h+1} = TR(S''_{j:h}, E''_{j:h}, M'')$

The previous lemma's assumptions can always be met starting from *any* interpretation M. Firstly, since in the worst case, from any interpretation we can obtain a successor starting at the institution's initial state – so both successors agree at least on the initial state.

Secondly, by making the non–deterministic choice in the event generation operation to select the same rule modifications for both the successor and the successor to the successor (in the worst case, no rule modifications). We can continue to incrementally produce successive interpretations that monotonically increase the time point they agreed upon. Note that, this may mean backtracking by changing preceding interpretations (e.g. selecting no rule modifications).

Lemma 6. *There exists a fixed point for the interpretation successor operation denoted* $SUCC^{\omega}(M, et)$ *for any* M *and* et.

In fact, there can be multiple fixed points, as exemplified:

Example 1. *An institution* $\mathcal{I}$ *contains a legislative rule with the id* $leg0 \in \mathbb{ID}$ *stating that an agent, Ada, voting to activate a rule* ($vote_a(act, id, t) \in \mathcal{E}_{obs}$) *counts–as activating the rule:* $\mathcal{G}(\top, \{vote_a(act, id, t)\}) \ni mod(act, id, t)$*. In the initial state the legislative rule is active* $\Delta = \{active(leg0)\}$*. In an observable event trace* $et = \langle O_0 \rangle$ *Ada votes to activate another rule with the id* $leg1 \in \mathbb{ID}$ *in the initial state* $O_0 = \{vote_a(act, leg1, 0)\}$.

From an initial empty interpretation M we have the following successors and interpretations for example 1(differences are in **bold**):

$$M^2 = SUCC(M, et) = SUCC^{\omega}(M, et) \text{ s.t. } V^2 = \langle V_0^2 \rangle, R^2(0) = 0, R^2(1) = 0,$$

$$S_{0:0}^2 = \{active(leg0)\}, S_{0:1}^2 = \{active(leg0)\}, E_{0:0}^2 = \{vote_a(act, leg1, 0)\}$$

$$M^1 = SUCC(M, et) = SUCC^{\omega}(M, et) \text{ s.t. } V^1 = \langle V_0^1, V_1^1 \rangle, R^1(0) = 1, R^1(1) = 1,$$

$$S_{0:0}^1 = \{active(leg0)\}, S_{0:1}^1 = \{active(leg0)\},$$

$$E_{0:0}^1 = \{vote_a(act, leg1, 0), \mathbf{mod(act, leg1, 0)}\}$$

$$S_{1:0}^1 = \{active(leg0)\}, S_{1:1}^1 = \{active(leg0)\}, E_{1:0}^1 = \{vote_a(act, leg1, 0)\}$$

$$M^0 = SUCC(M, et) = SUCC^{\omega}(M, et) \text{ s.t. } V^0 = \langle V_0^0, V_1^0 \rangle, R^0(0) = 1, R^0(1) = 1,$$

$$S_{0:0}^0 = \{active(leg0)\}, S_{0:1}^0 = \{active(leg0)\},$$

$$E_{0:0}^0 = \{vote_a(act, leg1, 0), \mathbf{mod(act, leg1, 0)}\}$$

$$S_{1:0}^0 = \{active(leg0)\}, S_{1:1}^0 = \{active(leg0), \mathbf{active(leg1)}\},$$

$$E_{1:0}^0 = \{vote_a(act, leg1, 0), \mathbf{mod(act, leg1, 0)}\}$$

Each fixed point has different rule modifications. M^2 does not add the rule *leg1*. M^1 contains an attempt to add the rule in the version zero but not in version one. Finally, M^0 adds the rule in the version zero and version one, version one being the current version when the rule is added meaning the rule addition is successful. In fact, the following prioritisation holds $M^0 < M^1 < M^2$ meaning that M^0 maximises successful rule modifications.

Models are interpretations which maximise successful rule modifications. Thus we characterise models by combining the successor interpretation fixed point and interpretation prioritisation. Given an empty interpretation we find a fixed point successor interpretation for a given event set trace (41.1). The fixed point is a model if there is no greater prioritised successor fixed point interpretation (41.2).

Definition 41. ***Models*** *Let* $M = \langle R, V \rangle$ *be an empty interpretation such that* $V = \langle V_0 \rangle$*,* $V_0 = \langle E_0, S_0 \rangle$*,* $E_0 = \langle\rangle$ *and* $S_0 = \langle\rangle$*. The interpretation* $M' = \langle R', V' \rangle$ *is a model for* $\mathcal{I}$ *w.r.t. an* *observable event set trace* $et = \langle O_0, ..., O_k \rangle$ *iff:*

$$M' = SUCC^{\omega}(M, et) \;\; \textbf{and} \tag{41.1}$$

$$\textit{There does not exist an } M'' < M' \textit{ meeting 41.1.} \tag{41.2}$$

From lemma 6 and definition 41 we have the following property.

Lemma 7. *There exists at least one model for any institution* $\mathcal{I}$ *w.r.t. an observable event set trace et.*

These semantics operationalise answering "when does a rule change count–as a legal rule change?". Generally, a physical or institutional actions counts–as the social action of a legal rule change *if and only if* the rule change is ascribed by secondary legal counts–as rules conditional on a social context, which can include hypothetical rule change effects, that holds before and after the rule change takes place. Models always contain 'legal' rule modifications, defined as fixed point interpretations which maximise rule modifications. So, 'legal' rule–changes occur in at least one model whilst illegal rule changes do not occur at all (the non–deterministic choice for a rule modification to occur in 37.4) and the institution continues to operate 'as usual', meeting our desiderata.

6.4. CASE STUDIES

Now we apply the framework to concrete case studies. For brevity we use variables to denote: all rule identifiers ($id \in \mathbb{ID}$), all rule change operations ($op \in \{act, deact\}$), and all time instants ($t \in \mathbb{N}$). The first case concerns a simple rule change procedure.

Case 6.4.1. *An institution* $\mathcal{I}^{sgov}$ *describes a* **s***imple* **gov***ernment comprising two agents, Ada and Bertrand. Both Ada and Bertrand voting to activate or deactivate a rule in the context that neither are criminals* ($crim(ada), crim(bert) \in \mathcal{F}^{sgov}_{dom}$) *counts–as activating/deactivating the rule. The rule modifying counts–as rules are identified with leg0* $\in \mathbb{ID}$ *and formalised as* $\mathcal{G}^{sgov}(\neg crim(ada) \wedge \neg crim(bert), \{vote_a(op,\ id,\ t), vote_b(op,\ id,\ t)\}) \ni mod(act,\ id,\ t)$*. At time point one Ada and Bertrand vote to add a rule with id crim0,* $O_1 = \{vote_a(act,\ crim0,\ 1), vote_b(act,\ crim0,\ 1)\}$*. The rule identified as crim0 states that if Ada or Bertrand are found guilty of a crime* ($g(ada), g(bert) \in \mathcal{E}^{sgov}_{obs}$) *then they become criminals, formally –* $\mathcal{C}^{\uparrow}(\top, g(ada)) \ni crim(ada)$ *and* $\mathcal{C}^{sgov\uparrow}(\top, g(bert)) \ni crim(bert)$*. Next, Bertrand is found guilty of a crime* $O_2 = \{g(bert)\}$*. Finally, Bertrand and Ada vote to deactivate the criminalising rule,* $O_3 = \{vote_a(act,\ crim0,\ 3), vote_b(act,\ crim0,\ 3)\}$*.*

For clarity, models are represented graphically. The model for case 6.4.1 is shown in Figure 6.1. Lines represent when domain and active rule fluents hold. We distinguish between whether a fluent holds in a state $S_{v:t}$: 1. retroactively in the version's past and *not* in the previous version, - - - - (i.e. $R(t) < v$ and $\langle M, S_{v-1:t} \rangle \not\models f$), 2. when the version is the *current version*, ——) (i.e. $R(t) = v$), and 3. when the version is *obsolete*, ······ (i.e. $R(t) > v$). Time instants are marked if they have successful or non–successful rule modification events in versions where modifications can have an effect (i.e. non–obsolete versions): 1. ▶ denoting that all the rule modification events occurring in the previous version occur again (i.e. $E_{v:t} \cap \mathcal{E}_{mod} = E_{v-1:t} \cap \mathcal{E}_{mod}$). Meaning, the conditions (contexts) for the rule modifying events to be ascribed are consistent with the version and therefore with applying the rule modifications (the non–deterministic choice to include a rule modification in $E_{v:t}$ according to 37.4 is always made) 2. ◀ denoting that at least one rule modification event which occurred in the previous version does not occur again (i.e. $E_{v:t} \cap \mathcal{E}_{mod} \neq E_{v-1:t} \cap \mathcal{E}_{mod}$). Meaning, the conditions (contexts) for rule modifying events to be ascribed are inconsistent with the version they occur in and therefore with applying the rule modifications (a non–deterministic choice according to 37.4 to *not* include a rule modification is made when building $E_{v:t}$).

Figure 6.1 shows case 6.4.1's model. Throughout version zero the legislative rule (*leg0*) is active, stating Ada and Bertrand voting to add a rule counts–as adding a rule. When at time instant one Ada and Bertrand vote to add a new rule (*crim0*), stating people found guilty become criminals, the model succeeds to version one where the new rule is successfully added. At time instant three Bertrand becomes a criminal. When they vote again to modify a rule it is unsuccessful, since rule change is conditional on neither being criminals. Adding a criminalising rule altered the built social reality in version one's future, changing what could be ascribed as a legal rule modification.

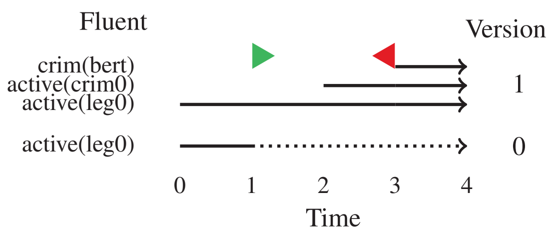

Figure 6.1: Model for case 6.4.1 with two institution versions.

The next case presents an institution $\mathcal{I}^{uk}$ representing the UK's legislation rules. The cases are based on past changes to a court decision on UK tax laws ([1]), and past changes to tax laws ([192]). The UK government can unconditionally enact any rules effective at any time. Observable events where the government activates/deactivates a rule (*gmod(op, id, t)*) count–as modifying the rule (*mod(op, id, t)*. Legislative rules identified as *leg0* $\in \mathbb{ID}$ cause rule activations $\mathcal{G}^{uk}(\top, \{gmod(act,\ id,\ t)\}) \ni mod(act,\ id,\ t)$ and legislative rules identified as *leg1* $\in \mathbb{ID}$ cause rule deactivations $\mathcal{G}^{uk}(\top, \{gmod(deact,\ id,\ t)\}) \ni mod(deact,\ id,\ t)$. A model $M^{uk} = \langle R^{uk}, V^{uk} \rangle$ is produced for an observable event trace $et = \langle O_0, O_1, O_2, O_3, O_4 \rangle$ for $\mathcal{I}^{uk}$. The model comprises four versions $V^{uk} = \langle V_0^{uk}, V_1^{uk}, V_2^{uk}, V_3^{uk} \rangle$. We begin the case:

Case 6.4.2. *A rule states that any UK resident (e.g. person a resides in the UK –r(a, uk)) in a business partnership in the UK (p(a, uk)) or elsewhere such as Jersey (p(a, jers)) in the first tax year month is obliged to pay tax (oblt). We have for all locations* $L \in \{uk, jers\}$ *a tax rule* $\mathcal{C}^{uk\uparrow}(r(a,\ uk) \wedge p(a,\ L), mon1) \ni oblt$ *identified as tax0* $\in \mathbb{ID}$. *Initially the legislative rules leg0 and leg1, and the tax rule tax0 are active* ($\Delta^{uk} = \{active(leg0), active(leg1), active(tax0)\}$). *At time point one it is the first tax year month* ($O_1 = \{mon1\}$). *Following a court challenge ([1]) the government retroactively replaces the tax rule with id tax0 with a new rule with id tax1* ($O_2 = \{gmod(deact,\ tax0,\ 0), gmod(act,\ tax1,\ 0)\}$). *The new rule, tax1, states that only people in a UK business partnership are obliged to pay tax –* $\mathcal{C}^{uk\uparrow}(r(a,\ uk) \wedge$

p(a, uk), mon1) $\ni$ *oblt.*

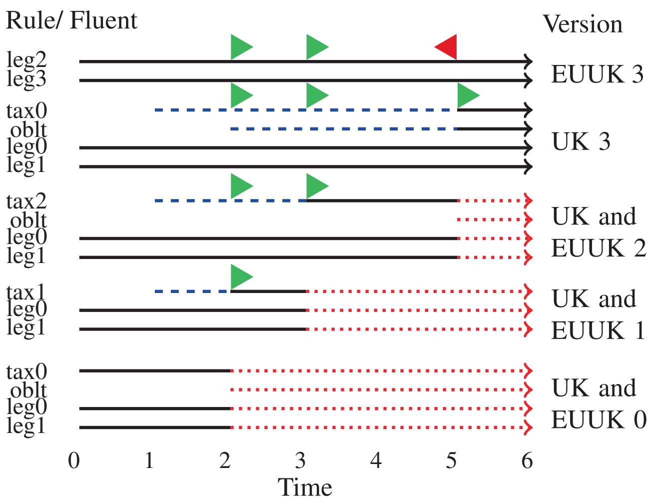

Figure 6.2: Model for case 6.4.2 with four institution versions for the institution $\mathcal{I}^{uk}$ (denoted UK) and a model for case 6.4.3 with four versions for the institution $\mathcal{I}^{euuk}$ (denote EUUK). Institutions $\mathcal{I}^{uk}$ and $\mathcal{I}^{euuk}$ have identical versions 0 to 2. Not shown, in all states person 'a' is in a Jersey–based business partnership (*p(a,jersey)*) and is a UK resident (*r(a,uk)*).

In Figure 6.2 version zero (V_0^{uk}) obliges person 'a' to pay tax in state two. In state two ($S_{0:2}^{uk}$) the current institution version changes when the rule obliging UK residents to pay tax (*tax0*) is replaced with the rule obliging UK business partners to pay tax (*tax0*) (*act(t1,0), deact(t0,0)* $\in E_{0:2}^{uk}$) to version one (V_1^{uk} s.t. $R^{uk}(2) = 1$). Due to this change, the version one (V_1^{uk}), does not oblige tax to be paid in its third state ($S_{1:2}^{uk}$) since person *a* resides in the UK but is in a Jersey–based business partnership.

Case 6.4.2 (continued)**.** *The government partially reverses the tax change at time point three. This is by retroactively replacing the rule obliging people in a UK business partnership to pay tax (tax1) with a new rule identified as tax2*
(O_3 = {gmod(deact, tax1, 0), gmod(act, tax2, 0)}). The new rule obliges UK residents in a business partnership to pay tax if it does not criminalise them retroactively (i.e. in a retroactive state an obligation to pay tax is initiated conditional on the obligation holding in the next state of the previous version). For all locations L $\in$ *{uk, jersey} the rule is* $\mathcal{C}^{uk\uparrow}$*((r(a, uk)* $\wedge$ *p(a, L))* $\rightarrow$ *(P)* $\rightarrow$ *PrV(NS(oblt))), mon1)* $\ni$ *oblt. Next, it is the first tax year month again (O_4 = {mon1}).*

In Figure 6.2 version two (V_2^{uk}), like version zero, does not oblige 'a' to pay tax in the past. But, it does oblige them to pay tax after the second time the first tax year month occurs (*mon1* $\in E_{2:4}^{uk}$).

Case 6.4.2 (continued)**.** *The UK government decides to reverse the previous judgements going back to the original rule set (O_5 = {gmod(deact, tax2, 0), gmod(act, tax0, 0)}).*

In Figure 6.2, version three (V_3^{uk}) reverts to the original legislation. Thus we have the same situation as if the legislation in version zero had not been modified. That is, an obligation to pay tax after the first occurrence of the first tax year month ($mon1 \in E_{3:1}^{uk}$).

The next case is a variation on the previous describing an institution $\mathcal{I}^{euuk}$, incorporating EU human rights law.

Case 6.4.3. *The European Convention on Human Rights [48, Art. 7] (ECHR) blocks retroactive legislative modifications that criminalise formerly innocent people. The institution $\mathcal{I}^{euuk}$ contains the same rules as $\mathcal{I}^{uk}$ with the same identifiers minus the legislative rules leg0 and leg1. Instead, legislative rules state that observable rule modifications count–as rule modifications conditional on the changes not retroactively criminalising people. In all states where rules are being applied retroactively, if there is not an obligation to pay tax in the previous version then there must not be an obligation to pay tax in the current version. We have rules with the identifier* *l2:* $\mathcal{G}^{euuk}(PaS(P \rightarrow PrV(\neg oblt) \rightarrow \neg oblt)), \{gmod(act, id, t)\}) \ni act(id, t)$, *and rules with the identifier l3:* $\mathcal{G}^{euuk}(PaS(P \rightarrow PrV(\neg oblt) \rightarrow \neg oblt)), \{gmod(deact, id, t)\}) \ni deact(id, t)$. *Initially, person 'a' is in a Jersey based business partnership (p(a,jersey)) and is a UK resident (r(a,uk)), and the first tax rule and the legislative rules conditional on being non retroactively criminalising are active such that* $\Delta^{euuk} = \{p(a,uk), r(a,uk), tax0, l2, l3\}$. *The same events occur as in case 6.4.2,* $et = \langle \emptyset, \{mon1\}, \{gmod(deact, t0, 0), gmod(act, t1, 0)\}, \{gmod(deact, t1, 0), gmod(act, t2, 0)\}, \{mon1\}, \{gmod(deact, t2, 0), gmod(act, t0, 0)\}\rangle$.

Figure 6.2 shows a model M^{euuk} for $\mathcal{I}^{euuk}$. The first three versions are identical to our previous case 6.4.2 (where the UK's legislature was not constrained by EU rules blocking retroactively criminalising modifications), since the first two rule modifications do not criminalise people retroactively. Unlike in our previous case 6.4.2, the version two contains no tax rules. The reason being that tax rule two – "obliging uk residents in a business partnership to pay tax but on the condition that if it is retroactive then those people were obliged to pay tax in the previous version", is deactivated since its deactivation does not criminalise retroactively. On the other hand, tax rule zero – "any UK resident in a business partnership in the first tax year month is obliged to pay tax" ($\mathcal{C}^{uk\uparrow}(r(a, uk) \wedge p(a, L), mon1) \ni oblt$) is not reactivated, even though it was reactivated in our previous case 6.4.2. Its reactivation would retroactively criminalise people if activated in version three, meaning its activation does not occur since legislative rule – *l2*: $\mathcal{G}^{euuk}(PaS(P \rightarrow PrV(\neg oblt) \rightarrow \neg oblt)), \{gmod(act, id, t)\}) \ni act(id, t)$ – has a condition that is not met.

The next cases look at modifying legislative rules themselves.

Case 6.4.4. *An institution $\mathcal{I}^p$ describes a parliament that can retroactively modify rules through a majority vote $pvote(act, id, t) \in \mathcal{E}_{obs}$. The legislative rules are identified the id $parl0 \in \mathbb{ID}$ for activating rules $\mathcal{G}^p(\top, \{pvote(act, id, t)\}) \ni mod(act, id, t)$ and with the id $parl1 \in \mathbb{ID}$ for deactivating rules $\mathcal{G}^p(\top, \{pvote(deact, id, t)\}) \ni mod(deact, id, t)$. In the initial state all rules are active such that $active(id) \in \Delta$. In an observable event set trace $tr = \langle O_0, O_1\rangle$ at time point one the parliament votes to retroactively remove the rule which ascribes retroactive modifications ($O_1 = \{pvote(deact, parl1, 0)\}$.*

Depicted in Figure 6.3 a single model $M^p = \langle R^p, V^p\rangle$ comprises two institution versions $V^p = \langle V_0^p, V_1^p\rangle$. An event occurs in version zero at time instant one, where the parliament votes to retroactively modify a rule and the corresponding rule modification event occurs ($E^p_{0:1} = \{$*pvote(deact, parl1, 0)*,*mod(deact, parl1, 0)*$\}$). Consequently the institution transitions to version one ($R^p(1) = 1$). Importantly, in version one, the same rule modifying event *does not occur*. The reason being, if the modification event did occur then the rule *parl1* ascribing the modification event – $\mathcal{G}^p(\top, \{$*pvote(deact, id, t)*$\}) \ni$ *mod(deact, id, t)* – would be inactive in version one state one $S^p_{1:1}$, and the deactivation could not occur in the first place (contradiction). This exemplifies how the formalism always guarantees a model, paradoxical rule modifications do not occur if they make the rule modifying event impossible in the first place.

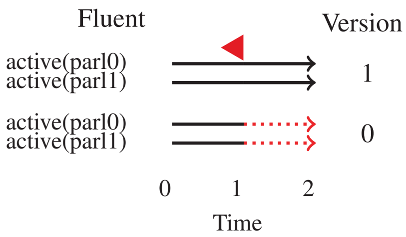

Figure 6.3: Model for case 6.4.4

The next case extends the previous case 6.4.4:

Case 6.4.5. *This case describes an institution $\mathcal{I}^{mp}$ where a monarch and a parliament can retroactively modify rules, including all the rules from the previous case's institution $\mathcal{I}^p$. Additionally, a rule identified as fence0 $\in \mathbb{ID}$ states that if a fence is built fb $\in \mathcal{E}^{mp}_{obs}$ it is obliged the fence is painted white oblpf $\in \mathcal{F}^{mp}_{inst}$ – $\mathcal{C}^{mp\uparrow}(\top, fb) \ni oblpf$. A rule identified as mon0 states the monarch issuing a rule change decree mdecree(act, id, t) $\in \mathcal{E}^{mp}_{obs}$ to activate a rule counts–as activating the rule – $\mathcal{G}^{mp}(\top, \{$mdecree(act, id, t)$\}) \ni$ mod(act, id, t). A rule identified as mon1 state the monarch issuing a decree to deactivate a rule counts–as deactivating the rule $\mathcal{G}^{mp}(\top, \{$mdecree(deact, id, t)$\}) \ni$ mod(deact, id, t). All legislative rules are initially active, but the fence painting rule is not (s.t. active(fence0) $\not\in \Delta^{mp}$). At time point one the parliament votes for the fence–painting obligation rule to be activated, ($O_1 = \{$pvote(act, fence0, 1)$\}$), a fence is built ($O_2 = \{fb\}$), the monarch issues by decree the fence–building rule to be retroactively deactivated at the time it was activated, cancelling its activation ($O_3 = \{$mdecree(deact, fence0, 1)$\}$). Finally, the parliament votes to retroactively dis–enable the monarch from deactivating rules ($O_4 = \{$pvote(deact, mon1, 0)$\}$).*

Depicted in Figure 6.4 the model $M^{mp} = \langle R^{mp}, V^{mp}\rangle$ comprises four versions $V^{mp} = \langle V^{mp}_0, V^{mp}_1, V^{mp}_2, V^{mp}_3\rangle$. At version zero time instant zero the parliament votes to add the rule obliging built fences to be painted white, causing a rule modification event ($E^{mp}_{0:1} = \{$*pvote(act, fence0, 1)*,*mod(act, fence0, 1)*$\}$) and the institution to transition to the version one ($R^{mp}(1) = 1$) where the same modification occurs
($E^{mp}_{1:1} = \{$*pvote(act, fence0, 1)*,*mod(act, fence0, 1)*$\}$). In the version one time instant two building a fence ($fb \in E^{mp}_{1:2}$) causes an obligation to paint the fence $oblpf \in S^{mp}_{1:3}$. At time instant three the monarch retroactively deactivates the fence painting rule
(*mdecree(deact, fence0, 1)* $\in E^{mp}_{1:3}$) causing the institution to transition to the version two

($R^{mp}(3) = 2$) where the modification takes effect (*mdecree(deact, fence0, 1)* $\in E^{mp}$). Consequently, the fence painting obligation rule is deactivated and its effects (an obligation) no longer hold. When the parliament retroactively removes the ability for the monarch to deactivate rules the institution transitions to the final version three ($R^{mp}(4) = 3$) where the parliament's retroactive rule removal takes effect
(*pvote(deact, mon1, 0),mod(deact, mon1, 0)* $\in E^{mp}_{3:4}$) causing the monarch's modifications to be unravelled (note that at the final version's third time instant the monarch's rule modification is unsuccessful even though it was successful in the previous version). Consequently, the fence painting obligation rule and its effects (an obligation) is reinstated by retroactively removing the ability to deactivate the fence painting rule.

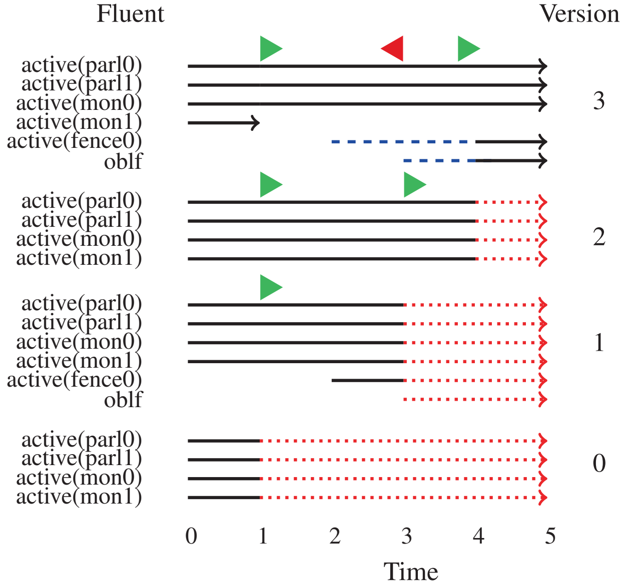

Figure 6.4: Model for case 6.4.5 with four institution versions.

6.4.1. RELATED WORK

In this chapter we contributed a framework for reasoning about the legality of rule changes. Key to our contribution is the idea of rule–modifying constitutive rules, which express things like "A counts–as modifying a rule in the past/present/future in context C". The main idea is to contribute a formal framework for reasoning about the legality of rule change ascribed by constitutive rules. In particular, accounting for the fact that changing rules in the past/present/future affects which rule changes are possible in the first place. Moreover, reasoning about conditions on rule change that are dependent on the rule change effects, such as not criminalising people in the past that were previously innocent (non–retroactively criminalising). Broadly speaking, there are three competing formalisations which look at the legality of rule change. We compare these as follows.

In ([100, 102]) a defeasible logic is proposed for temporal rule modification operations.

Operations include, in ([102]), complete rule removal (annulment) and removing immediate rule effects (abrogation). Meta–rules are used to introduce rule changes, which bear similarity to our rule–modifying counts–as rules. However, the meta–rules are only conditional on the social context at a single point in time. For comparison, we formalise richer conditions in rule–modifying constitutive rules. Specifically, rule change being conditional on the social reality in previous institution versions with previous rule–sets, previous institutional states and hypothetical rule changes required to capture a number of important examples we address (such as rule change being non–retroactively criminalising). The focus of these papers is on rule change operations found in the legal domain, rather than the relation between ascribing a social reality using constitutive rules and ascribing rule modifications with constitutive rules conditional on the social reality.

In ([159, 160]) electronic institutions are specified in the Z specification language where legislation norms restrict legislative actions. The conditions for legislation norms are less expressive than our proposal and the authors do not consider the interdependency between changing rules in the past/present/future and the built social reality.

On the other hand, in [28] rule modifications ascribed by counts–as rules are formalised where there is such a potential interdependency. The focus in [28] is on playing games in a setting where agents can act on the environment and also the rules which govern their actions. Most significantly, the main difference is that in [28] a static setting is formalised where institutions do not evolve over time from state to state or one set of rules to another. In our framework we explicitly look at these aspects not examined in [28].

Slightly further afield, the MOISE+ organisational framework has been used to construct organisations that act to enact organisational changes, in this case applied to a football team case study [123]. The focus is on the structural organisational elements required to re–organise an organisation. For example, the organisational roles required to assess when a re–organisation is required and what direction the organisational design should take. In contrast to this chapter, the focus is not on constraints making the social actions of norm change possible (or in their case, organisational change). Nor is the meta–level, the change of the rules making rule change possible. Instead, the claim is that they propose an organisational architecture required to make organisational change effective, and their claim is supported with an empirical simulation using a Markov Decision Process to select appropriate organisational changes.

More generally, other work looks at the problem of normative system change. This body of research includes: 1. norm change postulates ([28]), 2. detecting and/or resolving norm inconsistencies ([46, 130, 133, 149, 154, 232]) and 3. temporal norm updates ([7, 148]). However, these frameworks do not look at rule change legality ascribed by constitutive rules over time.

6.5. DISCUSSION

This chapter answers the question "how should we define when rule changes count–as legal and valid rule changes?" with a novel formal framework. Our framework supports reasoning about institutional rule change over time, where rule changes are ascriptively regulated by counts–as rules. We presented a novel semantics defining how an institution evolves from one social reality to the next and from one version of rules to another. Under the proposed semantics counts–as rules define the past/present/future social reality. In turn, rule modifications change counts–as rules in the past/present/future and therefore the constructed social reality. Rule modifications are in turn conditional on the built social reality and their potential effects. To summarise the general answer to the question "how should we define when rule changes count–as legal and valid rule changes?", according to our formalisation a legal rule change occurs 1. if the rule change is ascribed by counts–as rules, conditional on a context taking into account the potential changes to the context the rule modification *would* make 2. taking into account past/present/future rule modifications effect on counts–as rules and thus the context in which rule changes are conditional on.

In chapters 3, 4 and 5 our Searlian notion of an institution is a social language that ascribes institutional facts, such as obligations and various social notions such as murder. If an institutional fact is ascribed by a brute fact that is an agentive observable event (i.e. an agent *doing* something) then we say the agent that performed the action had the *legal power* in the Hohfeldian sense to cause the institutional fact to occur or hold. That is, in previous chapters 3, 4 and 5 we were dealing with the legal power for agents to bring about institutional facts, including obligations when we are dealing with norms. In comparison, in this chapter it is the norms and other counts-as rules themselves that are ascribed from brute facts, or brought about by agents with sufficient legal power. Thus, unlike in previous chapters, this chapter goes beyond merely the notion of legal power to ascribe institutional facts in a limited sense. This chapter, when it talks about legality, is really talking about the legal power to ascribe norms and other counts-as rules themselves, which in turn give agents legal power to bring about further institutional facts and rules.

In terms of future work, there are two important formal and application–orientated avenues. From a formal perspective, we considered rule changes conditional on their potential effects in the past/present/future, but not whether affecting the past/present/future in general. Yet, some administrative laws explicitly block retroactive modifications (e.g. [221, Art. 1 Sec. 9 Cl. 3] "No Bill of Attainder or ex post facto Law shall be passed"). In our framework a weak notion of preventing retroactive modifications can be expressed as a lack of retroactive rule–modifying counts–as rules. A stronger notion cannot be expressed *blocking* rules that allow modifications in the past from being introduced. A natural representation is higher–priority rule modifying counts–as rules (e.g. from a higher authority), but requires a defeasible logic left for further investigation.

In principle, the framework's fixed–point institution model characterisation can be implemented in any adequately expressive language. Future work should investigate if there is a suitable representation in declarative languages, well suited to rule–based reasoning. One candidate is Answer–Set Programming (ASP) [91, 93] as used in the InstAL framework [43]. The main obstacles are finding an adequate representation of the institutions presented in this chapter (i.e. rules that can be modified) and optimising for answer–sets that contain a superset of rule modifications over other answer–sets (thus corresponding to our notion of a model).

7

Application

If you find that you're spending almost all your time on theory, start turning some attention to practical things; it will improve your theories. If you find that you're spending almost all your time on practice, start turning some attention to theoretical things; it will improve your practice.

Donald Knuth

This chapter is loosely based on the following papers:

King, T. C., Liu, Q., Polevoy, G., Weerdt, M. de, Dignum, V., Riemsdijk, M. B. van, & Warnier, M. (2014). Request Driven Social Sensing (Demonstration). In A. Lomuscio, P. Scerri, A. Bazzan, & M. Huhns (Eds.), Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014) (pp. 1651 – 1652). Paris, France: International Foundation for Autonomous Agents and Multiagent Systems. [146]

King, T. C., Riemsdijk, M. B. Van, Dignum, V., & Jonker, C. M. (2015). 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 (pp. 114 – 131). Springer. [147]

In this dissertation each formal contribution has been developed and assessed against case studies. In this chapter, we describe an application and implementation in the context of the SHINE project under which this research was conducted. The SHINE project aims to provide suitable mechanisms that help obtaining detailed environmental data through user participation. The idea is to get users to participate in providing data by contributing the sensors they own together with any actions the user needs to take such as taking a photograph or ensuring a microphone is turned on. In return for donating their time, taking useful actions and contributing their resources, users are able to receive the valuable information provided by the participation of other users. In this scenario, institutions in the form of contracts require users to contribute their cellphone sensors, via cellphone apps, in order to collect and aggregate spatio–temporal weather data. Not all users will wish to accept these contracts if they are considered to be 'bad' in terms of how, when and for whom their cellphone sensors are used. Moreover, a contract might take away liberties the user wishes to maintain or not uphold highly–valued rights. Hence, we apply a version of the multi–level governance compliance checking framework (chapter 4 and 5) in order to automatically check whether these contracts are 'good' from the perspective of the user's wishes over how, when and for whom their cellphone sensors are used.

The rest of this chapter begins by describing the application domain in section 7.1. Then we provide an example in section 7.1.1. An architectural overview is given in section 7.2 of how the reasoning framework is used in the context of crowdsensing and describe the implementation. We finish with discussion on benefits and limitations of the application in section 7.3.

7.1. CONTRACTUALLY CROWDSENSING RAIN DATA

The SHINE (Sensing Heterogeneous Information Network Environment) project brings the perspective that there are many existing unused sensors owned by various stakeholders. At the same time, detailed environmental data is required to support governments and citizens in making decisions affected by environmental conditions. Such decisions include determining where to go to avoid flooded areas or deciding how city water infrastructure can be improved based on its current effectiveness. One possibility to provide detailed data is to buy sensors, deploy them and then collect data for a single purpose. But this is costly. SHINE aims to overcome a costly single–minded buy–deploy–sense cycle by re–purposing the existing, unused, sensors owned by independent stakeholders.

SHINE views the sensors as being heterogeneous and the stakeholders who own the sensors as being autonomous. Sensor heterogeneity means that the sensors' stakeholders own operate differently. For example, even if we only consider cellphone microphone sensors, one stakeholder might own a cellphone which runs a different operating system and therefore needs to be configured differently (e.g. ensuring data transfer is enabled or recording a video with a cellphone camera) to another stakeholder's cellphone. One way to address the heterogeneity of sensors is to ask stakeholders to ensure their sensors are configured correctly. Stakeholder autonomy means that we cannot force a stakeholder to donate the use of their sensors or to configure them in the correct way.

In SHINE stakeholder autonomy is addressed by contractually obliging stakeholders through bilateral agreement to ensure that they donate and configure their sensors to provide detailed data (e.g. rainfall locations). Specifically, in the context of this dissertation, SHINE aims to enact contracts between stakeholders and sensor networks to *crowdsource* the use of the cellphone sensors stakeholders already own, also known as *crowdsensing* [87]. Crowdsensing involves crowdsourcing existing sensing devices as a means to cost–effectively acquire detailed data in a variety of domains. For example, crowdsourcing people's cellphone battery sensors in order to obtain detailed temporal–spatial urban air temperatures [190], using microwave links in cellular networks to obtain detailed rainfall data [189] or using the cellphones in an area to form the wireless network to aggregate and transport the data [158]. The SHINE project uses contractual crowdsensing to acquire people's cellphone sensors as a means to cost–effectively acquire detailed data in a variety of domains.

We look at crowdsensing people's cellphones to obtain rain data with an implementation in a simulated prototype. In our case, rain data is obtained using a hypothetical cellphone app which uses the cellphone's audio sensors or communications links between devices to determine rainfall. The hypothetical owner of a cellphone is a human user and as such they have autonomy to join the crowdsensing system, ensure the cellphone sensors are on (i.e. the microphone, GPS and communications links) and that cellphone apps are able to transmit mobile data. The crowdsensing system binds users with contracts (a type of legal institution [208]) for use of their cellphone devices and sensors by imposing various regulations on those users and their cellphones. For example, "a cellphone is obliged to collect rain data when at a specific location and the user is prohibited from turning the sensor off until the data is collected". In return, a user might be promised access to the rain data obtained by the crowdsensing system. By requesting a large number of users to agree to such contracts and enough users agreeing to those contracts detailed data can be gathered through the crowdsourcing of sensors.

However, if there are many contracts offered for the use of a user's resource, it is unmanageable for a user to assess the details of each and every one. On the one hand, if a user agrees to a contract without reading the fine print, they are liable to engage in a 'bad' (e.g. unfair) contract. For example, a contract which requires the owner to keep audio sensors and data communications on even if the cellphone battery is depleted (i.e. below 10%). On the other hand, if contracts are rejected by default the user might miss out on the opportunity to obtain useful information (rain data). In the second case, the system is likely to not meet its aims if too many users reject contracts on the basis of not having the time to assess them. Consequently, feasibility demands the automation of contract acceptance and rejection. Yet, the automation must respect an owner's desire to maintain autonomy over how, when and for whom their cellphone contributes data. Unfortunately, existing crowdsensing systems (e.g. [64, 126]) do not let users control how their resources are used. In other words, the user needs a way to specify a policy stating regulations a contract should and should not impose for using the user's resource. We call such a policy a *use policy*. Conceptually, fitting with our earlier theoretical work, a contract is a first–level institution governing users in an MAS (the crowdsensing system), whilst the use policy specified by users is a second–level institution governing offered contracts. Hence, a contract sets the space of compliant behaviour required to ensure the collection of crowdsensed data. In comparison, a user's use policy effectively states how, when and for whom their cellphone may be used, or what obligations and prohibitions a contract may impose.

Taking a formalised use policy and contract offered to a user as input, software should automatically accept or reject the contract. Acceptance or rejection should be based on whether the contract places obligations and prohibitions on a user and their device that the user does not like in certain contexts (e.g. the context that the battery is depleted). We use a version of our computational framework from chapter 4 for checking compliance of lower–level institutions. We implemented an automated reasoner for contract acceptance/rejection based on compliance in a simulated crowdsensing prototype.

7.1.1. EXAMPLE

In the implemented contract reasoner contracts and use policies are specified in a high–level language. In principle, there are many possible contracts and use policies that can be specified. In order to exemplify the application we will present one contract and one use policy adapted from [147]. For simplicity of explanation, all institutions operate at the same level of abstraction and thus the norm abstraction reasoning from chapters 4 and 5 is not employed. In reality, the crowdsensing system owners can specify their own contracts to offer to users and users can specify their own use policies on the basis of which contracts are accepted or rejected.

The following contract and use policy is formalised using the institution syntax from chapter 4. The contract is a first–level institution $\mathcal{I}^c$ offered by the owner of the crowdsensing system, Ada, to a cellphone stakeholder called Bertrand for collecting data. The contract abstracts away from how the data is collected and sent, since this is dependent on the sensor. Data collection may or may not require user input, for example. The contract is formalised in table 7.1 and comprises the following rules:

- Ada is interested in collecting rain data for a specific location, a square, and therefore the contract obliges Bertrand's cellphone to gather rainfall data when he enters the square before leaving the square (rule Contr.1).
- Once Bertrand's cellphone has collected rain data he is obliged to send it within one minute (rule Contr.2).
- If Bertrand enters the square then the contract states that he is forbidden to turn the cellphone's sensors off until the data is collected (rule Contr.3).

Bertrand has specified a use policy, the institution $\mathcal{I}^{up}$, formalised in table 7.2. It defines the following second–order norms:

- Bertrand wishes to be paid for the use of his resource. Every time Bertrand discharges an obligation to send data, it is obliged that the counter–party, Ada, is obliged to pay Bertrand within two minutes where this obligation must be imposed *immediately* (rule UsePol.1).
- When Bertrand's cellphone battery is depleted (i.e. drops below 10%), he wishes to conserve it for more important tasks other than collecting rainfall data. Hence, when the battery becomes depleted it is forbidden to oblige Bertrand to keep the cellphone sensors on, until the battery is replenished sufficiently (rule UsePol.2).
- Bertrand wishes to maintain the liberty to move freely, hence it is always forbidden to oblige Bertrand to collect data before leaving his location (initial state fluent UsePol.3).

Table 7.1: A Contract for Crowdsensing Cellphone Sensors to Gather Rain Data

$$\mathcal{C}^c(\emptyset, enter(bertrand, square)) \ni obl(collectData(bertrand, rain), leave(bertrand, square)) \quad \text{(Contr.1)}$$

$$\mathcal{C}^c(\emptyset, collectData(bertrand, rain)) \ni obl(sendData(bertrand, rain), m1) \quad \text{(Contr.2)}$$

$$\mathcal{C}^c(\emptyset, enter(bertrand, square)) \ni pro(turnSensorsOff(bertrand), collectData(bertrand, rain)) \quad \text{(Contr.3)}$$

Table 7.2: A contract for Crowdsensing Cellphone Sensors to Gather Rain Data

$$\mathcal{C}^{up}(\emptyset, disch(obl(sendData(bertrand, rain), leave(bertrand, square)))) \ni obl(obl(pay(ada, bertrand), m5), now) \quad \text{(UsePol.1)}$$

$$\mathcal{C}^{up}(\emptyset, batteryDepleted(bertrand)) \ni pro(pro(turnSensorsOff(bertrand), collectData(bertrand, rain)), batteryReplenished(bertrand)) \quad \text{(UsePol.2)}$$

$$\Delta^{up} \ni pro(obl(collectData(bertrand, rain), leave(bertrand, square)), never) \quad \text{(UsePol.3)}$$

Intuitively, we can see that Ada's contract violates Bertrand's use policy. Firstly, Bertrand demands payment for providing data, but Ada's contract does not offer payment. Secondly, Bertrand wishes to be able to turn his cellphone sensors off when the cellphone battery is depleted, but Ada's contract requires the sensors to be kept on when collecting data. Thirdly, Ada wants Bertrand to collect data before leaving his location, however, Betrand has stated that he wants the liberty of free movement. Hence, the automated reasoner in this case would reject Ada's contract on the basis of violating the terms of Bertrand's sensor use policy.

7.2. SYSTEM OVERVIEW

Reasoning about contracts and use policies, as exemplified previously, was implemented as a part of a simulated crowdsensing prototype. In the prototype a simulated cellphone app is instantiated for each simulated user. Each cellphone app combines reasoning for resource governance and data collection. The cellphone app's architecture is depicted in figure 7.1 and described as follows in terms of interaction with the crowdsensing system as a whole and other user's individual cellphone apps:

1. Before a user's cellphone is crowdsourced in providing sensing information, a contract is offered to the cellphone to provide rainfall sensing data. The contract is ongoing in the sense that it is not for a single data point but rather multiple data points requested on–demand by the system.

2. A resource governance component, the multi–level governance compliance reasoning, accepts or rejects the contract based on whether it is compliant with a user's use policy. Each user has a different use policy and different users may be engaged in different contracts, hence cellphones do not necessarily uniformly accept/reject a contract. If the contract is accepted, then the cellphone joins the crowdsensing system.

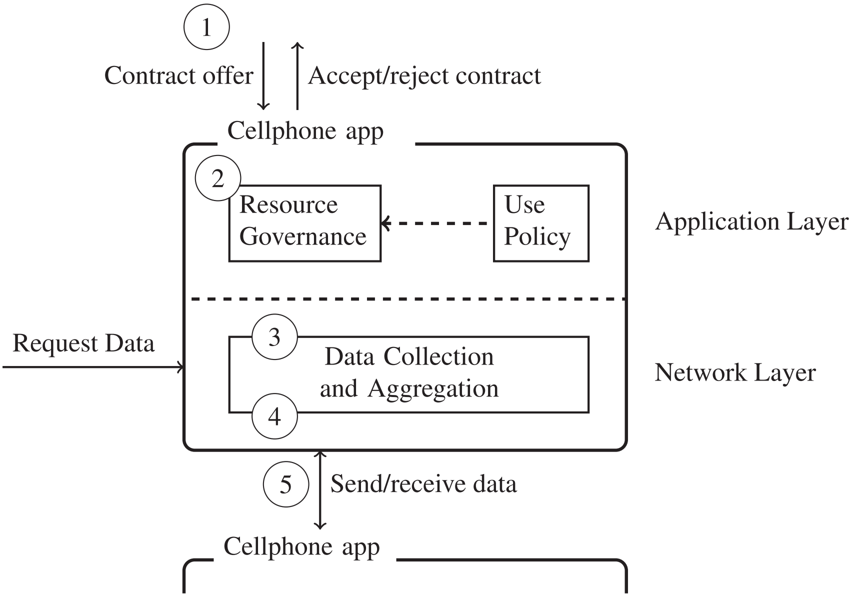

Figure 7.1: A cellphone app is modelled in the simulated prototype comprising two layers, an application layer for the governance of the user's resource (the cellphone and its sensors) and a network layer for using the resource (collecting, aggregating and sending data).

3. A cellphone which has accepted a contract is requested for data. When data is requested, the cellphone forms a cluster with nearby users that have also accepted a contract for crowdsourcing rain data in the vicinity. The cluster collectively decides how many and which users collect data, based on the granularity of data required according to the algorithms described by the authors of [158].

4. Once the data is collected, it is sent using one of two methods. If data delivery speed is unimportant, the cellphone communicates with other cellphones that have accepted a contract to form an ad–hoc network to transmit the data whilst conserving energy (compared to using a traditional cellular network). In this case, users' cellphones are also crowdsourced in helping to transport the data, where one cellphone receives data from other cellphones, (aggregates) packages it together and forwards it to the next cellphone in the ad–hoc network. If speed is more important, cellphones communicate using the cellular network.

5. In either case, other users' devices that are contracted in providing data also receive rain data as a part of the contractual agreement.

The prototype crowdsensing system is implemented in the NetLogo[1] simulation environment. In our prototype simulation, the population of use policies assigned to each user is

[1] `http://ccl.northwestern.edu/netlogo/`

Figure 7.2: Visualisation of data aggregation. Lines between users (not to scale) indicate data aggregation, adjacent users with the same colour indicate a cluster where only one in that cluster is providing data. Rainclouds are indicated with blue and black squares. The map is taken from ©OpenStreetMap.

configurable. Moreover, the contractual rules are also configurable. This is true both before and during runtime. Consequently, use policies can be edited dynamically to demonstrate their representation and how changing various second–order norms affects the acceptance and rejection of contracts. This supports policy makers (i.e. the owners of the crowdsensing system) in assessing the acceptability of contracts and the level of users likely to join the crowdsensing system before run time. However, the main point of the system is as a *proof of concept* rather than to conduct empirical research.

The visualisation of the prototype is shown in figure 7.2, where each cellphone owner that has accepted a contract and joined the crowdsensing system is depicted as a figure, a map represents the geographical area and rain clouds are represented with blocks in shades of black to light blue. Adjacent contracted sensing devices that have formed a cluster in order to select a single cellphone to collect data for a geographical area, share the same colour. Users that are crowdsourced into forming an ad–hoc network for delivering the data are linked with lines (a central station to which the data is eventually sent is denoted as a circle with alternating black and red inner circles).

7.3. DISCUSSION

In this chapter we have described a proof of concept for the application of the compliance checking system, contributed by chapters 3 and 4, to the contracted crowdsourcing of cellphone sensors for detecting rain levels. In principle, the same system can be used to automatically govern contracts for crowdsourcing other types of data, such as temperature, crowd levels or photographic weather reports. However, although this chapter illustrates the proof of concept, it does not assess the usability of the PARAGon framework and hence its applicability is *in principle* only. Future work should investigate whether users can adequately describe governance of socio–technical systems and the accuracy of the framework in correctly governing resources to meet user's expectations. Consequently, this chapter only argued the case for multi–level governance automation's applicability, in one case and a limited form, to SHINE–like systems.

To re–iterate, the main advantage of using the multi–level governance compliance checking components of the PARAGon framework is that it supports users in specifying how, when and for whom their devices are used. Users are able to state in which contexts specific obligations and prohibitions should (not) be placed on themselves and any counter–party. Moreover, a use policy, by containing temporal second–order norms, is able to declare temporal requirements for a contract to be accepted. For example, a user can specify that they should be paid in advance of being obliged to provide data, keep sensors turned on, etc. Moreover, the abstraction reasoning presented in chapter 3 and 4 was not implemented for this prototype, but in principle would support a user in defining a use policy using abstract terminology they understand. Hence, the PARAGon framework supports users in specifying use policies and automatically accepting or rejecting contracts governed by such use policies.

The main limitation of this approach is that a single occurrence of non–compliance in a certain social context is aggregated to the contract being non–compliant as a whole. In reality, contracts are neither always accepted nor rejected outright due to a single flaw from the perspective of one party nor are contracts binarily good or bad. Rather, to an individual there are many levels of preference in how good or bad (compliant) a contract is. Consequently, contracts are often negotiated in order to reach a compromise between two parties. The reasoning implemented does not support negotiating in terms of preferences for contracts for the main reason that use policies do not elicit a preference relation between offered contracts. Consequently, the formalism for compliance checking in multi–level governance would need to be extended to reasoning about different levels of ideality in order to form a partial–ordering between contracts. One way would be to adopt the approach in [230] where contrary–to–duty norms are used to represent different levels of ideality (e.g. you ought not murder, if you murder you ought to do so gently [83]). This approach is in a strictly non–temporal setting and so would need extending to reasoning about temporal norms. Another approach is to combine a temporal logic with a preference logic as in [21], describing preferences over temporal states of affairs (in our case, the obligations and prohibitions imposed by a contract) and do away with higher–order norms altogether. Adopting one of these approaches when combined with the norm abstraction reasoning presented earlier, would enable users to specify abstract temporal preferences over contracts as a use policy.

8

Conclusions

Of course, errors are not good for a chess game, but errors are unavoidable and in any case, a game without errors, or as they say 'flawless game' is colourless.

Mikhail Tal

When you see a good move, look for a better one.

Emanuel Lasker

This dissertation contributed formal accounts of institutions involved in the governing of governance, which collectively comprise the PARAGon framework. The formalisations focussed on two governance types involved in the governing of governance. Namely, institutional design governance and institutional enactment governance. These governance types are found in the social world, where institution designers are governed in how institutions *should* be designed and when institutional changes *can* be enacted. Concerning how institutions should be designed, this dissertation formalised compliance in multi–level governance. Moreover, this dissertation contributed reasoning for diagnosing and rectifying the underlying causes of non–compliant institution designs. Concerning when institutional changes can be enacted, this dissertation formalised secondary legal rules defining the space of possible institutional change enactments. The primary objective, namely formalisation, was complemented with two practical objectives. Firstly, reasoning about realistic institution designs. Secondly, a practical means to reason about governance, which means a computational implementation or an obvious way to build a logical model. Whilst this dissertation contributed prescriptive research, it is nevertheless important to interrogate the intuitions involved. To this end, PARAGon was assessed against a number of real world and imagined institution designs. Together, these contributions aimed to extend the knowledge in the area of institutional reasoning by formalising a new aspect of institutions, in a practical way, and assessing the formalisation intuitions by way of case study.

Central to this dissertation's results was a shift in perspective and a new view on the constitutive rules that comprise institutions. This dissertation's perspective shifted from institutions guiding agent behaviour typically focussed on in the field, to institutions guiding and governing institutional design and enactment. This dissertation's formalisations of institutions seen from this new perspective resulted in a new view of constitutive rules' roles in institutions. Whilst much of the literature viewed the role of constitutive rules as making abstract social actions and other institutional facts possible, this dissertation enlarged the responsibility of constitutive rules. For institutional design governance, this dissertation expanded constitutive rules' responsibility from defining abstract concepts to also indirectly defining the abstract meaning of concrete norms in multi–level governance. For institutional enactment governance, this dissertation expanded constitutive rules' responsibility from making social actions and facts possible to making social actions of institutional change enactment possible. These two new views of constitutive rules are central to this dissertation, as we will discuss further in this chapter.

The rest of this chapter continues by concluding the dissertation with a discussion of results in Section 8.1, we discuss how the results can be applied to the SHINE project under which this research was conducted in Section 8.2, then we discuss future work in Section 8.3 and finally we reflect on the dissertation as a whole in Section 8.4.

8.1. DISCUSSION OF RESULTS

Whilst it might have appeared that much of governance already had strong logical foundations, in Chapter 1 and Chapter 2 we saw that governing governance was, until now, mostly formless in the literature. Hence, this dissertation aimed to provide precise definitions for institutional governing of governance in terms of institutional design and enactment governance. This lead to the following overarching research question:

How can institutional design and enactment governance be supported with formal reasoning?

The overarching research question formed the basis for the sub–research questions presented in Chapter 1, which are listed as follows:

- **Sub research question 1:** What is a suitable representation to specify institutional design and enactment governance?
- **Sub research question 2:** How can we formalise compliance in multi–level governance?
- **Sub research question 3:** How can institutional design compliance in multi–level governance be computationally verified?
- **Sub research question 4:** How can non–compliant institution designs be explained in order to rectify non–compliance according to the institution designer's objectives?
- **Sub research question 5:** How can we formally define when legally valid institutional change enactments occur?

In the remainder of this section we discuss what we did to answer these research questions.

WHAT IS A SUITABLE REPRESENTATION TO SPECIFY INSTITUTIONAL DESIGN AND ENACTMENT GOVERNANCE?

In order to answer this question, we wanted a *natural* way to capture the representation for *temporal* institutions operating in the following two governance types identified in Chapter 1:

- **Institutional design governance:** multi–level governance where abstract temporal regulations contained within higher–level institutions govern the regulatory effects of lower–level institutions comprising concrete temporal regulations.
- **Institutional enactment governance:** the legality of institutional rule change according to secondary legal rules in a temporal setting.

The question was answered for both of these governance types by contributing a high–level representation for institutional design governance in Chapter 3 and in a limited form in Chapter 5, and for institutional enactment governance in Chapter 6. The notion of institution used in this dissertation was founded on Searle's counts–as rules [215] of the form "A counts–as B in context C". In our case, counts–as rules were used to represent an institution's temporal dynamics and played a central role for both institutional design and enactment governance. In order to provide a uniform representation for counts–as rules, we based our institution representations on those first proposed by the InstAL framework [42, 43].

A representation specific to institutional design governance was contributed in Chapter 3 for multi–level governance, where counts–as rules both define which concrete concepts constitute abstract concepts and modal norms. The requirement for a natural representation specific to multi–level governance was met in three ways. Firstly, by offering the option to define explicit or implicit abstract norms governing concrete normative effects. Secondly, by adopting modal norms, which as we discussed in Chapter 3, provide a natural generalisation to higher–order norms for the explicit governance of concrete normative effects. Thirdly, by *not* requiring any explicitly represented links between concrete and abstract regulations and instead pushing the burden of determining relations between concrete and abstract regulations to a semantics. The temporal requirement was addressed by adopting temporal, immediate and indefinite norms. We showed how institutions defined according to our representation for multi–level governance have an, arguable, natural correspondence to the written law in Chapter 3. In some cases explicit, but controversial, rule formalisation choices were necessary in the case studies examined. Specifically, in order to collapse obligations/prohibitions to non normative social facts, which although we provided argumentation for in Chapter 3, due to having a controversial meaning is a representation *choice*.

A representation specific to institutional enactment governance was contributed in Chapter 6. The central representation construct was counts–as rules used to represent Hart's [117, p.96] secondary legal rules. As we discussed in Chapter 2, following the conceptualisation of Biagoli [20], secondary rules make the social action of changing rules possible in the same way counts–as rules make social actions in general possible. In our representation, we contributed *rule–modifying counts–as rules*, which ascribe rule modifications in the past, present or future conditional on a social context. The requirement for a natural representation was addressed in two ways. Firstly, by defining a richer social context to represent rule–modifying counts–as rules for real legal case studies. For example, to represent a condition, such as found in EU law [48, Art. 7], on past rule modifications only being possible if they would not, hypothetically, cause an innocent person to become criminalised in the past. Secondly, by not assuming any special representation to define secondary rules which modify other secondary rules, since there is no evidence supporting a special representation being required in the legal case studies we looked at. A number of real–world case studies in Chapter 6 were looked at to show an, arguable, natural morphism between the written law's secondary rules and our formalisation of those rules in the representation we propose.

In summary, PARAGon contributed a representation which limits assumptions for the governance types looked at, by providing a representation which conforms to written or mental institution's representation and court interpretations. Where necessary we offered a representation choice rather than making strong representation assumptions. There may be other ways to answer this research question with different representations. However, our requirement was for a natural representation for temporal institutions and hence a suitable representation without requiring any unnecessary constructs was contributed. That is, as far as we are able to tell for the case studies we looked at in Chapter 3, Chapter 4, Chapter 5 and Chapter 6, our representation requires no further simplification in order to be natural nor any further complication in order to represent real–world institutional design and enactment governance.

HOW CAN WE FORMALISE COMPLIANCE IN MULTI–LEVEL GOVERNANCE? In order to answer this question, PARAGon contributed a semantics for operationalising multi–level governance compliance in Chapter 3. The semantics defined when a lower–level governance institution imposing relatively concrete regulations has regulatory effects that are non–compliant with higher–level institutions' abstract regulations.

Generally, the semantics assesses collective regulatory effects of lower–level institutions in different social contexts for whether they violate abstract regulations in higher–level institutions. Our semantics adopts the advice of David Makinson [164], "no logic of norms without attention to a system of which they form part". That is, norms are understood in terms of the other constitutive rules contained in the same system. Constitutive norms are used to re–interpret the concrete regulatory effects in lower–level institutions for whether they have abstract meanings in higher–level institutions. The abstract meaning of a lower–level institution's concrete normative effects (caused by its concrete norms), according to constitutive rules, determines if those concrete norms violate abstract obligations and prohibitions in higher–level institutions.

The abstract meaning of a lower–level institution's concrete normative effects is heavily dependent on the social *context* lower–level institutions' obligations and prohibitions are imposed in *from the perspective of* higher–level institutions. To return to a previous example, an obligation imposed by the UK's Data Retention Regulations to store communications metadata is abstractly interpreted by the Charter of Fundamental Rights' as unfair data processing *in the context that* the person who the data concerns has not consented. In turn, unfair data processing is prohibited by the Charter of Fundamental Rights, hence the UK's Data Retention Regulations institution violates a prohibition on unfair data processing when it imposes an obligation to store communications metadata *in the context that* the person who the data concerns has not consented. Consequently, not only is the abstract meaning of a lower–level institutions normative effects context sensitive, but so is the compliance of a lower–level institution's norms.

Since social contexts are important, we chose to take into account their dynamics. That ism we chose to interpret institutions operating in multi–level governance as state–event transition systems where an institution evolves from one state to another, meaning social contexts also evolves over time. Consequently, our notion of compliance rejects a naïve static comparison between different regulations which would focus on individual rules, in favour of focusing on the combination of regulatory effects and social contexts they are applied in.

We contributed one definitional answer to the aforementioned research question, but there may be many more. Alternative definitions may vary significantly due to representation choices. For example, in our representation we adopted modal norms for the simplicity of representing higher–order norms. An alternative norm form, evaluative norms as discussed in Chapter 2, does not have explicit deontic modalities and hence in this case a notion of compliance in multi–level governance might be defined differently. Moreover, there may be regulations found in multi–level governance that are not captured by our formalism.

To summarise, we answered this research question by defining compliance in multi–level governance as follows. An institutional design is compliant in multi–level governance *if and only if* its concrete regulatory effects in different social contexts have compliant abstract meanings in higher–level institutions according to those higher–level institutions' interpretive counts–as rules. This notion of compliance accounts for the different concrete and abstract ontological views between governance levels and the dynamic *meaning* of lower–level regulatory effects from the perspective of higher–levels.

HOW CAN INSTITUTIONAL DESIGN COMPLIANCE IN MULTI–LEVEL GOVERNANCE BE COMPUTATIONALLY VERIFIED?

In order to answer this research question, PARAGon contributed a computational framework for determining compliance in multi–level governance, presented in Chapter 4. The computational framework provided a mapping from the formal representation of institutions in a multi–level governance relationship and their semantics to a corresponding executable ASP program. Institutions were represented in the computational framework as rules in a similar fashion to their formal counterparts. The multi–level governance semantics were represented as general ASP rules to capture the temporal dynamics of institutions and *specific* ASP rules to represent the semantics for the abstraction of norms. Effectively, the ASP rules representing norm abstraction semantics flattened the deontological counts–as function, which defines how concrete normative effects are interpreted more abstractly, presented previously in the formal component of the framework for multi–level governance compliance (Chapter 3). Whist a non–flattened and more general form might have been preferable, we found it was not possible due to current limitations of ASP. Naturally, we contributed theorems and their proofs demonstrating that the computational framework is sound and complete with respect to the formal framework for multi–level governance compliance checking.

The computational framework was complemented with an implemented compiler. The compiler built on the InstAL framework's [42, 43] implementation, which allows institutions in a multi–level governance relationship to be specified in a high–level non–procedural language and compliance to be automatically checked. As we saw, compliance in multi–level governance is highly dependent on social context. Consequently, keeping with the aim of practicality, this means through our implementation we can reveal to institution designers in which social contexts their regulations are non–compliant. If certain social contexts are unreachable in reality and therefore compliance for these contexts is irrelevant, a user can decide based on the presented contextual information whether to ignore these contexts. Since we used ASP, it is possible domain knowledge can be incorporated into the resulting ASP programs in order to constrain compliance checks to only reachable social contexts. Consequently, the practical benefits were realised with an implemented and corresponding computational framework, which provides automated and contextualised compliance checks to users.

We only contributed one answer to this research question. Other corresponding computational frameworks may be possible. For example, using a higher–order logic as a meta–language to provide an axiomatisation of our formal semantics and a higher–order logic theorem prover to act as the computational implementation. Such an approach has been used to embed other logics, such as modal logic [18]. Speculatively, this approach could yield greater benefits by potentially enabling a computational framework comprising general rules/axioms rather than a flattening of the deontological counts–as function, which requires re–compilation every time institutional rules are changed. An entirely different answer to the research question could be to not provide a computational implementation, but to instead investigate the computational aspects of the formal framework. For example, decidability and complexity. In effect, our implementation shows our formal semantics is decidable, since ASP is decidable for finite Herbrand universes [92] and we proved a correspondence between the ASP implementation and the formal framework. However, an alternative approach could have been to contribute a decision procedure, which would have removed the need for elaborate and recursive transformations between the formal representation and ASP–based implementation. Hence, there may be alternative and perhaps more elegant solutions to this research question.

By contributing a computational framework, we were able to formalise and execute a substantial portion of two related real–world multi–level governance legal cases. Specifically, the first case where the EU Data Retention Directive was enacted in 2006 [71] to coordinate EU member state legislation in requiring communications metadata was stored and where the UK's implementation, its Data Retention Regulations [226] were enacted and found to be compliantly implement the directive in 2009. Moreover, the second case where in 2014 the Data Retention Directive was found to be non–compliant by the European Court of Justice [70] with the European Charter of Fundamental Rights [72].

By executing the case study, using interpretations provided by court judgements where possible, we were able to evaluate both our formal and computational multi–level governance compliance checking frameworks. One caveat is that the results for executing a case study can be highly dependent on how institutions are represented. Hence, we opted for a formalisation that arguably has a clear correspondence to the written law and court judgements. Bearing this in mind, we were able to see that we gained the same results as the real–world judgements of the European Court of Justice, using what we argue is a natural corresponding representation. Consequently, we contributed a computational implementation that demonstrates the formal definitions of compliance in multi–level governance are arguably 'correct' and provides practical benefits with automated compliance checking.

HOW CAN NON–COMPLIANT INSTITUTION DESIGNS BE EXPLAINED IN ORDER TO RECTIFY NON–COMPLIANCE ACCORDING TO THE INSTITUTION DESIGNER'S OBJECTIVES?

In order to answer this research question PARAGon contributed a formal definition of explanations for non–compliant institutional designs and a computational mechanism to *induce* those explanations in Chapter 5. That is, by generating explanations, treating those explanations as hypotheses for rectifying non–compliance and then testing those hypotheses for support over a trace of events and therefore multiple social contexts. The contribution means non–compliance can be automatically rectified by adopting the explanatory rectifications for non–compliance as institutional design changes.

The search for explanatory non–compliance rectifications was formalised as an Inductive Logic Programming (ILP) problem. We addressed the inconvenient idea that an institution designer prefers explanatory rectifications for non–compliance that keep an institution design as close to their original intentions, in order to ensure the proposal is of practical benefit. The idea being that our formal notion of explanatory rectifications for non–compliance are those that minimise changes to an institution's constructed social realities. Thereby, explanations for non–compliance offer rectifications that keep as closely as possible to the designer's original intentions, assuming those intentions were correctly captured by the original institutional design. Moreover, we formalised the notion of explanations as also optimizing, to a secondary degree, institutional design change simplicity. This means explanations are preferred if they are more compact, in the sense of optimising fewer rule changes explaining greater instances of non–compliance. Overall we formalised explanations as comprising four types of rule change which can be combined: adding rules, removing rules, adding rule constraints and removing rule constraints. Hence, we formalised explanatory rectifications for non–compliance as changes to rules that adhere to an institution designer's intentions, and are the simplest and most compact explanatory hypotheses.

We addressed explaining non–compliance from a practical perspective. That is, by implementing the search for the hypotheses that provide possible explanations and rectifications for non–compliance, as an ILP problem in ASP using existing embedding techniques. The implementation alleviates an otherwise arduous task of an institution designer in trying different institutional re–designs for the one which rectifies non–compliance in the best way according to our own optimisation criteria (closeness to the original design and simplicity). The general idea of the implementation is to take an ASP program defining a non–compliant institution and compile that program into another ASP program where hypotheses for rectifying non–compliance can be generated and tested for support. For an institution designer, the implementation provides flexibility in specifying for which social contexts rectifying non–compliance should be found and hence guide the process of finding explanatory rectifications.

In order to see how our formalisation of explanatory non–compliance rectifications behaved, we formalised a case study based on a fictional institution for a socio–technical system which in turn was governed by real–world and fictional laws. The system discovered a number of simple explanations for why the socio–technical system institution was non–compliant. These included the discovery of general design faults, which might potentially explain further issues the designer does not desire, removal of problematic rules and changes to existing rule constraints, and explanations comprising rule additions to ensure regulatory effects before rectification would still exist after rectification if they do not cause non–compliance.

We only contributed one possible answer to the aforementioned question based on the idea of inducing hypotheses for non–compliance. There may be other possible answers, such as by defining what constitutes an explanation differently. For example, rather than defining explanation as hypotheses comprising institutional rule changes, one possibility is to combine argumentation and abduction to provide explanations as causal stories (e.g. as proposed for legal cases [19]). In this approach, an explanation is a chain of inferences showing how abduced probable causes of a true proposition provide an explanation of that proposition. Hence, a narrative causal chain could instead be used in this way to provide explanations for non–compliance. We do not attempt to exhaust all the feasible ways to define explanations for non–compliance, but note formal argumentation could provide alternative answers.

Together, the formal notion of explanations for non–compliance and the implementation for finding such explanations complement each other. The formal aspect can be tested by the implementation with various case studies to determine if the formal notions give intuitive results. The practical aspect is supported by addressing the need to find those solutions for non–compliance which minimise design change and offer the simplest explanations for understandability. We tested both of these aspects using a case study combining fictional and real–world rules.

HOW CAN WE FORMALLY DEFINE WHEN LEGALLY VALID INSTITUTIONAL CHANGE ENACTMENTS OCCUR?

In order to answer this research question PARAGon contributed a semantics that defined when institutional change enactments are valid in Chapter 6. In this context, by institutional change enactment we mean changes to an institution's rules over time. An institutional rule change is valid if it is ascribed by an institution's secondary rules. These secondary rules were represented as a special type of counts–as rule of the form "A counts–as modifying a rule in the past/present/future in context C", which ascribe modifications to counts–as rules including other secondary rules. The semantics defined when social actions of institutional counts–as rule changes are ascribed, including to other secondary rules, and how the institution transitions from one version to another when rule changes are validly enacted.

The formalisation gave meaning to secondary (counts–as) rules. The secondary counts–as rules we formalised include a condition on a richer social context not given any formalisation, as far as we are aware, previously. In particular, social contexts acting as conditions on past institutional states, past institution versions and *hypothetical* rule change effects. These conditions were used to represent cases such as institutional rule changes occurring conditional on those changes not criminalising formerly innocent people in the past. In order to deal with such expressivity we proposed a semantics that defines a rule change as occurring if the necessary conditions are met and all of the effects of the rule change are consistent with those necessary conditions.

We acknowledged that in reality secondary rules can make it possible for past, present and/or future rule changes to be made including to other secondary rules. Consequently, our semantics dealt with difficult cases where changes to secondary rules potentially cause other, previously established, rule changes at various points in time to be unravelled or potentially further rule changes to be made. Moreover, our semantics handled cases where complex interactions between rule changes would, in a classical setting, cause inconsistency. For example, the case where the only rule ascribing rule modifications ascribes its own removal. In our case, we took the perspective that if a rule change causes inconsistency with the conditions for its own ascription then the rule change is not legally ascribed (i.e. is invalid) and thus does not occur in the first place.

We tested our definition against a number of case studies based on EU and UK law, as well as edge cases where we would expect there to be issues with an incorrect definition. The real world case studies demonstrated that through a simple representation we were able to reason about difficult kinds of rule change governance. For example, hard constraints on there being no changes to criminalising laws in the past, requiring a form of hypothetical reasoning to classify rule change effects. From the edge cases we saw that complex cases where rule changes interact, which are conceivably difficult to reason about without a mechanical process, are arguably handled intuitively by our formal account.

We did not attempt to exhaustively answer the research question. Instead, we offered a single formalisation of when institutional rule changes occur according to secondary rules. Other formalisations may be possible and, indeed, they exist for rule modifications in non–temporal settings [28] and rule modifications in temporal settings [100, 102] but with less expressive secondary rules. Moreover, an equivalent definitional answer could be given to this research question using a different formalisation. It's possible our semantics could be defined on top of an existing logic, either at the meta–level or object level. For example, we could try to adopt or modify default logic [203] by viewing conditions on rule change being ascribed as defaults that must hold before and after rule change. In our case, we chose to contribute a formal definition with a semantics that stands on its own, requiring no understanding of pre–existing logics aside from some rudimentary notions of sets and functions.

To summarise, we contributed a formal definition to answer this research question. Informally, a rule change occurs *if and only if* it is ascribed by secondary legal counts–as rules conditional on a social context, which can include hypothetical rule change effects, that holds before and after the rule change takes place. Our account was assessed against a number of case studies, including edge cases to assess the intuitiveness of results.

8.2. APPLYING THE RESEARCH TO THE SHINE PROJECT

The research conducted in this dissertation was supported by the SHINE project of TU Delft. The project aimed to develop techniques for acquiring and coordinating large numbers of heterogeneous data resources that already exist in the environment (e.g. cellphone sensors, radars and people) in order to collect environmental data. This dissertation contributes techniques for formalising governing governance, which is arguably suitable for governance and coordination of 'SHINE–like' sensor systems.

An implemented application of the research was presented in Chapter 7 for a simulated prototype as a proof–of–concept. In this application automated compliance checking in multi–level governance was used to determine if contracts offered to individuals to crowdsource their cellphone sensors should be accepted on the basis of being compliant with use policies. These use policies would be, conceivably, specified by cellphone owners stating how, when and for whom their cellphone sensors may be used. In this setting, the research was applied to giving people control over how their sensors are contributed to crowdsourcing systems without requiring sensor owners to arduously read the fine–print of the offered contracts.

The research in Chapter 5 was also applied to a fictional institution used to govern a crowdsourcing system as a whole, which in turn was governed by external laws partially inspired by real–world legislation. In this application we demonstrated an automated system to find explanations for why the crowdsourcing system institution design was non–compliant with external laws.

The research on compliance in multi–level governance in Chapter 3 and Chapter 4 was applied to a real–world multi–level governance case study. In this case we showed how the PARAGon framework was able to represent and reason about the EU Data Retention Directive [71] and its non–compliance [70] with the EU Charter of Fundamental Rights [72] due to requiring the storage of people's communications metadata without their consent, which took away the right to respect for private and family life. In this case, the relevance was showing how we can check the compliance of laws concerning the collection of data, in much the same way a SHINE institution, which aims to guide MAS participants in contributing data, might be found non–compliant with similar external human rights and liberties laws. Hence, we applied the framework to a simulated prototype and case studies concerning SHINE–systems and data collection.

Outside of these realised applications, the multi–level governance compliance reasoning presented in Chapter 3 and Chapter 4 offers a *potential* way to contribute automated reasoning for an architecture applied to governing heterogeneous sociotechnical systems. SHINE is concerned with heterogeneous sensor systems, which comprise different types of sensors. Rather than attempting to apply a single institution to govern heterogeneous sensors, which due to different behaviours might be unsuitable, multi–level governance offers a way to treat heterogeneous groups of sensors (e.g. crowdsourced cellphone sensors, a group of weather radars, etc.) as separate homogeneous self–governing sensor sub–systems. The idea is that each sensor sub–system has its own institutional governance designed by the stakeholders of that sub–system (e.g. sensor owners). Then, a SHINE institution is designed to govern institutions governing each sensor sub–system towards over–arching aims with abstract regulations. Thereby, multi–level governance delegates the design of sensor sub–system institution designs, removes problems of a single 'one–size–fits–all' institution design and coordinates separate sub–system institution designs towards overarching data collection objectives. In this scenario, this dissertation supports governance of sensor sub–system institution designs with automated multi–level governance compliance checking, but further investigation is needed to determine if automating such governance can work for heterogeneous sensor 'systems of sub–systems' in practice.

The institutional enactment governance presented in Chapter 6 provides a *potential* way for stakeholders in SHINE sensor systems to specify flexible and automated governance of sensor systems' institutional change enactment. This means, system stakeholders are able to use the formal reasoning to determine how institutional changes can be enacted according to an institution's secondary rules when changing sensor aims require changed regulations. Moreover, stakeholders have flexibility in defining the rule change process for a given governed sensor system. The main benefit being, owing to heterogeneity between different sensor systems, alternative rule change processes for each sensor sub–system can be defined and reasoned about. For example, one system's rule change process can be defined as a direct democratic vote with certain constraints (e.g. not removing the right to a democratic vote from participants in the past) and another as requiring elected technocrats to make a collective decision. Hence, although untested, our secondary rule formalisation potentially allows automated operationalisation for a diverse range of governance and regulatory change enactment styles to be defined for different sensor systems, in a way that enables those system's regulations to adapt, as deemed appropriate, to new aims and needs.

None of this is to say that this dissertation contributes an exhaustive application to the governance of SHINE–like sensor systems. However, what has been shown is that in at least one area, the governance of contracts for crowdsourcing, the framework was applied in a prototype. Moreover, the framework has been applied to case studies relevant to sensor systems such as those envisaged by the SHINE project. Finally, there are potential avenues for future applications to SHINE–like systems.

8.3. FUTURE WORK

Future work is split into analysis of the formal frameworks contributed by this dissertation, improvements to this dissertation's contributions and new research lines influenced by this dissertation's contributions.

8.3.1. ANALYSIS

The formalisms developed in this dissertation have been shown to meet various properties, such as there existing a sound and complete logic-programming implementation. Such properties were provided in order to demonstrate the expectations, set out in the introductions of the relevant chapters, have been met. However, there still remains work to be done on general properties that give a better understanding of how institutions behave under the defined semantics.

The formalism defining compliance in multi-level governance, in Chapter 3 could be analysed for properties relating to inconsistency, (non-)compliance and redundancy. Inconsistency can occur in two senses. The first type of inconsistency that should be investigated is, what we can call, non-deontic inconsistency where due to the operations defining the semantics being undefined in some cases (e.g. the event generation operation) there is consequently no model for those cases. The second type of inconsistency is what we call deontic inconsistency, where it is impossible to discharge all obligations/prohibitions, or worse that some obligations/prohibitions are necessarily violated no matter what event occurs. Properties that coincide with the dischargement and violation of abstract and higher-order norms should also be investigated. For example, if an abstract norm is violated then what concrete norms, if any, must also necessarily be violated and vice versa? Finally, an important aspect of an institution or normative system addressed in the literature is *redundancy* [22, 54] – when would removing a counts-as rule have no effect whatsoever on how the institution evolves over time (for all traces of observable events)? Whilst norm redundancy has already been formalised and investigated, it would be interesting to investigate given the novel semantics of the multi-level governance compliance formalisation if there are general properties that coincide with a rule being redundant. Investigating properties relating to inconsistency, (non-)compliance and redundancy for the formalisation of compliance in multi-level governance would give a better understanding of how the formalisation behaves, in particular given the novel abstraction semantics, and some useful properties for understanding how to improve an institution operating under those semantics.

The formalism defining the legality of rule change, in Chapter 6, could be analysed concerning (il)legal rule change, temporal dynamics and rationality. To recap the informal notion, a rule change is legal "*1. if the rule change is ascribed by counts–as rules, conditional on a context taking into account the potential changes to the context the rule modification would make 2. taking into account past/present/future rule modifications effect on counts–as rules and thus the context in which rule changes are conditional on.*" However, such informality still leaves future investigation of formal properties that coincide with (il)legal rule change. Moreover, in terms of temporal dynamics, it is important to analyse under which conditions changes occurring in the future can affect the past, including past states and also the legality of past rule changes. In particular, concerning both temporal dimensions of the formalism. On the one hand we showed future rule changes can make past changes in the *same version* become (il)legal and therefore potentially no longer occur. On the other hand we did not ascertain under which fragments of the language future rule changes are guaranteed to not affect the legality of rule changes in *past versions*. Further properties should be analysed concerning rationality. Famously, the AGM belief revision postulates [3] define a notion of rationality for belief revision operators and a related set of postulates has been proposed for norm revision [25]. It is important to analyse whether our formalisation of ascriptive legal rule change meets these postulates. However, since we are dealing with rules that ascribe the legality of changing other legal rules and not just ‘straightforward’ norm change, it is possible that the postulates are not met and moreover that a slightly different set of postulates may be more appropriate that do hold for our formalisation. Investigating properties relating to the legality of rule change, temporal dynamics and rationality would give a better understanding of what our formal definitions really mean at a deeper level.

In addition to analysing the behaviour of each individual formalisation, it is also important to formally compare with other formalisations. One possible way to do this would be to abstract the semantics. For example, by using Tosatto et al’s [223, 224] graph-based formalism where obligations, permissions and institutional facts are nodes and rules are edges between those nodes. The idea would be to take existing abstractions of formalisms using Tosatto et al’s abstraction and compare them to an abstraction of the formal semantics contributed by this dissertation. The benefit would be not just a formal comparison between the PARAGon framework and other frameworks, but also a comparison where patterns of reasoning and differences between them are visualised. However, this type of analysis requires further investigation since there, to the best of our knowledge, appears to be no work on abstracting systems of the ‘type’ we formalise. In particular in Chapter 6 where an institution evolves over two timelines and there can be complex interactions between past, present and future institutional facts. In order to analyse differences between our formalism and others, a means to abstract our semantics should be found, which is a significant undertaking and left for future work.

8.3.2. IMPROVEMENTS

PUTTING AUTOMATED MULTI–LEVEL GOVERNANCE INTO PRACTICE

In Chapter 3 and Chapter 4 we looked at reasoning about compliance in multi–level governance. A focus was placed on reasoning about abstraction at different levels of governance. This aspect of the PARAGon framework was assessed against a real–world case–study for the governance of human societies and governments.

An avenue for future work is to determine if the same framework can be applied to artificial societies. This would result in an formalised and automated operationalisation of two proposals. First, the proposal of what Pitt et al. [197] call polycentric governance for socio–technical systems. Here, they argue 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, in our own wording, separate lower level institutions should be designed appealing to subsidiarity. These institutions should be compliant with overarching institutions abstractly prescribing what regulations should be implemented. Secondly, Dignum et al. propose the OMNI design methodology framework for institutions/organizations governing artificial MASs [62]. 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. Then, abstract norms implementing these statutes. Implementing the abstract norms are concrete norms. In this methodology an assumed function that concretises abstract norms is used. PARAGon offers a semantics for reasoning about abstract norms that could potentially be used to weaken or remove the assumption of a concretising function. Future work should extend and apply the PARAGon framework to further operationalise and develop the ideas presented in these papers.

ENHANCING INSTITUTION ABSTRACTION REASONING

Another important future development is extending the semantics for abstract regulation reasoning presented theoretically in Chapter 3 to support ontological alignment of institutions. In our proposal lower–level institutions' regulatory effects were re–interpreted at the abstract level defined by higher–level institutions. For example, obliging storing communications metadata in a lower level institution is abstracted to obliging storing personal data in a higher–level institution. In one sense, this means lower level institutions' regulations were *aligned* with the abstractions defined by higher–level institutions.

However, we assumed that the concrete terminology shared by lower and higher–level institutions have the same meaning and are already ontologically aligned. For example, storing metadata in the lower–level institution coincides with storing metadata in the higher–level institution. Thus, if the lower–level obliges storing metadata, and the higher–level views storing metadata as storing personal data, then from the higher–level institution's perspective the lower–level is obliging that personal data is stored. In general, ontologically aligning institutions is an important problem that needs addressing for comparing regulations between institutions.

Aside from alignment, abstraction reasoning should also be improved to address further deontic concepts. We focussed on how to abstract obligations and prohibitions, based on counts–as rules describing the abstraction of events and fluents. However, there are many other deontic concepts, as described by Hohfeld [120], that play an important role in institutions. They include accounting for the concepts of right and liberty. For example, a right to life correlates with a prohibition on others to take life away. The liberty to choose who you love depends on there being no prohibition on choosing who you love. Each of these might be involved in multi–level governance in the form of obliged/prohibited rights and liberties, or rights and liberties represented in constitutions which require obligations/prohibitions to (not) be imposed in lower–level institutions. Such Hohfeldian notions of rights and liberties should be investigated for abstracting and governing.

Likewise, concepts of legal power should also be accounted for. Legal power, is the ability to (legally) affect an institution according to ascriptions provided by constitutive rules [136, 163]. To give an example in a multi–level governance context, a higher–level institution governing a marriage institution could state that "it is obliged that newly weds have the power to annul their marriage". From the perspective of this dissertation, an interesting aspect is determining from a relatively concrete lower–level institution (such as a marriage institution) whether concrete rules and regulations constitute an abstract power such as newly weds having the legal power to annul their marriage. Moreover, concepts related to power should also be accounted for. Namely, disability (the opposite of power), liability (you are only legally liable if you have legal power) and immunity (without liability you are immune from not fulfilling norms you have no legal power to do so).

One possibility to incorporate all of these concepts is to build on the logics of Hohfeldian rights and liberties [137–140], and institutional power [136]. That is, we could try to combine these formalisations of Hohfeldian legal concepts with multi–level governance formalisation in order to govern instances of Hohfeldian concepts.

PUTTING THE ASCRIPTION OF LEGAL RULE CHANGES INTO PRACTICE

The formalisation in Chapter 6 provides a fixed point characterisation for ascribing legal and valid rule change. The fixed point characterisation provides the mechanism to find models, which can be implemented in any sufficiently expressive language. Yet, there remain practical concerns for the formalism to be usable in a real–world setting.

An obvious improvement is to address the lack of implementation and complexity analysis. Many of the constructs in the formalism are similar to Event Calculus constructs [150] (e.g. a fluent holds if it is initiated until it is terminated). An axiomatisation for an Event Calculus (version) exists in ASP [141]. Hence, one way to implement the formalism would be to provide an Event Calculus–like axiomatisation in ASP. A benefit of this approach is that there are already existing complexity results for ASP [66, 74] which can be used to assess the complexity of an axiomatisation.

Another problem that needs addressing is the assumption that, in the formal framework, all rules which can and/or will be added/removed from an institution are fully characterised by an institution's event generation and state consequence functions. It makes more sense in practical terms to allow the functions describing rules or a database of rules to be updateable with new rules. Then, with new rules added to the database, rule enactment actions can be undertaken by agents to ensure the rules become a valid and recognised part of the institution. If an ASP approach is taken to implementing the formalism, then a logic program would need to be updateable with new rules during run–time. One possibility for updating a rule database is to build on recent advances in answer–set programming for dynamic domains [90].

LEGAL INTERPRETATION

In this dissertation we looked at the legality of institution designs and rule change but we did not investigate legal interpretation, which is closely related to the topics we did investigate as we will discuss. Legal interpretation is the act of courts deciding what various concepts mean, such as to determine the applicability of a rule or whether actions of an agent constitute legally obliged behaviour, even if the written law does not make explicit recommendations for such cases. Legal interpretation is important, since it allows legal norms to be adapted to unforeseen circumstances and changes to the understanding of various concepts by society not represented in the written law. For example, a categorical prohibition on storing private data might have previously meant preventing the storage of communications content between people, but as time went on it became clear that storing communications meta–data is also an infringement on privacy – hence a court would then update the notion of private data to include meta-data.

As Boella et al. show [23, 24] legal interpretation is, formally, a case of rule change. Specifically, whilst legislators have the legal power to change ontological and regulative rules at "design–time", courts have the legal power to change only ontological rules at "run–time" (i.e. as cases are brought before the court). Boella et al. provide a formal mechanism for re–interpreting the ontological meaning of literals in the antecedent of regulations (the applicability of the rule) and the consequent (the obliged actions or state of affairs) based on interpreting the original *goal* of the legislator (i.e. the intention behind the law). Hence, legal interpretation is an important aspect of rule change that ensures laws, at run-time, can be correctly applied and understood according to the goals behind enacting those laws at design–time.

Relating to these aspects, in this dissertation we contributed formalisms that can be used to reason about the *effects* of legal interpretation. In Chapter 3 and Chapter 4 we looked at interpreting the abstract meaning of detached deontological statements based on ontological interpretations provided by courts. But, the ontological interpretations were formalised *a–priori* as counts–as rules (e.g. meta–data count-as personal data) in the case study on EU multi-level governance.

We did, on the other hand, investigate the generation of hypotheses, using ILP, for explaining and rectifying non–compliant institution designs in Chapter 5. Effectively, ILP synthesised changes to ontological and deontological rules in order to ensure the regulatory effects of an institution do not violate the regulations of other institutions. Courts synthesise interpretations, which are effectively changes to the existing set of ontological rules. But, suitable legal interpretation ensures the law is applied and interpreted correctly with respect to its intention, which is different from our purpose of resolving non-compliant institution designs.

Later, we investigated rule change legality and its effects on how an institution evolves in the past/present/future in Chapter 6. Since legal interpretation is a case of rule change, the formalisation of when rule changes are legal could be used, with some enhancements, to reason about when legal interpretations are legal. For example, if we want to reason about a statement such as "a court can introduce a new counts-as rule so long as it is not regulatory and by introducing that rule the goal of the law being interpreted is promoted". By combining the work on interpreting the abstract meaning of regulations in Chapter 3 and Chapter 4 and the work on legal rule change in Chapter 6 we can potentially capture the effects of legally-valid legal interpretation, which is the addition of ontological rules ascribed by rule change regulating counts-as rules, on the abstract meanings of regulatory effects. Our work is related to reasoning about legal interpretation in terms of its effects and its legality when viewed as a type of rule change.

However, this dissertation did not investigate how legal interpretations could be generated in order to support the original goal of a law when it is unsupported according to the run–time interpretation of legal concepts. Potentially, existing approaches by Boella et al. [23, 24], which use defeasible logic to interpret the law, could be adopted to provide a more complete system for reasoning about institutions, their interpretation and governance.

8.3.3. New Research Lines

Social Commitments

Whilst this dissertation looks at prescriptions from the perspective of when institutional rule enactment and institution designs are legal, a similar type of prescription is social commitments. Social commitments, such as promises, emerge between agents. Social commitments establish relationships between parties and counter–parties and expectations between agents. In a social commitment, when one agent, called the debtor, commits to bring about X for the benefit of another agent, called the creditor, then the creditor has the *expectation* that X will be brought about for their benefit and the debtor knows that the creditor has such an expectation [36, 37]. The creditor, has flexibility and autonomy to bring about X in any way they choose including through delegation to another agent. To give an example from [243], a social protocol states whereupon a potential buyer requests a quote for electronic goods, the potential seller is committed to providing the quote and the buyer *expects* to receive a quote.

Social commitments should be formalised since they provide a powerful and ubiquitous means for agents to cooperate in a flexible way on the basis of mutual understanding, which supports social interaction planning whilst preserving participants' autonomy. Furthermore, social commitments allow social endeavours' progress to be assessed with in–built key performance indicators (KPIs), such as 'fulfilled' and 'violated'. Hence, social commitments have seen formal work in the normative reasoning sphere for creating new social protocols from existing commitment–based protocols [16, 94], planning the fulfilment of commitments on the basis of expectations and mutually held beliefs [243], addressing conflicting commitments [2], and converging on a mutual understanding of commitments and expectations [39], to name a few.

In this dissertation we looked at how to synthesise institution changes to rectify non–compliance and how to determine when rule changes are valid, such as to support planning valid legal rule changes. As we saw in Chapter 6, creating and altering rules makes it (im)possible to create and alter rules. This is just as true for social commitments. To give one example, in order to plan a larger project a project manager needs to get various project participants to commit to giving estimated work hours required for deliverables. When participants fulfil commitments, a project manager is able to plan further commitments. Or, to give another example, by creating a social commitment for the trading of goods and services it is possible for each participant to assess the trust they should place in the counter–party after the commitment is fulfilled or violated. Consequently, creating and fulfilling/violating social commitments makes it possible to determine whether further commitments should be created on the basis of new information or increased trust. However, whilst commitment creation and fulfilment is entwined with further commitment creation and planning, planning the synthesis of commitments has not been addressed taking into account this interdependence as far as we can see.

This dissertation offers ideas which can be built on to create such plans. Chapter 6's formalism shows that rule creation (e.g. commitments) is non–trivial given pre–existing rules which ascribe rule changes, where rule changes enable/disable further rule changes. An ASP axiomatisation of Chapter 6's formalism as outlined previously would potentially support planning rule changes. ILP applied to rule synthesis for compliant institution designs contributed by Chapter 5 can potentially be used as a basis for synthesising the right commitments for a given problem. But, the difficulty is determining how to synthesise commitments given that setting up expectations and/or agents fulfilling those expectations is necessary to build trust and meet sub–goals for further commitments to be planned for meeting an overarching social goal.

FORMALISING THE SUBSIDIARITY PRINCIPLE IN MULTI–LEVEL GOVERNANCE

This dissertation looked at formalising compliance in multi–level governance. Subsidiarity, the principle that what can be legislated at the local level should be left up to the local level, is also an aspect of multi–level governance. This dissertation contributes reasoning for abstract regulations which are used to realise subsidiarity, but this dissertation does not investigate subsidiarity itself.

Yet, subsidiarity is an important principle. Ostrom [185, p.90] argues that regulations must be based on local conditions. Whilst regulations should be coordinated across groups/jurisdictions (e.g. carbon emission targets), Wilson argues [238] implementation should be left to the lowest–level jurisdiction where implementation is effective. By involving local lay knowledge, as opposed to a one–size–fits–all approach, policies are more effective by being enhanced with learning from past policy decisions [181, 193, 242]. In general, a one–size–fits–all approach is argued as resulting in failure [187], which the subsidiarity principle aims to address.

The same arguments in favour of subsidiarity for human society governance are made for the governance and design of sociotechnical systems, such as smart grids. Frey et al. [85] and Pitt et al. [198], propose a design/governance framework for socio–technical systems, where higher–levels provide over–arching aims but, adhering to the principle of subsidiarity, delegate to lower–levels for implementation or further delegation to even lower–levels. For example, Pitt et al. [198] talk about decentralised community energy systems. These are small residential communities comprising residences which both produce energy (e.g. with wind turbines) and consume energy, and supply energy to one another. Institutions regulate the communities to manage the appropriation of common energy resources. Scaling involves composing a system of decentralised community energy sub–systems, each sub–system demanding and supplying energy from one another. In order to govern a system of systems, hierarchical governance is proposed, where sub–system's institutions are coordinated (e.g. to ensure energy is shared between sub–systems) but appealing to the principle of subsidiarity different sub–system institutions comprise different regulations as deemed appropriate for the community being governed (e.g. one operates a market economy and another operates a communal economy). Subsidiarity uses local lay knowledge to govern disparate societies or socio–technical systems towards a common aim.

In Chapter 3 and Chapter 4 we looked at the abstraction aspect of subsidiarity, where abstract regulations leave some institutional design decisions up to the local level. However, subsidiarity itself has not been formalised in terms of determining *what* should be left up to the local level. Nor does subsidiarity appear to have been given a precise measure. The question remains, subsidiarity is the principle that what can be done at the local level should be left up to the local level through delegation, but how far down should the delegation go and how much freedom should be given to lower–levels of institution designers? In particular, what level of subsidiarity is suitable taking into account a cross–institutional objective, and lower–level institution designer's capabilities and local lay knowledge?

Arguably, the lack of formalisation is problematic since subsidiarity is a structural optimisation principle and therefore if not measured multi–level governance is not optimisable. We can conceive that if subsidiarity is not realised through over–delegation and overly–abstract regulations at higher–levels, then designers of lower–level institutions have too much room to design compliant institutions. Therefore, coordination between individual sub–systems/societies to meet a global aim is not guaranteed. Moreover, if subsidiarity is not realised due to too little delegation and too concrete regulations in higher–level institutions then lower–level institution designers have too little room to optimise regulations for local lay knowledge to meet global aims. Consequently, subsidiarity needs to be formalised, according to these assumed problems, in order to promote measurement, optimisation, compliance and therefore meeting global goals, and automation.

There are many aspects to address when formalising subsidiarity. A metric should be defined for the *abstractness* level a particular institution possesses at a certain governance level. In Chapter 3 and Chapter 4 we saw how abstract regulations can be complied with but did not define how abstract each level is in a qualitative or quantitative sense. Abstraction in this context is viewed as providing many ways with which to comply, the shortcoming of this measure is that compliance and therefore 'abstractness' is highly dependent on the ontology used. Consequently, a novel measure of abstractness is required that does not depend on the number of ways abstract concepts are ascribed.

A metric should be defined to measure how disparate lower–level systems are and whether they should be viewed as separate systems in the first place, or whether there is a 'one–size–fits–all' institution for a particular goal and multiple jurisdictions. That is, we need a system disparity metric to calculate the level of abstractness each regulation and each governance level should be issued at. There already exist metrics for differences (inconsistencies) between propositions [151], knowledge bases [104] and separate agents' beliefs [127]. But it is unclear whether techniques used for such metrics can be extended to assessing system disparity.

Finally, it should be possible to determine what capability local levels have to regulate more appropriately than higher–levels using local lay knowledge for a particular objective. One possibility is to define reasoning about whether there is appropriate local–level knowledge such that lower–levels are more capable than higher–levels in regulating towards global aims. Such reasoning should be from the perspective of higher–levels that do not necessarily know exactly *what* the local knowledge is. However, how to extract information for whether local–level institution designers have appropriate knowledge is unclear. Abstractness, system disparity and local capabilities to self–govern appropriately should be measurable and combined to formalise a subsidiarity metric for a given objective, yet as we have speculated, each measurement is non–trivial to define.

8.4. CLOSING REMARKS

We finish with a few closing remarks regarding the aspects of governing governance this dissertation focuses on. This dissertation contributes formal and practical reasoning for governing institutional design and change with the PARAGon framework. The contribution is split into formalising institution design and enactment governance. Chapter 3, Chapter 4 and Chapter 5 contributed reasoning for how institutions *should* be designed. Chapter 6 contributed reasoning for how institutional change *can* be enacted. This raises the question, does institutional design and enactment governance involve deontic shoulds or alethic cans?

From the *should* perspective, higher–level institutional rules prescribe the design of lower–level institutions in multi–level governance and are formalised as soft constraints in Chapters 3, Chapter 4 and Chapter 5. In our case, higher–level institutions comprise primary rules that abstractly prescribe lower–level institutional designs. Such primary rules preserve the autonomy of lower–level institution designers since they can be violated by designing a non–compliant institution.

From the *can* perspective, institutional change enactment is governed by Hart's secondary legal rules [117, p.81]. These secondary rules define how institutional rules are created and changed, formalised as hard *institutional* constraints, making (im)possible rule changes in an institution, in Chapter 6. Specifically, secondary rules were formalised as Searle's constitutive rules. As discussed in the background section, constitutive rules *constitute* an institution. According to our conceptualisation an institution's secondary rules *constitute* the ways in which the institutional rule changes can be enacted. The formalism in Chapter 6 ensures only valid rule changes recognised by the institution can be made representing the *actual* state of an institution's rules according to the its own secondary rules.

Parallels can be drawn between institutions governing social behaviour, such as formalised in the literature, and the notions of institutions that govern governance defined by this dissertation's formalisations. In the institution of chess, according to Ricciardi [206], the rules constitute the game. Moves can be physically made outside of chess' rules, but then the game of chess is not being played. In other words, you cannot cheat at chess. An institution comprising prescriptions (ought statements), such as the law, may be considered as 'cheatable' if cheating is defined as the act of violating those prescriptions. An instance of an institution formalised in this dissertation, focussing on institutions involved in governing governance, comprises rules that *constitute* that institution. Institution designers governed by instances of our institutions can 'cheat' with respect to the institutions governing them in one sense and not in another. In our formalisation, you can cheat at institution design in the sense of designing a non–compliant institution (Chapter 3, Chapter 4 and 5), but you cannot cheat at institution design in the sense of enacting rule changes invalidly (Chapter 6). This dissertation formalises both of these kinds of governance. This dissertation argues that it has formalised institutional governing of governance, where regulations regulate how institutions *should* be designed and how institutional change enactments *can* be made.

9

APPENDICES

Appendices are available online at http://www.thomascking.com/dissertation/ThomasKingPhDAppendices.pdf (accessed at time of writing on 14-09-2016).

REFERENCES

[1] Padmore v IRC (1987) STC 36 affirmed by the Court of Appeal (1989) STC 493.

[2] Ajmeri, N., Jiang, J., Chirkova, R., Doyle, J., and Singh, M. P. (2016). Coco: Runtime Reasoning About Conflicting Commitments. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1 – 7.

[3] Alchourrón, C., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of symbolic logic*, 50(2):510–530.

[4] Aldewereld, H., Álvarez-Napagao, S., Dignum, F., and Vázquez-Salceda, J. (2010). Making Norms Concrete. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 807–814.

[5] Aldewereld, H. and Dignum, V. (2010). OperettA: Organization-oriented development environment. In *International Workshop on Languages, Methodologies and Development Tools for Multi-Agent Systems (LADS'010)*, pages 1 – 18.

[6] Alechina, N., Dastani, M., and Logan, B. (2012). Programming Norm-Aware Agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1057–1064. International Foundation for Autonomous Agents and Multiagent Systems.

[7] Alechina, N., Dastani, M., and Logan, B. (2013). Reasoning about normative update. In *Proceedings of the Twenty Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 20–26. AAAI Press.

[8] Ancona, D. and Mascardi, V. (2003). Coo-BDI: Extending the BDI Model with Cooperativity. In *International Workshop on Declarative Agent Languages and Technologies*, pages 109–134. Springer Berlin Heidelberg.

[9] Anderson, A. R. (1958). A reduction of deontic logic to alethic modal logic. *Mind*, 67(265):100–103.

[10] Anderson, A. R. and Moore, O. K. (1957). The Formal Analysis of Normative Concepts. *American Sociological Review*, 22(1):9–17.

[11] Anderson, J. E. (1994). *Public Policymaking: an introduction*. Cengage Learning.

[12] Andrighetto, G., Governatori, G., Noriega, P., and van der Torre, L. (2013). *Normative Multi-Agent Systems*, volume 4. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[13] Antoniou, G., Billington, D., Governatori, G., and Maher, M. (2001). Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2(2):255–287.

[14] Austin, J. (1832). *The Province of Jurisprudence Determined*. John Murray, London.

[15] Axelrod, R. (1986). An Evolutionary Approach to Norms. *The American Political Science Review*, 80(4):1095–1111.

[16] Baldoni, M., Baroglio, C., Chopra, A. K., and Singh, M. P. (2015). Composing and Verifying Commitment-Based Multiagent Protocols. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 10–17. AAAI Press.

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

[18] Benzmüller, C. and Woltzenlogel Paleo, B. (2014). Automating Gödel's ontological proof of god's existence with higher-order automated theorem provers. In *Proceedings of the 2014 European Conference on Artificial Intelligence (ECAI 2014), Frontiers in Artificial Intelligence and Applications 263*, pages 93–98. IOS Press.

[19] Bex, F. J., Van Koppen, P. J., Prakken, H., and Verheij, B. (2010). A hybrid formal theory of arguments, stories and criminal evidence. *Artificial Intelligence and Law*, 18(2):123–152.

[20] Biagioli, C. (1997). Towards a legal rules functional micro-ontology. In *Proceedings of the 1st LegOnt Workshop on Legal Ontologies*.

[21] Bienvenu, M., Fritz, C., and McIlraith, S. (2006). Planning with Qualitative Temporal Preferences. *Proceedings of the Tenth International Conference on the Principles of Knowledge Representation and Reasoning*, pages 134–144.

[22] Boella, G., Broersen, J., and van der Torre, L. (2008). Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In *Intelligent Agents and Multi-Agent Systems, 11th Pacific Rim International Conference on Multi-Agents*, pages 86–97.

[23] Boella, G., Governatori, G., Rotolo, A., and van der Torre, L. (2010a). A Logical Understanding of Legal Interpretation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference*, pages 563–565.

[24] Boella, G., Governatori, G., Rotolo, A., and Van Der Torre, L. (2010b). Lex minus dixit quam voluit, lex magis dixit quam voluit: A formal study on legal compliance and interpretation. In *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue.*, pages 162–183.

[25] Boella, G., Pigozzi, G., and van der Torre, L. (2009). Normative framework for normative system change. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 169–176. International Foundation for Autonomous Agents and Multiagent Systems.

[26] Boella, G. and van der Torre, L. (2003). Permissions and obligations in hierarchical normative systems. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 109–118.

[27] Boella, G. and van der Torre, L. (2004a). Contracts as Legal Institutions in Organizations of Autonomous Agents. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2 (AAMAS 2004)*, pages 948 – 955. IEEE Computer Society.

[28] Boella, G. and van der Torre, L. (2004b). 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.

[29] Boella, G. and van der Torre, L. (2006). Security policies for sharing knowledge in virtual communities. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(3):439–450.

[30] Börzel, T. A. and Risse, T. (2000). Who is Afraid of a European Federation? How to Constitutionalise a Multi-level Governance System. *What Kind of Constitution for What Kind of Polity? Responses to Joschka Fischer*, pages 45–60.

[31] Bratman, M. E. (1987). *Intention, Plans and Practical Reason*. Harvard University Press.

[32] Breuker, J., Valente, A., and Winkels, R. (1997). Legal ontologies: a functional view. In *Proceedings of the 1st LegOnt Workshop on Legal Ontologies*.

[33] Broersen, J., Dignum, F., Dignum, V., and Meyer, J. (2004). Designing a deontic logic of deadlines. *Deontic Logic in Computer Science*, pages 43–56.

[34] Bulling, N., Dastani, M., and Knobbout, M. (2013). Monitoring norm violations in multi-agent systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2013)*, pages 491–498. International Foundation for Autonomous Agents and Multiagent Systems.

[35] Castañeda, H.-N. (1970). On the semantics of the ought-to-do. *Synthese*, 21:449–468.

[36] Castelfranchi, C. (1995). Commitments: from individual intentions to groups and organizations. *Proceedings of the First International Conference on Multiagent Systems, ICMAS 1995*, pages 41–48.

[37] Castelfranchi, C. (1998). Modeling social actions for AI agents. *Artificial Intelligence*, 103(January 1997):157–182.

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

[39] Chopra, A. K. and Singh, M. P. (2015). Generalized Commitment Alignment. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 453–461.

[40] Chopra, A. K. and Singh, M. P. (2016). Custard: Computing Norm States over Information Stores. In *Proceedings of the 15th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2016)*, pages 1096–1105, Singapore. International Foundation for Autonomous Agents and Multiagent Systems.

[41] Cialdini, R. B., Reno, R. R., and Kallgren, C. A. (1990). A Focus Theory of Normative Conduct: Recycling the Concept of Norms to Reduce Littering in Public Places. *Journal of Personality and Social Psychology*, 58(6):1015–1026.

[42] Cliffe, O. (2007). *Specifying and Analysing Institutions in Multi-Agent Systems Using Answer Set Programming*. PhD thesis, University of Bath.

[43] Cliffe, O., De Vos, M., and Padget, J. (2007). Answer Set Programming for Representing and Reasoning About Virtual Institutions. *Computational Logic in Multi-Agent Systems*, pages 60–79.

[44] Conte, R. and Castelfranchi, C. (1995). *Cognitive and Social Action*. UCL Press.

[45] Corapi, D., Ray, O., Russo, A., Bandara, A., and Lupu, E. (2009). Learning Rules from User Behaviour. *Artificial Intelligence Applications and Innovations III*, 296:459–468.

[46] Corapi, D., Russo, A., De Vos, M., Padget, J., and Satoh, K. (2011). Normative design using inductive learning. *Theory and Practice of Logic Programming*, 4-5(11):783–799.

[47] Corapi, D., Russo, A., and Lupu, E. (2010). Inductive logic Programming as Abductive Search. In *Proceedings of the International Conference on Logic Programming (ICLP)*, pages 54–63.

[48] Council of Europe (1953). European Convention on Human Rights.

[49] Criado, N., Argente, E., Noriega, P., and Botti, V. (2013). Reasoning about constitutive norms in BDI agents. *Logic Journal of the IGPL*, 22(1):66–93.

[50] Criado, N., Black, E., and Luck, M. (2015). A coherence maximisation process for solving normative inconsistencies. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 30(4):640 – 680.

[51] D'Altan, P., Meyer, J.-J. C., and Wieringa, R. J. (1996). An Integrated Framework for Ought-to-Be and Ought-to-Do Constraints. *Artificial Intelligence and Law*, 4:77–111.

[52] Dastani, M. (2008). 2APL: A Practical Agent Programming Language. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 16(3):214–248.

[53] Dastani, M., Grossi, D., Meyer, J.-J. C., and Tinnemeier, N. (2008). Normative multi-agent programs and their logics. In *Proceedings of Knowledge Representation for Agents and Multi-Agent Systems (KRAMAS'08)*, pages 236–243.

[54] Dastani, M., van der Torre, L., and Yorke-Smith, N. (2015). Commitments and interaction norms in organisations. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*.

[55] De Boer, F. S., Hindriks, K. V., Van Der Hoek, W., and Meyer, J.-J. C. (2000). Agent Programming with Declarative Goals. In *Proceedings of the International Workshop on Agent Theories, Architectures, and Languages*, pages 228–243.

[56] Dennett, D. (1987). *The Intentional Stance*. Cambridge, MA: MIT Press.

[57] Diaconescu, A. and Pitt, J. (2014). Holonic Institutions for Multi-scale Polycentric Self-governance. *Coordination, Organizations, Institutions, and Norms in Agent Systems X, Lecture Notes in Computer Science*, 9372:19–35.

[58] Dignum, F. (2002). Abstract Norms and Electronic Institutions. In *International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*, pages 93 – 104.

[59] Dignum, F. and Kuiper, R. (1998). Specifying Deadlines with Dense Time Using Deontic and Temporal Logic. *International Journal of Electronic Commerce*, 3(2):67–86.

[60] Dignum, F., Weigand, H., and Verharen, E. (1996). Meeting the deadline: on the formal specication of temporal deontic constraints. *International Symposium on Methodologies for Intelligent Systems*, pages 243–252.

[61] Dignum, V. (2003). *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, University of Utrecht, Utrecht, The Netherlands.

[62] Dignum, V., Vázquez-Salceda, J., and Dignum, F. (2004). OMNI: Introducing Social Structure , Norms and Ontologies into Agent Organizations. In *Programming Multi-Agent Systems*, pages 181–198. Springer Berlin Heidelberg.

[63] D'Inverno, M., Luck, M., Noriega, P., Rodriguez-Aguilar, J. a., and Sierra, C. (2012). Communicating Open Systems. *Artificial Intelligence*, 186:3146–3150.

[64] Dutta, P., Aoki, P. M., Kumar, N., Mainwaring, A., Myers, C., Willett, W., and Woodruff, A. (2009). Common sense: participatory urban sensing using a network of handheld air quality monitors. In *Proceedings of the 7th ACM conference on embedded networked sensor systems*, pages 349–350.

[65] Eichener, V. (1997). Effective European problem-solving: lessons from the regulation of occupational safety and environmental protection. *Journal of European Public Policy*, 4(4):591–608.

[66] Eiter, T., Faber, W., Fink, M., and Woltran, S. (2007). Complexity results for answer set programming with bounded predicate arities and implications. *Annals of Mathematics and Artificial Intelligence*, 51(2-4):123–165.

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

[68] Emerson, E. (1990). Temporal and modal logic. *Handbook of Theoretical Computer Science*, B:955–1072.

[69] Emerson, E. A. and Halpern, J. Y. (1985). Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24.

[70] European Court Reports (2014). 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.

[71] European Parliament and the Council of the European Union (2006). 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.

[72] European Union (2000). Charter of Fundamental Rights of the European Union 2000/C 364/01. *Official Journal of the European Communities*.

[73] European Union (2008). Consolidated Version of the Treaty on the Functioning of the European Union.

[74] Faber, W. and Leone, N. (2007). On the Complexity of Answer Set Programming with Aggregates. In *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'07)*, pages 97–109, Tempe, Arizona.

[75] Faci, N., Modgil, S., Oren, N., Meneguzzi, F., Miles, S., and Luck, M. (2008). Towards a monitoring framework for agent-based contract systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5180 LNAI, pages 292–305.

[76] Folsom, R., Lake, R. B., and Nanda., V. P. (1996). *European Union Law After Maastricht: Practical Guide for Lawyers Outside the Common Market*. Kluwer law international.

[77] Fornara, N. (2011). Specifying and monitoring obligations in open multiagent systems using semantic web technology. *Semantic Agent Systems*, pages 25–45.

[78] Fornara, N. and Colombetti, M. (2009). Specifying artificial institutions in the event calculus. *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models, Information science reference*, pages 335–366.

[79] Fornara, N. and Colombetti, M. (2010). Representation and monitoring of commitments and norms using OWL. *AI Communications*, 23(4):341–356.

[80] Fornara, N., Okouya, D., and Colombetti, M. (2012). Using OWL 2 DL for expressing ACL Content and Semantics. In *European Workshop on Multi-Agent Systems*, pages 97–113.

[81] Fornara, N., Viganò, F., and Colombetti, M. (2007). Agent communication and artificial institutions. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 14(2):121–142.

[82] Fornara, N., Viganò, F., Verdicchio, M., and Colombetti, M. (2008). Artificial institutions: A model of institutional reality for open multiagent systems. *Artificial Intelligence and Law*, 16(1):89–105.

[83] Forrester, J. W. (1984). Gentle Murder, or the Adverbial Samaritan. *The Journal of Philosophy*, 81(4):193–197.

[84] Frankfurt, H. G. (1971). Freedom of the Will and the Concept of a Person. *The Journal of Philosophy*, 68(1):5–20.

[85] Frey, S., Diaconescu, A., Menga, D., and Demeure, I. (2013). A holonic control architecture for a heterogeneous multi-objective Smart Micro-Grid. *7th IEEE international conference on self-adaptive and self- organizing systems, SASO 2013*, pages 21–30.

[86] Gabbay, D., Horty, J., Parent, X., van der Meyden, R., and van der Torre, L., editors (2013). *Handbook of Deontic Logic and Normative Systems vol. 1*.

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

[88] Gardner, J. (2001). Legal Positivism: 5 1/2 Myths. *American Journal of Jurisprudence*, 46(1):199–228.

[89] Gauci, M., Chen, J., Li, W., Dodd, T. J., and Gross, R. (2014). Clustering objects with robots that do not compute. In *Proceedings of the 2014 International Conference on Autonomous agents and multi-agent systems (AAMAS 2014)*, pages 421–428. International Foundation for Autonomous Agents and Multiagent Systems.

[90] Gebser, M., Janhunen, T., Jost, H., Kaminski, R., and Schaub, T. (2015). ASP solving for expanding universes. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 354–367, Lexington, USA. Springer International Publishing.

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

[92] Gelfond, M. (2008). Answer Sets. *Foundations of Artificial Intelligence*, 3:285–316.

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

[94] Gerard, S. N., Telang, P. R., Kalia, A. K., and Singh, M. P. (2015). Positron: Composing Commitment-Based Protocols. In *Proceedings of the 13th IEEE International Conference on Service-Oriented Computing (ICSOC)*, pages 19 – 35.

[95] Ginet, C. (1990). *On Action*. Cambridge: Cambridge University Press.

[96] Goldman, A. (1970). *A Theory of Human Action*. Englewood Cliffs, NJ: Prentice-Hall.

[97] Governatori, G. (2005). Representing business contracts in RuleML. *International Journal of Cooperative Information Systems*, 14(2-3):181–216.

[98] Governatori, G., Hulstijn, J., and Rotolo, A. (2007). Characterising deadlines in Temporal Modal Defeasible Logic. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence*, pages 486–496.

[99] Governatori, G., Olivieri, F., Rotolo, A., and Scannapieco, S. (2013). Computing strong and weak permissions in defeasible logic. *Journal of Philosophical Logic*, 42(6):799–829.

[100] Governatori, G., Palmirani, M., Riveret, R., Rotolo, A., and Sartor, G. (2005a). Norm modifications in defeasible logic. In *Legal Knowledge and Information Systems (JURIX 2005)*, pages 13–22. IOS Press.

[101] Governatori, G. and Rotolo, A. (2009). How Do Agents Comply with Norms? In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 488–491. IEEE Computer Society.

[102] Governatori, G. and Rotolo, A. (2010). Changing Legal Systems: Legal Abrogations and Annulments in Defeasible Logic. *Logic Journal of IGPL*, 18:157–194.

[103] Governatori, G., Rotolo, A., and Sartor, G. (2005b). 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. ACM Press.

[104] Grant, J. and Hunter, A. (2016). Analysing inconsistent information using distance-based measures. *International Journal of Approximate Reasoning*.

[105] Grossi, D. (2008). Pushing Anderson's Envelope: The Modal Logic of Ascription. In *9th International Conference on Deontic Logic in Computer Science (DEON 2008)*, pages 263–277.

[106] Grossi, D. (2011). Norms as ascriptions of violations: An analysis in modal logic. *Journal of Applied Logic*, 9(2):95–112.

[107] Grossi, D., Aldewereld, H., and Dignum, F. (2007). Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In *Coordination, organizations, institutions, and norms in agent systems II*, pages 101 – 114. Springer Berlin Heidelberg.

[108] Grossi, D., Aldewereld, H., Vázquez-Salceda, J., and Dignum, F. (2006a). Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational and Mathematical Organization Theory*, 12:251–275.

[109] Grossi, D. and Dignum, F. (2005). From abstract to concrete norms in agent institutions. In *Formal Approaches to Agent-Based Systems*, pages 12–29.

[110] Grossi, D., Meyer, J.-J., and Dignum, F. (2005). 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.

[111] Grossi, D., Meyer, J.-J. C., and Dignum, F. (2006b). 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.

[112] Grossi, D., Meyer, J. J. C., and Dignum, F. (2008). The many faces of counts-as: A formal analysis of constitutive rules. *Journal of Applied Logic*, 6(2):192–217.

[113] Günay, A. and Yolum, P. (2012). Detecting conflicts in commitments. *Declarative Agent Languages and Technologies IX*, pages 51–66.

[114] Hansson, S. O. (2013). Formal Philosophy - A Guarded Defence. *Theoria*, 79(4):287–289.

[115] Hardin, G. (1968). The Tragedy of the Commons. *Science*, 162(3859):1243–1248.

[116] Harel, D., Tiuryn, J., and Kozen, D. (1984). Dynamic logic. In Gabbay, D. and Guenthner, F., editors, *Handbook of Philosophical Logic*, volume 2, chapter 10, pages 497–604. Reidel, Dordrecht.

[117] Hart, H. L. A. (1961). *The Concept of Law*. Clarendon Press., Oxford.

[118] Hewitt, C. E. (1986). Offices Are Open Systems. *ACM Transactions of Office Automation Systems*, 4(3):271–287.

[119] Hindriks, K. V., de Boer, F. S., van der Hoek, W., and Meyer, J.-J. C. (2000). *A Programming Logic for Part of the Agent Language 3APL*, volume 78-89. Springer Berlin Heidelberg.

[120] Hohfeld, W. N. (1913). Some Fundamental Legal Conceptions as Applied in Judicial Reasoning. *The Yale Law Journal*, 23(1):16 – 59.

[121] Hooghe, L. and Marks, G. (2003). Unraveling the central state, but how? Types of multi-level governance. *American political science review*, 97(2):233–243.

[122] Horty, J. F. and Belnap, N. (1995). The deliberative stit: A study of action, omission, ability, and obligation. *Journal of Philosophical Logic*, 24(6):583–644.

[123] Hübner, J., Sichman, J., and Boissier, O. (2004). Using the Moise+ for a Cooperative Framework of MAS Reorganisation. In *17th Brazilian Symposium on Artificial Intelligence (SBIA'04)*, pages 506–515, Berlin.

[124] Hübner, J., Sichman, J., and Boissier, O. (2007). Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3):370–395.

[125] Hübner, J. F., Sichman, J. S., and Boissier, O. (2005). S-Moise+ : A Middleware for developing Organised Multi-Agent Systems. *Coordination, Organization, Institutions and Norms I. LNAI.*, 3913:64—-78.

[126] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., and Madden, S. (2006). Cartel: a distributed mobile sensor computing system. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 125–138. ACM.

[127] Hunter, A., Parsons, S., and Wooldridge, M. (2014). Measuring Inconsistency in Multi-Agent Systems. *KI - Künstliche Intelligenz*, 28(3):169–178.

[128] Jensen, A. S., Dignum, V., and Villadsen, J. (2014). The AORTA architecture: Integrating organizational reasoning in Jason. In *International Workshop on Engineering Multi-Agent Systems*, pages 112–128. Springer International Publishing.

[129] Jensen, A. S., Dignum, V., and Villadsen, J. (2015). A framework for organization-aware agents. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS 2016)*, pages 1133 – 1134. Springer US.

[130] Jiang, J. (2015). *Organizational Compliance: An Agent-based Model for Designing and Evaluating Organizational Interactions*. PhD thesis, TU Delft, Delft University of Technology.

[131] Jiang, J., Aldewereld, H., Dignum, V., and Tan, Y.-H. (2012). Norm Contextualization. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems (COIN 2012)*, pages 141 – 157. Springer Berlin Heidelberg.

[132] Jiang, J., Dignum, V., Tan, Y., and Overbeek, S. (2011). A Context-Aware Inter-organizational Collaboration Model Applied to International Trade. In *International Conference on Electronic Government*, pages 308–319. Springer Berlin Heidelberg.

[133] Jiang, J., Pitt, J., and Diaconescu, A. (2015). Rule Conflicts in Holonic Institutions. In *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 49–54. IEEE.

[134] Jones, A. J. I. and Sergot, M. (1992). Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*, 1(1):45–64.

[135] Jones, A. J. I. and Sergot, M. (1993). On the Characterisation of Law and Computer Systems: The Normative Systems Perspective. In *Deontic Logic in Computer Science: Normative System Specification*, pages 275–307. John Wiley & Sons.

[136] Jones, A. J. I. and Sergot, M. (1996). A Formal Characterisation of Institutionalised Power. *Journal of IGPL*, 4(3):427–443.

[137] Kanger, S. (1971). New Foundations for Ethical Theory. *Deontic Logic: Introductory and Systematic Readings*, pages 36 – 58.

[138] Kanger, S. (1972). Law and Logic. *Theoria*, 38:105 – 132.

[139] Kanger, S. (1985). On Realization of Human Rights Kanger. *Action, Logic and Social Theory*, 38.

[140] Kanger, S. and Kanger, H. (1966). Rights and Parliamentarianism. *Theoria*, 32:85–115.

[141] Kim, T.-w., Lee, J., and Palla, R. (2009). Circumscriptive Event Calculus as Answer Set Programming. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 823 – 829.

[142] King, T. C. (2015). Governing Government: A Framework for Multi-tier Institution Specification and Reasoning (Doctoral Consortium). In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 2011 – 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[143] King, T. C., Dignum, V., and Jonker, C. M. (2016a). 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. IOS Press.

[144] King, T. C., Li, T., De Vos, M., Dignum, V., Jonker, C. M., Padget, J., and Riemsdijk, M. B. V. (2015a). 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. International Foundation for Autonomous Agents and Multiagent Systems.

[145] King, T. C., Li, T., Vos, M. D., Jonker, C. M., Padget, J., and Riemsdijk, M. B. V. (2016b). Revising Institutions Governed by Institutions for Compliant Regulations. *Coordination, Organizations, Institutions, and Normes in Agent Systems XI: COIN 2015 International Workshops, COIN@ AAMAS, Istanbul, Turkey, May 4, 2015, COIN@ IJCAI, Buenos Aires, Argentina, July 26, 2015, Revised Selected Papers.*, 9628:191 – 208.

[146] King, T. C., Liu, Q., Polevoy, G., de Weerdt, M., Dignum, V., van Riemsdijk, M. B., and Warnier, M. (2014). Request Driven Social Sensing (Demonstration). In Lomuscio, A., Scerri, P., Bazzan, A., and Huhns, M., editors, *Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, pages 1651 – 1652, Paris, France. International Foundation for Autonomous Agents and Multiagent Systems.

[147] King, T. C., Riemsdijk, M. B. V., Dignum, V., and Jonker, C. M. (2015b). 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.

[148] Knobbout, M., Dastani, M., and Meyer, J.-j. C. (2014). Reasoning about Dynamic Normative Systems. In *European Workshop on Logics in Artificial Intelligence*, pages 628–636.

[149] Kollingbaum, M. J., Vasconcelos, W. W., García-Camino, A., and Norman, T. J. (2007). Managing conflict resolution in norm-regulated environments. In *International*

Workshop on Engineering Societies in the Agents World, pages 55 – 71. Springer Berlin Heidelberg.

[150] Kowalski, R. and Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing*, 4(1):67–95.

[151] Lafage, C. and Lang, J. (2001). Propositional Distances and Preference Representation. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'01*, pages 48–59.

[152] Lee, J., Padget, J., Logan, B., Dybalova, D., and Alechina, N. (2014). N-Jason: Runtime norm compliance in AgentSpeak(L). *Engineering Multi-Agent Systems*, 8758:367–387.

[153] Leibniz, G. W. (1685). *The Art of Discovery*.

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

[155] Li, T., Balke, T., De Vos, M., Padget, J., and Satoh, K. (2013a). 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*.

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

[157] Li, T., Vos, M. D., Padget, J., Satoh, K., and Balke, T. (2015). Debugging ASP using ILP. In *Technical Communcations of International Conference of Logic Programming 2015*.

[158] Liu, Q., Dulman, S., and Warnier, M. (2013). AREA: an Automatic Runtime Evolutionary Adaptation mechanism for Creating Self-Adaptation Algorithms in Wireless Networks. In *Spatial Computing, the 6th International Workshop, colocated with AAMAS 2013*, pages 23 – 28, Saint Paul, MN.

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

[160] López, F. L. y., Luck, M., and D'Inverno, M. (2006). A normative framework for agent-based systems. *Computational and Mathematical Organization Theory*, 12(2-3):227–250.

[161] Lowe, E. (2008). *Personal Agency: The Metaphysics of Mind and Action*. Oxford: Oxford University Press.

[162] Lu, H., Pan, W., Lane, N., Choudhury, T., and Campbell, A. (2009). SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 165–178.

[163] Makinson, D. (1986). On the Formal Representation of Rights Relations. *Journal of Philosophical Logic*, 15:403–425.

[164] Makinson, D. (1999). On a Fundamental Problem of Deontic Logic. *Norms, Logics and Information Systems. New Studies in Deontic Logic and Computer Science*, pages 29–53.

[165] Makinson, D. and van der Torre, L. (2003). What is input/output logic? *Trends in Logic*, 17:163–174.

[166] Marks, G. (1993). Structural Policy and Multilevel Governance in the European Community.

[167] Max Knobbout (2016). *Logics for Modelling and Verifying Normative Multi-Agent Systems*. PhD thesis, Utrecht University.

[168] McNamara, P. (2014). Deontic Logic. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab Center for the Study of Language and Information, Stanford University Stanford, CA 94305-4115, winter 201 edition.

[169] Mele, A. (2003). *Motivation and Agency*. Oxford: Oxford University Press.

[170] Meneguzzi, F. and Luck, M. (2009). Norm-based behaviour modification in BDI agents. In *Proceedings of the eighth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2009)*, pages 177–184.

[171] Meneguzzi, F., Modgil, S., Oren, N., and Miles, S. (2012). Applying electronic contracting to the aerospace aftercare domain. *Engineering Applications of Artificial Intelligence*, 25(7):1471–1487.

[172] Merriam-Webster Online. Compliance - www.merriam-webster.com. Accessed on 2016-06-25.

[173] Miller, S. (2014). Social Institutions. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab Center for the Study of Language and Information, Stanford University Stanford, CA 94305-4115, winter 201 edition.

[174] Mitchell., R. B. (1996). Compliance theory: an overview. *Improving compliance with international environmental law*, pages 3–28.

[175] Modgil, S., Faci, N., and Meneguzzi, F. (2009). A framework for monitoring agent-based normative systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1 (AAMAS 2009)*, pages 153–160. International Foundation for Autonomous Agents and Multiagent Systems.

[176] Morales, J., López-Sánchez, M., and Esteva, M. (2011). Using experience to generate new regulations. In *IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, number 2011, pages 307–312. ACM.

[177] Morales, J., Lopez-Sanchez, M., Rodriguez-Aguilar, J. A., Wooldridge, M., and Vasconcelos, W. (2013). Automated synthesis of normative systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2013)*, pages 483–490.

[178] Morales, J., Lopez-Sanchez, M., Rodriguez-Aguilar, J. A., Wooldridge, M., and Vasconcelos, W. (2014). Minimality and Simplicity in the On-line Automated Synthesis of Normative Systems. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2014)*, pages 109–116.

[179] Moretti, A. (2009). *The Geometry of Logical Opposition*. PhD thesis, University of Neuchâtel.

[180] Muggleton, S. (1995). Inverse entailment and Progol. *New generation computing*, 13(3-4):245–286.

[181] Newig, J. and Fritsch, O. (2009). Environmental governance: Participatory, multi-level - and effective? *Environmental policy and governance*, 19(3):197 – 214.

[182] North, D. C. (1990). *Institutions, institutional change and economic performance*. Cambridge university press.

[183] Nute, D. (1987). Defeasible logic. *Handbook of Logic in Artificial Intelligence and Logic Programming*, 3.

[184] O'Connor, T. (2000). *Persons and Causes: The Metaphysics of Free Will*. Oxford: Oxford University Press.

[185] Ostrom, E. (1990). *Governing the commons: The evolution of institutions for collective action*. Cambridge University Press, Cambridge, UK.

[186] Ostrom, E. (2005). *Understanding institutional diversity*. Princeton University Press.

[187] Ostrom, E. (2009). A general framework for analyzing sustainability of social-ecological systems. *Science (New York, N.Y.)*, 325(5939):419–22.

[188] Ostrom, E. (2011). Thinking About Climate Change as a Commons. In *15th Annual Philip Gamble Memorial Lecture*, pages 1–34, UMass Amherst.

[189] Overeem, A., Leijnse, H., and Uijlenhoet, R. (2013a). Country-wide rainfall maps from cellular communication networks. *Proceedings of the National Academy of Sciences of the United States of America*, 110(8):2741–2745.

[190] Overeem, A., R. Robinson, J. C., Leijnse, H., Steeneveld, G. J., P. Horn, B. K., and Uijlenhoet, R. (2013b). Crowdsourcing urban air temperatures from smartphone battery temperatures. *Geophysical Research Letters*, 40(15):4081–4085.

[191] Oxford Dictionaries Online. Paragon - www.oxforddictionaries.com. Accessed on 2016-04-28.

[192] Parliament of the United Kingdom (2008). Finance Act 2008, Chapter 9.

[193] Pellizzoni, L. (2003). Uncertainty and participatory democracy. *Environmental Values*, 12(2):195–224.

[194] Pieters, W., Padget, J., and Dechesne, F. (2013). 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.

[195] Pieters, W., Padget, J., Dechesne, F., Dignum, V., and Aldewereld, H. (2015). Effectiveness of qualitative and quantitative security obligations. *Journal of Information Security and Applications*, 22:3 – 16.

[196] Pinker, S., Bizzi, E., Brenner, S., Chomsky, N., Minsky, M., Partee, B. H., and Winston, P. H. (2016). The Golden Age – A Look at the Original Roots of Artificial Intelligence, Cognitive Science, and Neuroscience (partial transcript).

[197] Pitt, J. and Artikis, A. (2015). The open agent society: retrospective and prospective views. *Artificial Intelligence and Law*, 23(3):241 – 270.

[198] Pitt, J. and Diaconescu, A. (2015). Structure and Governance of Communities for the Digital Society. In *IEEE International Conference on Autonomic Computing (ICAC)*, pages 279–284.

[199] Pnueli, A. (1977). The Temporal Logic of Programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57. IEEE.

[200] Prakken, H. and Sartor, G. (1996). A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4:331–368.

[201] Prakken, H. and Sergot, M. (1996). Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115.

[202] Rao, A. S. (1996). AgentSpeak(L): BDI Agents speak out in a logical computable language. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 42 – 55. Springer Berlin Heidelberg.

[203] Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132.

[204] Republic of Italy (1947). Constitution of the Italian Republic.

[205] R.H.Bordini, J.F.Hubner, and M.Wooldridge (2007). *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley & Sons.

[206] Ricciardi, M. (1997). Constitutive rules and institutions. In *meeting of the Irish Philosophical Club, Ballymascanlon*.

[207] Ross, A. Tû-Tû. *Harvard Law Review (1956 - 1957)*, 70:812–825.

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

[209] Ryu, Y. U. and Lee, R. M. (1994). Defeasible deontic reasoning: A logic programming model. In *Deontic logic in computer science*, pages 225–241. John Wiley and Sons Ltd.

[210] Scharpf, F. W. (1997). Introduction: the problem-solving capacity of multi-level governance. *Journal of European Public Policy*, 4(4):520–538.

[211] Schlosser, M. (2015). Agency. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy (Fall 2015 Edition)*. The Metaphysics Research Lab Center for the Study of Language and Information, Stanford University Stanford, CA 94305-4115.

[212] Searle, J. R. (1964). How to Derive "Ought" from "Is". *The Philosophical Review*, 73(1):43–58.

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

[214] Searle, J. R. (1995). *The Construction of Social Reality*. The Free Press, New York.

[215] Searle, J. R. (2005). What is an institution? *Journal of Institutional Economics*, 1:1–22.

[216] Sergot, M. (1988). *Representing legislation as logic programs*. Oxford University Press.

[217] Sergot, M. J. (1982). Prospects for representing the law as logic programs. *Logic Programming*, pages 33–42.

[218] Shams, Z., Vos, M. D., Oren, N., and Padget, J. (2016). Normative Practical Reasoning via Argumentation and Dialogue. In *Proceedings of the 25th International Conference on Artificial Intelligence*. AAAI Press.

[219] Smith, M. E. (2004). Toward a Theory of EU Foreign Policy Making: Multi-level Governance, Domestic Politics, and National Adaptation to Europe's Common Foreign and Security Policy. *Journal of European Public Policy*, 11(4):740–758.

[220] Suber, P. (1990). *The Paradox of Self-Amendment: A Study of Law, Logic, Omnipotence, and Change*. Peter Lang International Academic Publishers.

[221] The United States of America. The United States Constitution.

[222] Tinnemeier, N. A. M., Dastani, M. M., Meyer, J. J. C., and Van Der Torre, L. (2009). Programming normative artifacts with declarative obligations and prohibitions. *Proceedings - 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2009*, 2:145–152.

[223] Tosatto, S. C., Boella, G., van der Torre, L., and Villata, S. (2012a). Abstract Normative Systems: Semantics and Proof Theory. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference*, pages 358–368.

[224] Tosatto, S. C., Boella, G., Van Der Torre, L., and Villata, S. (2012b). Visualizing normative systems: An abstract approach. In *International Conference on Deontic Logic in Computer Science*, pages 16–30. Springer Berlin Heidelberg.

[225] Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematics Society*, 42(2):230–265.

[226] UK (2009). The Data Retention (EC Directive) Regulations 2009, No. 859.

[227] United States Federal Law (1998). Children's Online Privacy Protection Act.

[228] van der Torre, L. (1997). *Reasoning about obligations: defeasibility in preference-based deontic logic*. PhD thesis, Erasmus University Rotterdam.

[229] van der Torre, L. and Tan, Y. (1998). The temporal analysis of Chisholm's paradox. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*, pages 650–655.

[230] van der Torre, L. and Tan, Y. (1999). Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence*, 27 (1-4):49–78.

[231] Vasconcelos, W., Kollingbaum, M. J., and Norman, T. J. (2007). Resolving conflict and inconsistency in norm-regulated virtual organizations. In *Proceedings of the 6th International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2007)*, pages 632–639, New York, New York, USA. IFAAMAS.

[232] Vasconcelos, W. W., Kollingbaum, M. J., and Norman, T. J. (2008). Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 19(2):124–152.

[233] Vazquez-Salceda, J. (2003). *The role of norms and electronic institutions in multi-agent systems applied to complex domains. The HARMONIA framework*. PhD thesis, Technical University of Catalonia.

[234] von Wright, G. (1951). Deontic logic. *Mind*, 60(237):1–15.

[235] von Wright, G. (1968). An essay in deontic logic and the general theory of action. *Acta Philosophica Fennica*, 21.

[236] von Wright, G. H. (1983). Norms of Higher Order. *Studia Logica*, 42(119).

[237] Wansing, H. (1998). Nested deontic modalities: Another view of parking on highways. *Erkenntnis*, 49(2):185–199.

[238] Wilson, D. S., Ostrom, E., and Cox, M. E. (2013). Generalizing the core design principles for the efficacy of groups. *Journal of Economic Behavior and Organization*, 90.

[239] Wittgenstein, L. (1958). *Philosophical Investigations*. Blackwell Publishers.

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

[241] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice.

[242] Yearley, S., Cinderby, S., Forrester, J., Bailey, P., and Rosen, P. (2003). Participatory modelling and the local governance of the politics of UK air pollution: A three-city case study. *Environmental Values*, 12(2):247–262.

[243] Yolum, P. and Singh, M. (2004). 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.

[244] Zuckerman, H. S. (1979). Multi-Institutional Systems: Promise and Performance. *Inquiry*, 16(4):291–314.

ACKNOWLEDGEMENTS

This dissertation came about through periods of frenzy and pauses for thought, apathy and persistence, ignorance and exploration, disorientation and discovery, and disagreement and acquiescence. I cannot say I found the process particularly easy. However, "omelets are not made without breaking eggs". Although not something I completely agree with, the man who coined the phrase (François de Charette) was executed in 1796 for breaking too many eggs, I sympathise with short–term difficulties being necessary for certain ends.

Those end results are not just papers, presentations and propositions, but also personal progress in acquiring more practical perspectives. One such perspective was becoming tolerant of my work being "naïve", such as by not capturing all aspects of a particular concept. Naïve work can help our understanding, as the field of Deontic Logic has shown through decades of conceptual changes and refinements. The Japanese aesthetic of Wabi Sabi encapsulates such a view, where transience and imperfection are sought and not fought. That outlook extends to people, including myself, as much as it applies to technical work. Moreover, it may be that if the response to a difficult position is not fight then it is flight, persistently tolerating issues is important – perhaps this all sounds clichéd, but the grass is always greener on the other side. Acknowledging and eventually feeling that not every problem is to be (re)solved, meant that the final work and paper were a walk in the park.

Getting to that point, it must be emphasised, was not a solitary effort. There are many people that, in ways small and big, shaped my personal and research outlook in constructive ways and helped to ensure that the outcome was overall positive. I would therefore like to segue to thanking some of those people that helped me along the way.

Firstly, I would like to thank, in no particular order, my supervisory team comprising my promotor Prof. dr. Catholijn M. Jonker, and copromotors Dr. Birna van Riemsdijk and Dr. Virginia Dignum for their support, discussion and useful comments. Catholijn, thank you for giving me the opportunity to study at Interactive Intelligence – a research group that is both positively diverse in people and research topics – for having confidence when things were not going well, and for helping me with career advice and the next steps towards the end. Birna, thank you for giving me time to discover my research interests, putting together a diverse and open ended project, emphasising simplicity (but not too simple!) over complexity and helping me through the early stages of learning formal work. Virginia, thank you for helping me in understanding where and when to stop solving a large problem and start focussing on solving a small problem well, ensuring different aspects of my research were more modular than iterative, and especially, thank you for understanding and resolving seemingly insurmountable difficulties. Of course, there is more to thank for, but I feel these aspects were at the core of getting me to the end.

Moreover, I would like to extend my thanks to my mentor, Dr. Klaas Pieter Hart. Although we met infrequently, Klaas listened during times of strife, gave practical advice and, much to my reassurance, validated my concerns.

Special thanks to the examination committee: Prof. dr. Katie Atkinson, Prof. dr.

Frances Brazier, Prof. dr. Jeremy Pitt and Prof. dr. Leon van der Torre. Thank you for reading my dissertation. Moreover, thank you Katie Atkinson for the remarks, they helped with clarification and it is good to know there is agreement on the dissertation's contributions and shortcomings. Thank you Leon van der Torre for the detailed comments, they helped to better position the work and gave useful insights into nascent research topics.

I would also like to thank Dr. Tingting Li, Dr. Julian Padget and Dr. Marina de Vos for the collaborative research and co–authorships. In particular, thank you for being so welcoming when I visited Bath, introducing me to your research on institutions and your overall approach. Our co–authorships greatly influenced my dissertation and research.

I'd also like to thank everyone who was involved in the SHINE multi–disciplinary project under which this research was conducted. Especially, I'd like to thank the SHINE PhD/postdoc researchers – Qingzhi Liu, Dr. Paolo Palmieri, Gleb Polevoy, Venkat Roy and Sergio Velasco – for the enjoyable project meetings and finding ways to work together.

I was between two departments during my PhD. First, I would like to thank all of my wonderful colleagues at the ICT section of TBM and the Interactive Intelligence group of EWI, there are too many to mention so thank you all. Both groups were diverse in research topics and people, so thank you for widening my personal and research perspectives.

I would specifically like to thank my TBM office–mates: Jie Jiang and Janneke van der Zwaan for discussing common problems encountered, being so welcoming and thank you Jie for those early days explaining the area as a whole and what you thought I needed to do to get started. I would also like to thank Virginia's latest PhD students (at the time of writing), Klara Pigmans and Rijk Mercuur for the interesting discussions about their own research and perspectives, which I look forward to continuing. I would also like to thank my II office mates: Alex Kayal for his anecdotes and banter, Marieke Peeters for listening to my cynicism and countering it with research rigour, and Wenxin Wang for conversing about family and showing it can coincide with work. Moreover, I would like to thank the following inspiring individuals for helping with the softer problems encountered: Reyhan Aydogan for her positivity and emphasis on work–life balance, Chris Detweiler for his dry humour and relateability, Maaike Harbers for her outgoing enthusiasm and compassion, Max Knobbout for his insightful knowledge and openness, Arman Noroozian for his academic enthusiasm and rational calmness, and Vanessa Vakili for her grounded outlook and sympathetic listening.

Moreover, I would also like to thank from the II group Fawad Khan for helping with implementation, and Wouter Pasman with helping me to resolve code problems and for always having an alternative take on things. Thank you Anita Hoogmoed and Caroline de Wit for always being willing to help with organisational matters and maintaining a cheerful state. I'd like to thank Ruud de Jong and Bart Vastenhouw for happily assisting in technical matters (and also Ruud for the vast coffee supplies).

My long–time friend Sonal Sodha/Keay was there for me during the difficult times. Moreover, long long before I even embarked on my PhD she gave me a sharp push that helped me get to where I am now. Thank you, Sonal.

Finally, I would like to thank my wonderful life partner, Rūta, for always being there for me, providing both joy and support. Thank you for all of the adventures helping me to get away from work and for being understanding when all I could do was work. Thank you for your love and support – I could not have done it without you Rūta, I love you insanely.

Curriculum Vitae

Thomas C. King was born on June 18, 1987 in London, United Kingdom. He began studying for a Bachelors in Computer Science in 2008 at King's College London (KCL). He postponed his Bachelors' studies, after two years whereupon he worked as a software developer and architect for UBS. Thomas resumed his studies one year later during which he became interested in AI, in particular on logic based approaches to topics such as belief and change. He completed his Bachelors at KCL, First Class with Honours, and was awarded the best KCL Computer Science dissertation of 2012. He then decided to pursue his interests in AI further by undertaking a PhD. Having considered a PhD in sub–symbolic AI such as genetic algorithms or symbolic AI choosing symbolic AI, he chose the latter focusing on normative reasoning, having become interested in ascribing meaning with tangible and precise conceptualisations.

His PhD research was conducted from 2012 until 2016 at TU Delft, on formalising institutions, in the philosophical and legal sense (e.g. conventions, laws, contracts and promises), with a focus on institutions that ascribe and regulate the creation and behaviour of other institutional rules. Thomas enjoyed attending and presenting his work at numerous conferences and workshops, as well as making research visits to the University of Bath, gaining an interest in the combination of practical and conceptual work on normative reasoning as a result. Hence – in order to pursue the combination of practical and formal work further – from September 2016, Thomas started working at the School of Computing and Communications of Lancaster University, United Kingdom, as a post–doctoral Senior Research Associate researching social commitment reasoning as an abstraction for: databases, decentralising software and verifying potential agent interactions.

SIKS Dissertation Series

2016-41 Thomas C King (TUD), *Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance.*
2016-40 Christian Detweiler (TUD), *Accounting for Values in Design.*
2016-39 Merijn Bruijnes (UT), *Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect.*
2016-38 Andrea Minuto (UT), *MATERIALS THAT MATTER - Smart Materials meet Art & Interaction Design.*
2016-37 Giovanni Sileno (UvA), *Aligning Law and Action - a conceptual and computational inquiry.*
2016-36 Daphne Karreman (UT), *Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies.*
2016-35 Zhaochun Ren, *Monitoring Social Media: Summarization, Classification and Recommendation.*
2016-34 Dennis Schunselaar (TUE), *Title: Configurable Process Trees: Elicitation, Analysis, and Enactment.*
2016-33 Peter Bloem (UVA), *Single Sample Statistics, exercises in learning from just one example.*
2016-32 Eelco Vriezekolk (UT), *Assessing Telecommunication Service Availability Risks for Crisis Organisations.*
2016-31 Mohammad Khelghati (UT), *Deep web content monitoring.*
2016-30 Ruud Mattheij (UvT), *The Eyes Have It.*
2016-29 Nicolas Höning (TUD), *Peak reduction in decentralised electricity systems -Markets and prices for flexible planning.*
2016-28 Mingxin Zhang (TUD), *Large-scale Agent-based Social Simulation - A study on epidemic prediction and control.*
2016-27 Wen Li (TUD), *Understanding Geo-spatial Information on Social Media.*
2016-26 Dilhan Thilakarathne (VU), *In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains.*
2016-25 Julia Kiseleva (TU/e), *Using Contextual Information to Understand Searching and Browsing Behavior.*
2016-24 Brend Wanders (UT), *Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach.*
2016-23 Fei Cai (UVA), *Query Auto Completion in Information Retrieval.*
2016-22 Grace Lewis (VU), *Software Architecture Strategies for Cyber-Foraging Systems.*
2016-21 Alejandro Moreno Célleri (UT), *From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground.*
2016-20 Daan Odijk (UVA), *Context & Semantics in News & Web Search.*
2016-19 Julia Efremova (Tu/e), *Mining Social Structures from Genealogical Data.*

2016-18 Albert Meroño Peñuela, *Refining Statistical Data on the Web.*
2016-17 Berend Weel (VU), *Towards Embodied Evolution of Robot Organisms.*
2016-16 Guangliang Li (UVA), *Socially Intelligent Autonomous Agents that Learn from Human Reward.*
2016-15 Steffen Michels (RUN), *Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments.*
2016-14 Ravi Khadka (UU), *Revisiting Legacy Software System Modernization.*
2016-13 Nana Baah Gyan (VU), *The Web, Speech Technologies and Rural Development in West Africa - An.*
2016-12 Max Knobbout (UU), *Logics for Modelling and Verifying Normative Multi-Agent Systems.*
2016-11 Anne Schuth (UVA), *Search Engines that Learn from Their Users.*
2016-10 George Karafotias (VUA), *Parameter Control for Evolutionary Algorithms.*
2016-09 Archana Nottamkandath (VU), *Trusting Crowdsourced Information on Cultural Artefacts.*
2016-08 Matje van de Camp (TiU), *A Link to the Past: Constructing Historical Social Networks from Unstructured Data.*
2016-07 Jeroen de Man (VU), *Measuring and modeling negative emotions for virtual training.*
2016-06 Michel Wilson (TUD), *Robust scheduling in an uncertain environment.*
2016-05 Evgeny Sherkhonov (UVA), *Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers.*
2016-04 Laurens Rietveld (VU), *Publishing and Consuming Linked Data.*
2016-03 Maya Sappelli (RUN), *Knowledge Work in Context: User Centered Knowledge Worker Support.*
2016-02 Michiel Christiaan Meulendijk (UU), *Optimizing medication reviews through decision support: prescribing a better pill to swallow.*
2016-01 Syed Saiden Abbas (RUN), *Recognition of Shapes by Humans and Machines.*
2015-35 Jungxao Xu (TUD), *Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction.*
2015-34 Victor de Graaf (UT), *Gesocial Recommender Systems.*
2015-33 Frederik Schadd (TUD), *Ontology Mapping with Auxiliary Resources.*
2015-32 Jerome Gard(UL), *Corporate Venture Management in SMEs.*
2015-31 Yakup Koç (TUD), *On the robustness of Power Grids.*
2015-30 Kiavash Bahreini(OU), *Real-time Multimodal Emotion Recognition in E-Learning.*
2015-29 Hendrik Baier (UM), *Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains.*
2015-28 Janet Bagorogoza (TiU), *KNOWLEDGE MANAGEMENT AND HIGH PERFORMANCE; The Uganda Financial Institutions Model for HPO.*
2015-27 Sándor Héman (CWI), *Updating compressed colomn stores.*
2015-26 Alexander Hogenboom (EUR), *Sentiment Analysis of Text Guided by Semantics and Structure.*
2015-25 Steven Woudenberg (UU), *Bayesian Tools for Early Disease Detection.*
2015-24 Richard Berendsen (UVA), *Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation.*

2015-23 Luit Gazendam (VU), *Cataloguer Support in Cultural Heritage.*
2015-21 Sibren Fetter (OUN), *Using Peer-Support to Expand and Stabilize Online Learning.*
2015-20 Loïs Vanhée (UU), *Using Culture and Values to Support Flexible Coordination.*
2015-19 Bernardo Tabuenca (OUN), *Ubiquitous Technology for Lifelong Learners.*
2015-18 Holger Pirk (CWI), *Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories.*
2015-17 André van Cleeff (UT), *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs.*
2015-16 Changyun Wei (UT), *Cognitive Coordination for Cooperative Multi-Robot Teamwork.*
2015-15 Klaas Andries de Graaf (VU), *Ontology-based Software Architecture Documentation.*
2015-14 Bart van Straalen (UT), *A cognitive approach to modeling bad news conversations.*
2015-13 Giuseppe Procaccianti(VU), *Energy-Efficient Software.*
2015-12 Julie M. Birkholz (VU), *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks.*
2015-11 Yongming Luo(TUE), *Designing algorithms for big graph datasets: A study of computing bisimulation and joins.*
2015-10 Henry Hermans (OUN), *OpenU: design of an integrated system to support lifelong learning.*
2015-09 Randy Klaassen(UT), *HCI Perspectives on Behavior Change Support Systems.*
2015-08 Jie Jiang (TUD), *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions.*
2015-07 Maria-Hendrike Peetz(UvA), *Time-Aware Online Reputation Analysis.*
2015-06 Farideh Heidari (TUD), *Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes.*
2015-05 Christoph Bösch (UT), *Cryptographically Enforced Search Pattern Hiding.*
2015-04 Howard Spoelstra (OUN), *Collaborations in Open Learning Environments.*
2015-03 Twan van Laarhoven (RUN), *Machine learning for network data.*
2015-02 Faiza Bukhsh (UvT), *Smart auditing: Innovative Compliance Checking in Customs Controls.*
2015-01 Niels Netten (UvA), *Machine Learning for Relevance of Information in Crisis Response.*
2014-47 Shangsong Liang (UVA), *Fusion and Diversification in Information Retrieval.*
2014-46 Ke Tao (TUD), *Social Web Data Analytics: Relevance, Redundancy, Diversity.*
2014-45 Birgit Schmitz (OUN), *Mobile Games for Learning: A Pattern-Based Approach.*
2014-44 Paulien Meesters (UvT), *Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden..*
2014-43 Kevin Vlaanderen (UU), *Supporting Process Improvement using Method Increments.*
2014-42 Carsten Eijckhof (CWI/TUD), *Contextual Multidimensional Relevance Models.*
2014-41 Frederic Hogenboom (EUR), *Automated Detection of Financial Events in News Text.*

2014-40 Walter Omona (RUN), *A Framework for Knowledge Management Using ICT in Higher Education.*
2014-39 Jasmina Maric (UvT), *Web Communities, Immigration, and Social Capital.*
2014-38 Danny Plass-Oude Bos (UT), *Making brain-computer interfaces better: improving usability through post-processing..*
2014-37 Maral Dadvar (UT), *Experts and Machines United Against Cyberbullying.*
2014-36 Joos Buijs (TUE), *Flexible Evolutionary Algorithms for Mining Structured Process Models.*
2014-35 Joost van Ooijen (UU), *Cognitive Agents in Virtual Worlds: A Middleware Design Approach.*
2014-34 Christina Manteli(VU), *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems..*
2014-33 Tesfa Tegegne (RUN), *Service Discovery in eHealth.*
2014-32 Naser Ayat (UvA), *On Entity Resolution in Probabilistic Data.*
2014-31 Leo van Moergestel (UU), *Agent Technology in Agile Multiparallel Manufacturing and Product Support.*
2014-30 Peter de Cock (UvT), *Anticipating Criminal Behaviour.*
2014-29 Jaap Kabbedijk (UU), *Variability in Multi-Tenant Enterprise Software.*
2014-28 Anna Chmielowiec (VU), *Decentralized k-Clique Matching.*
2014-27 Rui Jorge Almeida (EUR), *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty.*
2014-26 Tim Baarslag (TUD), *What to Bid and When to Stop.*
2014-25 Martijn Lappenschaar (RUN), *New network models for the analysis of disease interaction.*
2014-24 Davide Ceolin (VU), *Trusting Semi-structured Web Data.*
2014-23 Eleftherios Sidirourgos (UvA/CWI), *Space Efficient Indexes for the Big Data Era.*
2014-22 Marieke Peeters (UU), *Personalized Educational Games - Developing agent-supported scenario-based training.*
2014-21 Kassidy Clark (TUD), *Negotiation and Monitoring in Open Environments.*
2014-20 Mena Habib (UT), *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link.*
2014-19 Vinicius Ramos (TUE), *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support.*
2014-18 Mattijs Ghijsen (UVA), *Methods and Models for the Design and Study of Dynamic Agent Organizations.*
2014-17 Kathrin Dentler (VU), *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability.*
2014-16 Krystyna Milian (VU), *Supporting trial recruitment and design by automatically interpreting eligibility criteria.*
2014-15 Natalya Mogles (VU), *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare.*
2014-14 Yangyang Shi (TUD), *Language Models With Meta-information.*
2014-13 Arlette van Wissen (VU), *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains.*
2014-12 Willem van Willigen (VU), *Look Ma, No Hands: Aspects of Autonomous Vehicle*

Control.
2014-11 Janneke van der Zwaan (TUD), *An Empathic Virtual Buddy for Social Support.*
2014-10 Ivan Salvador Razo Zapata (VU), *Service Value Networks.*
2014-09 Philip Jackson (UvT), *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language.*
2014-08 Samur Araujo (TUD), *Data Integration over Distributed and Heterogeneous Data Endpoints.*
2014-07 Arya Adriansyah (TUE), *Aligning Observed and Modeled Behavior.*
2014-06 Damian Tamburri (VU), *Supporting Networked Software Development.*
2014-05 Jurriaan van Reijsen (UU), *Knowledge Perspectives on Advancing Dynamic Capability.*
2014-04 Hanna Jochmann-Mannak (UT), *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation.*
2014-03 Sergio Raul Duarte Torres (UT), *Information Retrieval for Children: Search Behavior and Solutions.*
2014-02 Fiona Tuliyano (RUN), *Combining System Dynamics with a Domain Modeling Method.*
2014-01 Nicola Barile (UU), *Studies in Learning Monotone Models from Data.*
2013-43 Marc Bron (UVA), *Exploration and Contextualization through Interaction and Concepts.*
2013-42 Léon Planken (TUD), *Algorithms for Simple Temporal Reasoning.*
2013-41 Jochem Liem (UVA), *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning.*
2013-40 Pim Nijssen (UM), *Monte-Carlo Tree Search for Multi-Player Games.*
2013-39 Joop de Jong (TUD), *A Method for Enterprise Ontology based Design of Enterprise Information Systems.*
2013-38 Eelco den Heijer (VU), *Autonomous Evolutionary Art.*
2013-37 Dirk Börner (OUN), *Ambient Learning Displays.*
2013-36 Than Lam Hoang (TUe), *Pattern Mining in Data Streams.*
2013-35 Abdallah El Ali (UvA), *Minimal Mobile Human Computer Interaction.*
2013-34 Kien Tjin-Kam-Jet (UT), *Distributed Deep Web Search.*
2013-33 Qi Gao (TUD), *User Modeling and Personalization in the Microblogging Sphere.*
2013-32 Kamakshi Rajagopal (OUN), *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development.*
2013-31 Dinh Khoa Nguyen (UvT), *Blueprint Model and Language for Engineering Cloud Applications.*
2013-30 Joyce Nakatumba (TUE), *Resource-Aware Business Process Management: Analysis and Support.*
2013-29 Iwan de Kok (UT), *Listening Heads.*
2013-28 Frans van der Sluis (UT), *When Complexity becomes Interesting: An Inquiry into the Information eXperience.*
2013-27 Mohammad Huq (UT), *Inference-based Framework Managing Data Provenance.*
2013-26 Alireza Zarghami (UT), *Architectural Support for Dynamic Homecare Service Provisioning.*
2013-25 Agnieszka Anna Latoszek-Berendsen (UM), *Intention-based Decision Support. A*

new way of representing and implementing clinical guidelines in a Decision Support System.
2013-24 Haitham Bou Ammar (UM), *Automated Transfer in Reinforcement Learning.*
2013-23 Patricio de Alencar Silva(UvT), *Value Activity Monitoring.*
2013-22 Tom Claassen (RUN), *Causal Discovery and Logic.*
2013-21 Sander Wubben (UvT), *Text-to-text generation by monolingual machine translation.*
2013-20 Katja Hofmann (UvA), *Fast and Reliable Online Learning to Rank for Information Retrieval.*
2013-19 Renze Steenhuizen (TUD), *Coordinated Multi-Agent Planning and Scheduling.*
2013-18 Jeroen Janssens (UvT), *Outlier Selection and One-Class Classification.*
2013-17 Koen Kok (VU), *The PowerMatcher: Smart Coordination for the Smart Electricity Grid.*
2013-16 Eric Kok (UU), *Exploring the practical benefits of argumentation in multi-agent deliberation.*
2013-15 Daniel Hennes (UM), *Multiagent Learning - Dynamic Games and Applications.*
2013-14 Jafar Tanha (UVA), *Ensemble Approaches to Semi-Supervised Learning Learning.*
2013-13 Mohammad Safiri(UT), *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly.*
2013-12 Marian Razavian(VU), *Knowledge-driven Migration to Services.*
2013-11 Evangelos Pournaras(TUD), *Multi-level Reconfigurable Self-organization in Overlay Services.*
2013-10 Jeewanie Jayasinghe Arachchige(UvT), *A Unified Modeling Framework for Service Design..*
2013-09 Fabio Gori (RUN), *Metagenomic Data Analysis: Computational Methods and Applications.*
2013-08 Robbert-Jan Merk(VU), *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators.*
2013-07 Giel van Lankveld (UvT), *Quantifying Individual Player Differences.*
2013-06 Romulo Goncalves(CWI), *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience.*
2013-05 Dulce Pumareja (UT), *Groupware Requirements Evolutions Patterns.*
2013-04 Chetan Yadati(TUD), *Coordinating autonomous planning and scheduling.*
2013-03 Szymon Klarman (VU), *Reasoning with Contexts in Description Logics.*
2013-02 Erietta Liarou (CWI), *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing.*
2013-01 Viorel Milea (EUR), *News Analytics for Financial Decision Support.*
2012-51 Jeroen de Jong (TUD), *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching.*
2012-50 Steven van Kervel (TUD), *Ontologogy driven Enterprise Information Systems Engineering.*
2012-49 Michael Kaisers (UM), *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions.*
2012-48 Jorn Bakker (TUE), *Handling Abrupt Changes in Evolving Time-series Data.*

2012-47 Manos Tsagkias (UVA), *Mining Social Media: Tracking Content and Predicting Behavior.*
2012-46 Simon Carter (UVA), *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation.*
2012-45 Benedikt Kratz (UvT), *A Model and Language for Business-aware Transactions.*
2012-44 Anna Tordai (VU), *On Combining Alignment Techniques.*
2012-43 Withdrawn, *N/A.*
2012-42 Dominique Verpoorten (OU), *Reflection Amplifiers in self-regulated Learning.*
2012-41 Sebastian Kelle (OU), *Game Design Patterns for Learning.*
2012-40 Agus Gunawan (UvT), *Information Access for SMEs in Indonesia.*
2012-39 Hassan Fatemi (UT), *Risk-aware design of value and coordination networks.*
2012-38 Selmar Smit (VU), *Parameter Tuning and Scientific Testing in Evolutionary Algorithms.*
2012-37 Agnes Nakakawa (RUN), *A Collaboration Process for Enterprise Architecture Creation.*
2012-36 Denis Ssebugwawo (RUN), *Analysis and Evaluation of Collaborative Modeling Processes.*
2012-35 Evert Haasdijk (VU), *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics.*
2012-34 Pavol Jancura (RUN), *Evolutionary analysis in PPI networks and applications.*
2012-33 Rory Sie (OUN), *Coalitions in Cooperation Networks (COCOON).*
2012-32 Wietske Visser (TUD), *Qualitative multi-criteria preference representation and reasoning.*
2012-31 Emily Bagarukayo (RUN), *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure.*
2012-30 Alina Pommeranz (TUD), *Designing Human-Centered Systems for Reflective Decision Making.*
2012-29 Almer Tigelaar (UT), *Peer-to-Peer Information Retrieval.*
2012-28 Nancy Pascall (UvT), *Engendering Technology Empowering Women.*
2012-27 Hayrettin Gurkok (UT), *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games.*
2012-26 Emile de Maat (UVA), *Making Sense of Legal Text.*
2012-25 Silja Eckartz (UT), *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application.*
2012-24 Laurens van der Werff (UT), *Evaluation of Noisy Transcripts for Spoken Document Retrieval.*
2012-23 Christian Muehl (UT), *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction.*
2012-22 Thijs Vis (UvT), *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?.*
2012-21 Roberto Cornacchia (TUD), *Querying Sparse Matrices for Information Retrieval.*
2012-20 Ali Bahramisharif (RUN), *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing.*
2012-19 Helen Schonenberg (TUE), *What's Next? Operational Support for Business Process Execution.*

2012-18 Eltjo Poort (VU), *Improving Solution Architecting Practices.*
2012-17 Amal Elgammal (UvT), *Towards a Comprehensive Framework for Business Process Compliance.*
2012-16 Fiemke Both (VU), *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment.*
2012-15 Natalie van der Wal (VU), *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes..*
2012-14 Evgeny Knutov(TUE), *Generic Adaptation Framework for Unifying Adaptive Web-based Systems.*
2012-13 Suleman Shahid (UvT), *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions.*
2012-12 Kees van der Sluijs (TUE), *Model Driven Design and Data Integration in Semantic Web Information Systems.*
2012-11 J.C.B. Rantham Prabhakara (TUE), *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics.*
2012-10 David Smits (TUE), *Towards a Generic Distributed Adaptive Hypermedia Environment.*
2012-09 Ricardo Neisse (UT), *Trust and Privacy Management Support for Context-Aware Service Platforms.*
2012-08 Gerben de Vries (UVA), *Kernel Methods for Vessel Trajectories.*
2012-07 Rianne van Lambalgen (VU), *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions.*
2012-06 Wolfgang Reinhardt (OU), *Awareness Support for Knowledge Workers in Research Networks.*
2012-05 Marijn Plomp (UU), *Maturing Interorganisational Information Systems.*
2012-04 Jurriaan Souer (UU), *Development of Content Management System-based Web Applications.*
2012-03 Adam Vanya (VU), *Supporting Architecture Evolution by Mining Software Repositories.*
2012-02 Muhammad Umair(VU), *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models.*
2012-01 Terry Kakeeto (UvT), *Relationship Marketing for SMEs in Uganda.*
2011-49 Andreea Niculescu (UT), *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality.*
2011-48 Mark Ter Maat (UT), *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent.*
2011-47 Azizi Bin Ab Aziz(VU), *Exploring Computational Models for Intelligent Support of Persons with Depression.*
2011-46 Beibei Hu (TUD), *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work.*
2011-45 Herman Stehouwer (UvT), *Statistical Language Models for Alternative Sequence Selection.*
2011-44 Boris Reuderink (UT), *Robust Brain-Computer Interfaces.*
2011-43 Henk van der Schuur (UU), *Process Improvement through Software Operation Knowledge.*

2011-42 Michal Sindlar (UU), *Explaining Behavior through Mental State Attribution.*
2011-41 Luan Ibraimi (UT), *Cryptographically Enforced Distributed Data Access Control.*
2011-40 Viktor Clerc (VU), *Architectural Knowledge Management in Global Software Development.*
2011-39 Joost Westra (UU), *Organizing Adaptation using Agents in Serious Games.*
2011-38 Nyree Lemmens (UM), *Bee-inspired Distributed Optimization.*
2011-37 Adriana Burlutiu (RUN), *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference.*
2011-36 Erik van der Spek (UU), *Experiments in serious game design: a cognitive approach.*
2011-35 Maaike Harbers (UU), *Explaining Agent Behavior in Virtual Training.*
2011-34 Paolo Turrini (UU), *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations.*
2011-33 Tom van der Weide (UU), *Arguing to Motivate Decisions.*
2011-32 Nees-Jan van Eck (EUR), *Methodological Advances in Bibliometric Mapping of Science.*
2011-31 Ludo Waltman (EUR), *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality.*
2011-30 Egon van den Broek (UT), *Affective Signal Processing (ASP): Unraveling the mystery of emotions.*
2011-29 Faisal Kamiran (TUE), *Discrimination-aware Classification.*
2011-28 Rianne Kaptein(UVA), *Effective Focused Retrieval by Exploiting Query Context and Document Structure.*
2011-27 Aniel Bhulai (VU), *Dynamic website optimization through autonomous management of design patterns.*
2011-26 Matthijs Aart Pontier (VU), *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots.*
2011-25 Syed Waqar ul Qounain Jaffry (VU)), *Analysis and Validation of Models for Trust Dynamics.*
2011-24 Herwin van Welbergen (UT), *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior.*
2011-23 Wouter Weerkamp (UVA), *Finding People and their Utterances in Social Media.*
2011-22 Junte Zhang (UVA), *System Evaluation of Archival Description and Access.*
2011-21 Linda Terlouw (TUD), *Modularization and Specification of Service-Oriented Systems.*
2011-20 Qing Gu (VU), *Guiding service-oriented software engineering - A view-based approach.*
2011-19 Ellen Rusman (OU), *The Mind ' s Eye on Personal Profiles.*
2011-18 Mark Ponsen (UM), *Strategic Decision-Making in complex games.*
2011-17 Jiyin He (UVA), *Exploring Topic Structure: Coherence, Diversity and Relatedness.*
2011-16 Maarten Schadd (UM), *Selective Search in Games of Different Complexity.*
2011-15 Marijn Koolen (UvA), *The Meaning of Structure: the Value of Link Evidence for*

Information Retrieval.
2011-14 Milan Lovric (EUR), *Behavioral Finance and Agent-Based Artificial Markets.*
2011-13 Xiaoyu Mao (UvT), *Airport under Control. Multiagent Scheduling for Airport Ground Handling.*
2011-12 Carmen Bratosin (TUE), *Grid Architecture for Distributed Process Mining.*
2011-11 Dhaval Vyas (UT), *Designing for Awareness: An Experience-focused HCI Perspective.*
2011-10 Bart Bogaert (UvT), *Cloud Content Contention.*
2011-09 Tim de Jong (OU), *Contextualised Mobile Media for Learning.*
2011-08 Nieske Vergunst (UU), *BDI-based Generation of Robust Task-Oriented Dialogues.*
2011-07 Yujia Cao (UT), *Multimodal Information Presentation for High Load Human Computer Interaction.*
2011-06 Yiwen Wang (TUE), *Semantically-Enhanced Recommendations in Cultural Heritage.*
2011-05 Base van der Raadt (VU), *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline..*
2011-04 Hado van Hasselt (UU), *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference.*
2011-03 Jan Martijn van der Werf (TUE), *Compositional Design and Verification of Component-Based Information Systems.*
2011-02 Nick Tinnemeier(UU), *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language.*
2011-01 Botond Cseke (RUN), *Variational Algorithms for Bayesian Inference in Latent Gaussian Models.*
2010-53 Edgar Meij (UVA), *Combining Concepts and Language Models for Information Access.*
2010-52 Peter-Paul van Maanen (VU), *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention.*
2010-51 Alia Khairia Amin (CWI), *Understanding and supporting information seeking tasks in multiple sources.*
2010-50 Bouke Huurnink (UVA), *Search in Audiovisual Broadcast Archives.*
2010-49 Jahn-Takeshi Saito (UM), *Solving difficult game positions.*
2010-48 Withdrawn, .
2010-47 Chen Li (UT), *Mining Process Model Variants: Challenges, Techniques, Examples.*
2010-46 Vincent Pijpers (VU), *e3alignment: Exploring Inter-Organizational Business-ICT Alignment.*
2010-45 Vasilios Andrikopoulos (UvT), *A theory and model for the evolution of software services.*
2010-44 Pieter Bellekens (TUE), *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain.*
2010-43 Peter van Kranenburg (UU), *A Computational Approach to Content-Based Retrieval of Folk Song Melodies.*
2010-42 Sybren de Kinderen (VU), *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach.*

2010-41 Guillaume Chaslot (UM), *Monte-Carlo Tree Search.*
2010-40 Mark van Assem (VU), *Converting and Integrating Vocabularies for the Semantic Web.*
2010-39 Ghazanfar Farooq Siddiqui (VU), *Integrative modeling of emotions in virtual agents.*
2010-38 Dirk Fahland (TUE), *From Scenarios to components.*
2010-37 Niels Lohmann (TUE), *Correctness of services and their composition.*
2010-36 Jose Janssen (OU), *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification.*
2010-35 Dolf Trieschnigg (UT), *Proof of Concept: Concept-based Biomedical Information Retrieval.*
2010-34 Teduh Dirgahayu (UT), *Interaction Design in Service Compositions.*
2010-33 Robin Aly (UT), *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval.*
2010-32 Marcel Hiel (UvT), *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems.*
2010-31 Victor de Boer (UVA), *Ontology Enrichment from Heterogeneous Sources on the Web.*
2010-30 Marieke van Erp (UvT), *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval.*
2010-29 Stratos Idreos(CWI), *Database Cracking: Towards Auto-tuning Database Kernels.*
2010-28 Arne Koopman (UU), *Characteristic Relational Patterns.*
2010-27 Marten Voulon (UL), *Automatisch contracteren.*
2010-26 Ying Zhang (CWI), *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines.*
2010-25 Zulfiqar Ali Memon (VU), *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective.*
2010-24 Dmytro Tykhonov, *Designing Generic and Efficient Negotiation Strategies.*
2010-23 Bas Steunebrink (UU), *The Logical Structure of Emotions.*
2010-22 Michiel Hildebrand (CWI), *End-user Support for Access to Heterogeneous Linked Data.*
2010-21 Harold van Heerde (UT), *Privacy-aware data management by means of data degradation.*
2010-20 Ivo Swartjes (UT), *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative.*
2010-19 Henriette Cramer (UvA), *People's Responses to Autonomous and Adaptive Systems.*
2010-18 Charlotte Gerritsen (VU), *Caught in the Act: Investigating Crime by Agent-Based Simulation.*
2010-17 Spyros Kotoulas (VU), *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications.*
2010-16 Sicco Verwer (TUD), *Efficient Identification of Timed Automata, theory and practice.*
2010-15 Lianne Bodenstaff (UT), *Managing Dependency Relations in Inter-Organizational*

Models.
2010-14 Sander van Splunter (VU), *Automated Web Service Reconfiguration.*
2010-13 Gianluigi Folino (RUN), *High Performance Data Mining using Bio-inspired techniques.*
2010-12 Susan van den Braak (UU), *Sensemaking software for crime analysis.*
2010-11 Adriaan Ter Mors (TUD), *The world according to MARP: Multi-Agent Route Planning.*
2010-10 Rebecca Ong (UL), *Mobile Communication and Protection of Children.*
2010-09 Hugo Kielman (UL), *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging.*
2010-08 Krzysztof Siewicz (UL), *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments.*
2010-07 Wim Fikkert (UT), *Gesture interaction at a Distance.*
2010-06 Sander Bakkes (UvT), *Rapid Adaptation of Video Game AI.*
2010-05 Claudia Hauff (UT), *Predicting the Effectiveness of Queries and Retrieval Systems.*
2010-04 Olga Kulyk (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments.*
2010-03 Joost Geurts (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents.*
2010-02 Ingo Wassink (UT), *Work flows in Life Science.*
2010-01 Matthijs van Leeuwen (UU), *Patterns that Matter.*
2009-46 Loredana Afanasiev (UvA), *Querying XML: Benchmarks and Recursion.*
2009-45 Jilles Vreeken (UU), *Making Pattern Mining Useful.*
2009-44 Roberto Santana Tapia (UT), *Assessing Business-IT Alignment in Networked Organizations.*
2009-43 Virginia Nunes Leal Franqueira (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients.*
2009-42 Toine Bogers (UvT), *Recommender Systems for Social Bookmarking.*
2009-41 Igor Berezhnyy (UvT), *Digital Analysis of Paintings.*
2009-40 Stephan Raaijmakers (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language.*
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets.*
2009-38 Riina Vuorikari (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context.*
2009-37 Hendrik Drachsler (OUN), *Navigation Support for Learners in Informal Learning Networks.*
2009-36 Marco Kalz (OUN), *Placement Support for Learners in Learning Networks.*
2009-35 Wouter Koelewijn (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling.*
2009-34 Inge van de Weerd (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach.*
2009-33 Khiet Truong (UT), *How Does Real Affect Affect Affect Recognition In Speech?.*
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU), *Architectural Knowledge Man-*

agement: Supporting Architects and Auditors.
2009-31 Sofiya Katrenko (UVA), *A Closer Look at Learning Relations from Text.*
2009-30 Marcin Zukowski (CWI), *Balancing vectorized query execution with bandwidth-optimized storage.*
2009-29 Stanislav Pokraev (UT), *Model-Driven Semantic Integration of Service-Oriented Applications.*
2009-28 Sander Evers (UT), *Sensor Data Management with Probabilistic Models.*
2009-27 Christian Glahn (OU), *Contextual Support of social Engagement and Reflection on the Web.*
2009-26 Fernando Koch (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services.*
2009-25 Alex van Ballegooij (CWI), *"RAM: Array Database Management through Relational Mapping".*
2009-24 Annerieke Heuvelink (VUA), *Cognitive Models for Training Simulations.*
2009-23 Peter Hofgesang (VU), *Modelling Web Usage in a Changing Environment.*
2009-22 Pavel Serdyukov (UT), *Search For Expertise: Going beyond direct evidence.*
2009-21 Stijn Vanderlooy (UM), *Ranking and Reliable Classification.*
2009-20 Bob van der Vecht (UU), *Adjustable Autonomy: Controling Influences on Decision Making.*
2009-19 Valentin Robu (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets.*
2009-18 Fabian Groffen (CWI), *Armada, An Evolving Database System.*
2009-17 Laurens van der Maaten (UvT), *Feature Extraction from Visual Data.*
2009-16 Fritz Reul (UvT), *New Architectures in Computer Chess.*
2009-15 Rinke Hoekstra (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense.*
2009-14 Maksym Korotkiy (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA).*
2009-13 Steven de Jong (UM), *Fairness in Multi-Agent Systems.*
2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), *Operating Guidelines for Services.*
2009-11 Alexander Boer (UVA), *Legal Theory, Sources of Law & the Semantic Web.*
2009-10 Jan Wielemaker (UVA), *Logic programming for knowledge-intensive interactive applications.*
2009-09 Benjamin Kanagwa (RUN), *Design, Discovery and Construction of Service-oriented Systems.*
2009-08 Volker Nannen (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments.*
2009-07 Ronald Poppe (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion.*
2009-06 Muhammad Subianto (UU), *Understanding Classification.*
2009-05 Sietse Overbeek (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality.*
2009-04 Josephine Nabukenya (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering.*

2009-03 Hans Stol (UvT), *A Framework for Evidence-based Policy Making Using IT.*
2009-02 Willem Robert van Hage (VU), *Evaluating Ontology-Alignment Techniques.*
2009-01 Rasa Jurgelenaite (RUN), *Symmetric Causal Independence Models.*