FROM NORMS TO OBLIGATIONS REVISITED: A CASE OF THREE-DIMENSIONAL VIRTUAL WORLDS

Vytautas Čyras / Friedrich Lachmayer

Docent, Vilnius University, Faculty of Mathematics and Informatics
Naugarduko 24, 03225 Vilnius, LT
vytautas.cyras@mif.vu.lt; http://www.mif.vu.lt/~cyras/

Professor, University of Innsbruck, Faculty of Law
Innrain 52, 6020 Innsbruck, AT
friedrich.lachmayer@uibk.ac.at; http://www.legalvisualization.com/

Keywords: virtual law, legal rules, technical rules, ought-to-be in-world reality, normative resultant, conceptualisation

Abstract: In this paper we reflect on constructing legal frameworks which affect online virtual worlds. A virtual world is viewed as a whole and depicted by the metaphor of stage. This is a model, a symbolisation of the conceptual framework of the virtual world, within the methodological concept of cognitive spatialisation. In this paper symbolisation precedes formalisation on the way to construction and legal knowledge representation. Thus visualization of virtual world issues can have a favourable effect on modelling, formal representation and operational implementation of rules. Rules (norms) are divided into classes such as legal, technical (technological), professional, reputation, energy, etc. Distinguished classes have different modes of effect: legal rules can be violated but it is impossible to violate technical rules. We also attempt to formalise the normative status of the rule’s addressee. Therefore, the concepts of unified duty and summary duty are revisited. The input/output of a normative status synthesizer is specified. An example explores duties and permissions of a girl in a cafe. A list of duties is interpreted with a connective intersection and a list of permissions – with union.

1. Introduction

This paper explores a person’s normative status that is entailed by a given situation. Therefore the theory of norms which was developed by Lachmayer (1977) is employed. However the present context is new. We concern the normative regulation of avatars in three-dimensional online virtual worlds² (see Fig. 1). Hence, virtual worlds are viewed as an application domain.

Fig. 1: Interacting with a complex 3D geometric object, dodecahedron, in the context of a geometry lesson within a virtual world [Čyras and Lapin 2010]

To explore the above mentioned issues, normative multiagent systems are reviewed. The norms and obligations of software agents have been investigated traditionally in informatics: computer science researchers proposing formalisms and programming languages for operationalisation of norms in software.

Sketching the legal issues of virtual worlds is first dealt in this paper. The metaphor of stage is employed here. Such a general sketch presents a personal reflection of the authors; it serves as visualization³. It plays the role of a model, a symbolisation of the conceptual framework. It is generally called spatialisation.

---

² Non-game virtual worlds such as Second Life, developed by Linden Lab, http://secondlife.com/, are in the primary focus. About ‘serious’ virtual worlds – opposed to leisure-based – see a scoping study by Sara de Freitas (2008).
³ Eppler and Burkhard (2006, p. 553) emphasize the role of visual cognition and perception and note that a majority of our brain’s activity deals with processing and analysing visual images: “If visualization is applied correctly, it dramatically increases our ability to think and communicate.”
Further the differences between the types of rules are addressed. Technical rules are impossible to violate, but legal rules (norms) can be violated.

All the rules are subject to implementation in virtual world software. In this context, law enforcement is a tough issue. The addressee concept of classical rules – persons – is supplemented with avatars. A sample “toy” rule ‘Keep off the grass’ can illustrate constraints on avatar behaviour. Various methodologies are investigated worldwide to program the rules. Programming languages include external tools and script languages (e.g. LSL, Python, Lua, etc.). As a programming exercise, consider a virtual world and the following scenario. Suppose you build a house with a door which is controlled by a rule which permits entering avatars from a certain group only, e.g. your trusted friend avatars; other avatars are forbidden. In Second Life, this rule can be programmed by LSL (Linden Script Language). To program rules, each virtual world platform needs a specific framework.

The explored issues tackle the virtual world platform which is being developed within the VirtualLife project; see Section 3 for a fuller outline. Here the legal framework being constructed comprises a Virtual Constitution. The operational implementation of rules pays respect to Vázquez-Salceda et al. (2008).

A major part of this paper is devoted to formalising normative status. The notions of unified duty and summary duty are investigated. An example illustrates duties and permissions of a girl who sits in a cafe with her father.

In summary, this paper concerns structuring of legal semantics, in particular, duties and obligations.

2. The Frame of a Virtual World

This section describes informally different kinds of rules in three dimensional virtual worlds that are inspired by a metaphor of architecture. Such an immersive virtual environment is different from 2D menu paradigm.

The frame – the reality of a virtual world – is constitutive. The entities are comprised of avatars, and 3D objects such as buildings, rooms, information stands, avatar inventories, etc. (see Fig. 1). An avatar is a computer user’s representation (usually a 3D model). Every avatar is monitored by its user. The avatar can also be viewed as the cursor on the screen.

The frame of a virtual world is established from the outset. A conceptualisation of the “theatre” is depicted in Fig. 2. The entities of major importance are avatars,

---

their actions and rules, which regulate the behaviour. The avatars can also engage in joint actions. Avatar chat communication can be via voice or text chat.

Rules establish a regime (paradigm) of the virtual world. They can be divided into different classes such as technological rules (further – technical rules), legal rules, reputation rules, energy rules, professional rules, etc.:

**Technical rules.** They establish factual limitations. Real world examples are to fence the grass, to close the door to forbid entering, to refuse money by a cash machine if you do not provide the PIN code, etc. Violations are impossible: there is no possibility of violating a technical rule unless you break the artefact completely. Hence technical rules are strictly enforceable.

**Legal rules.** Their nature is that they can be violated. For example, you can step the grass, you can take a train without a ticket, etc. But you risk being sanctioned. A procedure is enforced by an authority such as police, peacekeepers in a virtual world, etc. A sanction is imposed with a probability. Hence the legal rules are not strictly enforceable. Herein is the spirit of law. There is no strict boundary between technical and legal rules; a grey area exists.

The structural elements of legal rules are the same as of real-world norms. The subject is a natural person, a juridical person, etc. The modus is a deontic notion: obligatory, permitted, prohibited, etc.

Legal rules are of two kinds. The primary rules come first. Sanctions are determined by secondary rules. They are associated with authorities which enforce the rules. Virtual procedures can be raised for violations. For instance, online dispute resolution is a kind of procedure.
The modes of effect can be different. A sanction can be raised randomly with a certain probability $p\%$. An example is a road rule which forbids crossing a street against traffic lights. You can go through a red light and if you are lucky a policeman will not punish you.

Legal rules are essential in virtual worlds – similarly as in the real world. It is impossible to implement normative regulation in a virtual world by means of technical rules only. Consider the norm that indecent content is forbidden in the virtual world. Such an abstract norm can hardly be implemented by automatic checking. For instance, a naked body need not be automatically treated as indecent content; this may be a picture of a statue in a virtual museum.

In VirtualLife a user agrees on a so called Virtual Constitution which consists of abstract legal rules. The Virtual Constitution is implemented as a click-wrap agreement. In this way it is part of End User License Agreement (EULA) and binds the users on the contractual level.

**Energy rules.** They sanction in the following way. If a prescribed behaviour is violated appropriate “energy” points are reduced. The addressee can be sanctioned effectively, i.e. 100%.

This can be illustrated by security/trust rules, reputation rules and the avatar identity card in VirtualLife. Each avatar has an ID card, which contains information about both his virtual and real life identities; see Fig. 3 [Bogdanov et al. 2009].

![Fig. 3: Virtual identity card in VirtualLife [Bogdanov et al. 2009]. The reputation of avatars can be taken into account while chatting](image)

The ID card includes simple indicators of trust. A red (entrusted) bar means that the avatar is a guest and has not proved his identity; a yellow (weakly trusted) bar – the avatar has an identity, but it has not been verified by any certification authority; and a green (trusted) bar – the avatar’s identity has been verified by a certification authority. Each avatar also has an economic, social and civic reputation, whose indicators are handled by a sophisticated reputation system,
depending on the avatar’s behaviour. Thus the rules are implemented in software by hard constraints.

**Professional rules.** Other kinds of rules comprise moral rules, user community rules, etc. Examples are provided further from the VirtualLife project where the users can engage in a trusted community called Virtual Nation. Different communities are governed by different rules. For instance, Students and Teachers form distinguished Virtual Nations in a University Virtual Campus that is purposed at learning support [Čyrs and Lapin 2010].

The users are not absolutely free to agree on the community rules. Spindler et al. (2010) emphasize that the users cannot escape from the real world law.

**Virtual Nation scenarios.** A number of examples for allowing different scenarios can be proposed. Different values and rules are covered by NoCopy, CopyRight and CopyLeft nations⁵. Permission language tables serve to implement distinguished rights. The permissions are represented explicitly in computer code.

In a NoCopy nation, no one is allowed to make copies. In a strict copyright nation, only the author of the object can make copies. The receiver of the copied object cannot make copies.

In a non-strict copyright nation, the original author of an object can decide if the new owner can copy the object or not. By default, an author or owner can copy.

In a CopyLeft nation, the creator’s information is preserved. Each user can change and copy its own objects.

In a Second Life model nation, you can sell objects controlling whether they are editable, copyable or sellable. The creator cannot modify a sold object. Of course, you can always move your own objects. For example, the seller can decide to sell non-copyable objects. This object cannot be duplicated by the new owner. Therefore if she puts them in her inventory they disappear from the world.

**3. Principles of Construction of a Virtual World Legal Framework**

Rules can form different normative systems. Virtual world rules have different modes of effect or relevance, such as “barrier”, “occasional”, “step-by-step”, etc. (see Fig. 4).

**Barrier effect.** A door is an example of a barrier technical rule. The technical rules are implemented with hard constraints. For implementation see Guideline 2 in Boella et al. (2009).

---

⁵ The descriptions and designs in the virtual reality engine of VirtualLife software are mainly by Kevin Glass.
Occasional effect. The occasional type rule sanctions you with a certain probability. An example is a legal rule to buy the ticket when you take a train. However, you can succeed to travel without the ticket, but if a conductor does not sanction you.

A dream of computer scientists (knowledge engineers) might be to implement the sanctions of each legal rule by means of technical rules. However this is against the spirit of law; see also the discussion of Philip Leith (2010).

Step-by-step effect. It is characteristic to reputation and energy rules. For example, each time you violate a rule, your points are decreased by 10%.

Ontologies. A core ontology is immanent within the whole set of rules. A special (material) domain ontology is present within each kind of rules.

Technical filters. They can be set to enforce rules. For instance, in the context of e-law, legal texts can go through an XML filter. A document cannot be put on the “transport belt” of legislative workflow in case the document’s XML structure is flawed, e.g. marked-up with errors. Such a filter is an example of a technical rule. The ruling is of the type “Entering without stop is refused.”

3.1. Three Legal Stages

Three legal stages can be distinguished within a virtual world: (1) the legislative stage, (2) the negotiation stage – the stage of the game, and (3) the judicial stage; see Fig. 5. Their functions accord with the functions of law.

The legislative stage serves to produce rules. Here the whole in-world community can be involved. Next, the negotiation stage serves for everyday life. Social games including contracts are performed here. The judicial stage serves for
judgments. This stage is not entered in each case. Distinguished modes of (virtual) legal proceedings are already identified above. Here activities such as story-telling, adjudication dialogs and legal disputes can be invoked.

Fig. 5: Three legal stages of a virtual world

The legislative stage consists of two sub-stages. On the first one, 1a in Fig. 6, legislative (general) rules are produced. It is compared with parliamentary activities, but performed by the community. On the second sub-stage, 1b, contract (individual) rules are produced.

Fig. 6: Two legislative sub-stages serve to produce general rules and individual rules

A sub-stage of access can be distinguished within the stage of the game. It can be compared with an entering scenario. For example, in an airport a citizen with the valid ticket passes a procedure to become a passenger.

The people think in roles, not rules. In a virtual world, the concept of role is more important than the rule concept. In the real world people on the streets think in terms of roles. The people even do not know rules and paragraphs of the law. The first attempt to model the role concept is to put a label. Roles are widely
employed in software; recall the roles of administrator, user, guest, etc. and
distinguishing their access rights.

3.2. Formalising Rules

The concept of technical rule is formalised in terms of necessity (a causal relationship). A legal rule is formalised as a formula in deontic logic; see among others, [Lachmayer 1977] and [Jones et al. 1992]. Energy rule prohibits certain behaviour; if violated addressee’s “energy points” are decreased.

A sketch, a kind of symbolisation, is below. Formalising normative systems, modelling legal argument and operational implementation of rules attracted a broad research of other authors; see e.g. [Prakken 1997], [Sartor 2007].

3.2.1. Technical Rules

Technical rules are based on natural necessity (causation). They are formalised “If $P$, then $Q$”. In inference they can be employed with the modus ponens rule:

$$
\text{If } P, \text{ then } Q.
\text{Therefore } Q.
$$

Or in the sequent notation, $P \rightarrow Q$. An example is the cash machine’s rule that is worded:

‘if PIN code is provided then the cash machine gives money’

Formally: $\text{Rule}(\text{pin} \rightarrow \text{money})$.

The idea of the rule is the entailment that in the case the PIN code is not provided, the cash machine gives no money, formally:

$$
\text{Rule}(\text{pin} \rightarrow \text{money}), \text{Fact}(\neg \text{pin}) \rightarrow \neg \text{Fact}(\neg \text{money})
$$

It can be discussed whether the notation of the technical rule is “if and only if” (denoted iff or ↔):

‘if pin is provided then cash machine gives money, and
if pin is not provided then cash machine does not give money’

Formally: $\text{Rule}(\text{pin} \leftrightarrow \text{money})$.

It should be noted that technical rules do not have the structural element modus. Therefore ‘ought’ (obligatory) or ‘may’ (permitted) is not included even in the grammatical formulation.

The technical rule concept can be viewed in the context of rule-based knowledge representation and compared with active rules in databases and business rules; cf. [Vasilecas et al. 2009].
3.2.2. Legal Rules

Deontic status (e.g. obligatory, permitted, prohibited, etc.) is a structural element of legal rules. The nature of the legal rules is that a prescribed behaviour, $Q$, can be violated. Following is an example of an observed situation:

\[
\begin{align*}
\text{Obligation}(P \rightarrow Q) & \quad \text{– rule.} \\
\text{Fact}(P) & \quad \text{– fact.} \\
\hline
\text{Fact}(\neg Q) & \quad \text{– observed violation fact.}
\end{align*}
\]

For example, the traffic lights rule permits you to cross only if the lights are green: ‘Permitted to cross iff green’. However, you can cross against the lights:

\[
\begin{align*}
\text{Permission}(P \leftrightarrow Q) & \quad \text{– cross only on green.} \\
\text{Fact}(\neg P) & \quad \text{– red is on.} \\
\hline
\text{Fact}(Q) & \quad \text{– however you cross.}
\end{align*}
\]

The observed behaviour is interpreted that you are simply a bad guy and nobody can stop you crossing.

Rule violations are sanctioned randomly with certain probabilities.

3.2.3. Energy Rules

Energy (reputation, etc.) rules prohibit certain behaviour. Consider you violate a rule. Therefore, your energy points are reduced. Hence, you are sanctioned 100%.

An example is as follows:

\[
\begin{align*}
\text{Norm}(\neg A) & \quad \text{– A is prohibited.} \\
\text{Fact}(A) & \quad \text{– you violate the rule.} \\
\hline
\text{Points of A are reduced} & \quad \text{– e.g., } A := 0.9 \times A
\end{align*}
\]

Fig. 7: Energy points are reduced upon the violations of a rule
The interpretation is that your energy points are reduced to $A_1$, then $A_2$ and so on until $A_n$. At last the state $\neg A$ is reached, which denotes no energy at all (Fig. 7). The shark fish is an example in the nature. A shark drowns if swallows too much.

4. Normative Resultants

We view avatars as subjects to normative positions of virtual world rules (Fig. 8).

![Fig. 8: A spatialisation of norm and status; cf. [Lachmayer 1977, p. 88]](image)

This section is devoted to formalise the concept of normative resultants of rules. Two concepts – unified status (einheitlicher Status) and summary status (zusammenfassender Status) – are explored. Here we follow the theory of norms and the concept of normative resultants (normative Resultanten) [Lachmayer 1977, p. 88-92].

Consider a set of rules $\{r_1, r_2, \ldots, r_n\}$. Then different variants of normative status can be built. For simplicity a case of duties is analysed below. Hence, the normative status of the addressee’s duties is concerned.

**Denotations.** The structural elements of rule are the same in both the real and a virtual world. The essential elements of a rule, $r$, are denoted as follows:

1. **Condition** $CON$. Denoted by $condition(r)$
2. **Disposition**
   a. **subject** $SUB$. It is denoted by $subject(r)$. This is the addressee;
   b. **action** $A$. It is denoted by $action(r)$;
   c. **normative modus** ($obligatio$, $permissio$, etc.). It is denoted by $modus(r)$;
   d. **object** $OBJ$. It is denoted by $object(r)$.
3. **Sanction.** It is established by a secondary rule.

Hence the rule, $r$, is constructed of its elements:
The constructor of $r$ is denoted for short by $r = \langle \text{CON, SUB, A, MOD, OBJ} \rangle$.

The existence (validity) of the rule is denoted by a predicate $N$:

$$N(\text{condition}=\text{CON}, \text{subject}=\text{SUB}, \text{action}=\text{A}, \text{modus}=\text{MOD}, \text{object}=\text{OBJ})$$

This is also denoted by $N(A)$, or $N(A)$ – to stress the behaviour $A$.

In virtual worlds the subject concept is extended. Apart of natural persons and juridical persons, the subject comprises avatars. Hence, in the case of technical rules, the addressee is extended to real/virtual world objects such as cash machines, doors, houses, geometrical bodies (cubes, dodecahedrons), etc.

### 4.1. Deontic Field

Relations between rule (norm) and the normative status of the duty have to be commented; see [Lachmayer 1977, p. 75-76]. Consider a norm $N(A)$. The 4 cases can be created:

1. $N(A)$ – commandment (obligatory rule, Gebot);
2. $\neg N(A)$ – absence of commandment;
3. $\neg N(\neg A)$ – absence of prohibition;
4. $N(\neg A)$ – prohibition (prohibitive rule, Verbot).

From each of the four cases above, a respective normative status is derived:

5. $N(A) \rightarrow O(A)$ – From commandment an obligatory duty arises (Aus dem Gebot ergibt sich eine gebietende Pflicht).
6. $\neg N(A) \rightarrow \neg O(A)$ – From the absence of commandment no obligatory duty arises (Aus dem Mangel des Gebotes kann keine gebietende Pflicht abgeleitet werden).
7. $\neg N(\neg A) \rightarrow \neg O(\neg A)$ – From the absence of prohibition no prohibitive duty arises (Liegt kein Verbot vor, so folgt daraus keine verbietende Pflicht).
8. $N(\neg A) \rightarrow O(\neg A)$ – From prohibition a prohibitive duty arises (Liegt ein Verbot vor, so ergibt sich eine verbietende Pflicht).

The spirit of commandment and prohibition is expressed by 5 and 8 respectively.

### 4.2. The Unified Status

The unified status (eieheitlicher Status) is obtained when two (or more) variants of normative status have the same contents.
A simple example is of two rules, \( r1 \) and \( r2 \). This means that the rules have (1) the same addressee and (2) the same action, \( A \):

\[
\text{subject}(r1) = \text{subject}(r2) \\
\text{action}(r1) = \text{action}(r2) = A
\]

As mentioned above, the contents are the same: \( \text{contents}(r1) = \text{contents}(r2) \).

The existence (validity) of \( r1 \) and \( r2 \) is denoted respectively by \( N(A)_{r1} \) and \( N(A)_{r2} \). The addressee is bound by two duties, \( O(A)_{r1} \) and \( O(A)_{r2} \), which have the same contents.

A key question is “Can the two duties, \( O(A)_{r1} \) and \( O(A)_{r2} \), of the same addressee be put together to form a normative resultant, which is called a unified duty and denoted by \( O(A)_{e(r1-r2)} \)?” The concept is depicted in Fig. 9.

![Fig 9: The unified duty denoted \( O(A)_{e(r1-r2)} \) arises from two rules, \( r1 \) and \( r2 \)](image)

The unified duty established by \( n \) rules \( \{r1, r2, ..., rn\} \) is denoted by \( O(A)_{e(r1-r2-...-rn)} \) or simply \( O(A)_{e(1-2-...-n)} \). Each norm \( ri \) has the same condition, subject, action, etc. The construction of the unified duty brings a simplification. Separate duties and rules need not to be broken up. Instead of many duties, the sole unified duty is regarded.

An example of two rules is as follows. The first, \( r1 \), is the road rule ‘Crossing the road on a red light is prohibited.’ The second rule, \( r2 \), is implemented with a technological devise – the traffic lights.

---

6 The notation in [Lachmayer 1977, p. 88] is followed.
4.3. The Summary Status

Consider two rules, \( r_1 \) and \( r_2 \), with the same addressee, \( \text{SUB} \), but different – though compatible – obligatory behaviours, \( A \) and \( B \), respectively.

\[
\begin{align*}
\text{subject}(r_1) &= \text{subject}(r_2) = \text{SUB} \\
\text{action}(r_1) &= A \\
\text{action}(r_2) &= B \\
\text{modus}(r_1) &= \text{modus}(r_2) = \text{obligatio}
\end{align*}
\]

Though actions \( A \) and \( B \) are different, the obligatory behaviour prescribed by the first rule, \( r_1 \), is compatible with the obligatory behaviour of the second rule, \( r_2 \). On the basis of \( r_1 \), a duty \( O(A)_{r_1} \) arises. On the basis of \( r_2 \), a duty \( O(B)_{r_2} \) arises. The second rule does not cover the obligatory behaviour of the first rule. Therefore the duty to do \( A \) does not arise from \( r_2 \). Absence of this duty is denoted by \( \neg O(A)_{r_2} \).

Similar is valid for the first rule. Therefore the duty to do \( B \) does not arise from \( r_1 \). Absence of this duty is denoted by \( \neg O(B)_{r_1} \).

In case only one rule – not both – is considered, the situation is as follows. On the basis of one obligatory rule, a duty arises. Here a question can be raised “Can both cases of normative status (duty, nonexistence of duty) be summed up in a normative resultant?” Such a normative resultant which sums different normative statuses up is called a summary status (zusammenfassender Status).

In the case of two rules, \( r_1 \) and \( r_2 \), one duty and nonexistence of another duty lead to two variants. Thus two summary duties arise, actually, the first, \( O(A)_{r_1-r_2} \), concerning the first obligatory behaviour, and the second, \( O(B)_{r_1-r_2} \), concerning the second obligatory behaviour. The concept is depicted in Fig. 10.

![Fig. 10: Two cases of summary duty which results from two rules, \( r_1 \) and \( r_2 \)](image-url)
In the case of \( n \) rules, \( r_1, r_2, \ldots, r_m \), consider a summary duty \( O(X)_{r_1 \cdots r_m} \). The index shows that the obligatory behaviour, \( X \), is commanded by at least one rule, \( ri \). The index does not show how many of such \( ri \) exist. One thing is for sure – not all the rules prescribe the same \( X \). Otherwise the unified duty is obtained. It is also true that no rule commands the opposite behaviour, \( \neg X \), which is incompatible with \( X \).

The above example is built on two rules which lay down two different, though compatible, behaviours. How does the case of several rules \( r_1, r_2, \ldots, r_m \) look?

Let \( A_i \) denote the obligatory behaviour of \( ri \). Consider rules \( N(A_i)_{r_1}, \ldots, N(A_i)_{r_m} \). For each \( A_i \) of the compatible behaviours a summary duty can be established which is based on all the rules. A particular subset of rules induces a set of summary duties, which is equally large.

The construction of normative resultants is significant for the following reason. A behaviour prescribed by a separate rule is of little importance for the rule’s addressee, respectively the rule’s object. Instead, the behaviour which arises from the whole normative order is important.

Here the problem of rule contradictions can be tackled. Rules are produced by a certain subject (legislator). Therefore the rules can have contradictory contents. Science is not attributed the task to clean the contradictions. Science detects that the contradictions are available.

The task to clean contradictions can be assigned to certain authorities. Just some domains can be mentioned. Statutory rules are processed in legislative workflow. Software requirements are processed in software life cycle. Technical rules including those implemented in software are processed in software life cycle, too. Here the rules are verified, tested and validated.

A specification of a normative status synthesizer is as follows. Input is a list of rules \( r_1, r_2, \ldots, r_m \). Output is the normative resultant \( O(X)_{r_1 \cdots r_m} \). Duties and permissions are in the focus.

In reality, the machinery and synthesis task requires human intelligence. A reason is that legal rules are formulated very abstractly. In the case of a big number of rules artificial intelligence techniques cope with complexity problems. Therefore, the first attempt to identify the resulting normative status is the concept of role.

4.4. An Example of Normative Resultant: A Girl with her Father in a Café

Consider a situation – a girl with her father in a café. The girl’s normative status is represented as follows:

1. The father says “Sit down” \( \rightarrow \) Obligatio to sit = \( O(A)_{r_1} \)
2. Implicit permission to speak \( \rightarrow \) Permissio to speak = \( P(B)_{r_2} \)
3. A general prohibition to smoke in the café → Vetum to smoke \(V(C)_{r3} = O(\neg C)_{r3}\)

4. A specific prohibition to smoke for juvenile → Vetum to smoke \(V(C)_{r4} = O(\neg C)_{r4}\)

5. The father asks “What meals will you take?” → Permission for meals = \(P(D)_{r5}\)

The normative resultant can be formalised as a list of all the five consequents from above:

\[
\{ O(A)_{r1}, P(B)_{r2}, O(\neg C)_{r3}, O(\neg C)_{r4}, P(D)_{r5} \}
\]

In further simplifications, indices can be omitted as rule numbers are of little importance for the addressee (the girl). Duplicated duties are abridged, hence \(O(\neg C)\) will appear only once. The normative resultant is represented as the following list:

\[
\text{normative}_\text{resultant}_{r1-r2-r3-r4-r5} = \{ O(A), P(B), O(\neg C), P(D) \}
\] (1)

When interpreting the elements of the list, the commonsense understanding of duties and strong permissions has to be regarded. Duties are interpreted below with intersection. Therefore conjunction & is a connective in the formula:

\[
duties_{r1-r2-r3-r4-r5} = O(A) & O(\neg C)
\]

Permissions are interpreted with union. Therefore disjunction \(\lor\) is a connective:

\[
\text{permissions}_{r1-r2-r3-r4-r5} = P(B) \lor P(D)
\]

In this example, duties and permissions, \(A, B, C\) and \(D\), are viewed as being independent and no partial ordering is introduced.

Connecting the duties with conjunction and permissions with disjunction follows from deontic interpretation; see also [Sartor 2007, p. 453-497]. In the case of duties with conflicting actions, such as \(\{O(\text{goNorth}), O(\text{goSouth})\}\), the smallest duty, their intersection \(O(\text{goNorth}) & O(\text{goSouth})\), results. In the case of permissions with conflicting actions, such as \(\{P(\text{goNorth}), P(\text{goSouth})\}\), the largest permission, their union \(P(\text{goNorth}) \lor P(\text{goSouth})\), results.

Consider a one more permission for the girl – to take exclusively one soft drink:

6. The father asks “What one soft drink will you take, or cola (\(E1\)), or juice (\(E2\)), or water (\(E3\))?" → Permission for one soft drink = \(P(E1)_{r6} \xor P(E2)_{r6} \xor P(E3)_{r6}\)

The above permission is added to the normative resultant (1) above:

\[
\text{normative}_\text{resultant}_{r1-r2-r3-r4-r5-r6} = \\
\{ O(A), P(B), O(\neg C), P(D), ( P(E1) \xor P(E2) \xor P(E3) ) \}
\]

The normative resultant of the six rules \(r1,...,r6\) comprises permissions represented by the following formula:
The interplay of obligations and strong permissions is studied by Boella and van der Torre (2008). They provide more elaborate examples in the context of hierarchies of authorities. They use Makinson’s and van der Torre’s framework of input/output logic [Makinson & van der Torre 2000]. The “Rex, Minister and Subject” game devised by Bulygin (1986) serves as an introduction.

4.5. Incompatible Rules Result with ‘Impossible’ Summary Obligation

In the case of incompatible obligatory actions, ‘impossible’ summary duty results. In logic it is denoted by false, \( \bot \) (bottom), or \( \emptyset \) (empty). To illustrate consider a situation with two valid but incompatible rules:

\[
\text{text}(r1) = \text{‘Save the mother’,} \quad \text{text}(r2) = \text{‘Save the child’}
\]

Conflicting actions mean an empty intersection \( A \cap B = \emptyset \). Formally, the set intersection of the operational implementations of the actions \( A \) and \( B \) is empty:

\[
\text{operationalisations}(A) \cap \text{operationalisations}(B) = \emptyset
\]

Another example of hardly compatible rules is when the father says go (\( r1 \)) and the mother says stay (\( r2 \)). The resultant is a contradiction, go & \( \neg \)go. It is worth to note that the real world pragmatics forces the addressee (child) to sum the arising duties up to a middle behaviour like staying in the yard. Hence, other criteria sway which are not explicit in the rules, but implicit in the context.

4.6. Synthesizing Normative Status

The problem of (semi-automatic) synthesis of normative status is formulated below. A specification of a normative status synthesizer can be formulated rather simply. Input is a list of rules, \( r1, r2, \ldots, rn \). Output is the normative resultant, \( O(X)_{r1,r2,\ldots,rn} \); see [Lachmayer 1977, p. 88-92]. This is depicted in Fig. 11.

```
N(A1)  r1  N(A1)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
N(A2)  r2  N(A2)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
N(A3)  r3  N(A3)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
N(An)  rn  N(An)
|  |   |  |
N(A1)  r1  N(A1)
|  |   |  |
N(A2)  r2  N(A2)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
N(A3)  r3  N(A3)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
N(An)  rn  N(An)
```

Fig. 11: The input and output of a synthesizer of the normative status. The rules \( r1, r2, \ldots, rn \) serve as input. Output is the normative resultant of the rules.

Implementation of the machinery is not an easy task. Synthesis of both duties and permissions is in the scope. In reality, the task requires human intelligence.

\[
\text{permissions}_{r1,r2,r3,r4,r5,r6} = P(B) \lor P(D) \lor (P(E1) \oplus P(E2) \oplus P(E3))
\]
reason is that legal rules are usually formulated very abstractly. In the case of a big number of rules artificial intelligence faces complexity problems. Therefore, an attempt of identifying the resulting normative status is to assign a role, such as a passenger, a teacher, etc.

5. Related Work

**Multiagent Systems.** Relevant formalisations are investigated by JURIX, DEON and Normative Multiagent Systems communities. Their formalisms are mathematically more precise than visualizations in this paper. Boella et al. (2009) introduce five guidelines for the use of normative systems in computer science; see also a whole volume devoted to agents and AI & Law. Boella et al. (2010) concern violations of legal rules and address goal-based interpretation and conflicting goals. Dignum et al. (2002) seek a long-term goal to extend the Belief – Desire – Intention (BDI) model, at a theoretical and practical level, with social concepts such as norms and obligations.

**The obligation concept.** With regard to the analysis of obligations, we have found useful the thesis of Adam Wyner (2008). He considers deontic logic as presented by Paul McNamara (2006) and provides linguistic considerations and natural language syntactic and semantic analysis. Wyner’s analysis of the Contrary-to-Duty Paradox (the Gentle Murder Paradox and the Good Samaritan Paradox) contributes to better understanding of obligations. The reader will understand the lexical semantics of obligations in case he finds mathematical axioms of deontic logic too heavy.

**Virtual Institutions.** Anton Bogdanovych’ (2007) work on non-game virtual worlds is very relevant. He developed the concept of Virtual Institutions (VI) “being a new class of Normative Virtual Worlds, that combine the strengths of 3D Virtual Worlds and Normative Multiagent Systems, in particular, Electronic Institutions [Esteva 2003].” Virtual Institutions are targeted at e-commerce. In Virtual Institutions an avatar is treated as an agent. The avatar can stop for a while interacting with its human master and continue independently as an agent that is governed by rules. Following is an example a rule: “Only participants with certain roles are permitted to enter particular rooms.” For instance, a Buyer is permitted to an Auction room but a Seller is prohibited.

Esteva (2003) sees Electronic Institutions as structures and mechanisms of social order and cooperation governing the behaviour of two or more individuals.

The ancient city of Uruk project [Bogdanovych et al. 2009] is interesting for the following reasons. Its specification is rather simple and consists only of a dozen abstract norms.

Virtual Institutions are defined as “3D virtual worlds with normative regulation of interactions” [Bogdanovych et al. 2007, p. 69]. A virtual institution is enabled by a three layered architecture. The Normative Control Layer employs AMELI to
regulate the interactions between participants by enforcing the institutional rules. The Communication Layer causally connects the institution dimensions with the virtual world. The Visual Interaction Layer (supported by Second Life) visualizes the virtual world. Electronic Institutions technology brings the set of tools for formal specification of the institutional rules, formal verification of those and a system that controls the enforcement of these rules [Bogdanovych et al. 2009].

**Law-Governed Interaction.** The concept of ‘law’ in VirtualLife differs from that in Law-Governed Interaction (LGI); see Naftaly Minsky (2005). LGI is a decentralized coordination and control mechanism for distributed systems – it can also be viewed as a “policy mechanism”. The term “policy” is used for an informal statement of a set of rules-of-engagement, reserving the term “law” for a formalisation of this policy in a manner that can be enforced via LGI [Minsky 2007, p. 2]. LGI identifies two modes of law enforcement:

(a) by preventing violations of the given law;
(b) by reacting to violations.

LGI adopts the former stronger mode of law enforcement, (a). VirtualLife tackles both: the Virtual Constitution concept introduces the ‘soft’ interpretation of rules; thus (a) is tackled. VirtualLife adopts (b) by reacting to violations. This is a case of the enforcement of rules which are created with the rule editor. Thus the ‘strong’ interpretation of rules is adopted. Mode (a) is difficult to implement as punishment cannot be implemented in full scale in VirtualLife; mode (b) is easier.

The role of a law \( L \) under LGI is to decide what should be done if a given event \( e \) occurs at an agent \( x \) operating under this law, when the control-state of \( x \) is \( s \). This decision, the ruling of the law, can be represented by the sequence of primitive operations mandated by the law, to be carried out, automatically, at \( x \). More formally, \( L \) is a function \( L: E \times S \rightarrow O^* \), where \( E \) is a set of all possible regulated-events, \( S \) is a set of all possible states, and \( O \) is a set of all primitive operations [p. 8]. The event-condition-action rules have the form UPON event IF condition DO operation-list. The rules are implemented in Prolog or Java. Thus, LGI rules are implemented as technical rules.

**Computer code is law.** The role of computer code in cyberspace can be compared with laws in real space.\(^7\) In particular, MMOGs (massively multiple online games) are also “determined by the code – the software, or architecture, that makes the MMOG space what it is” [Lessig 2006, p. 14]. Therefore the issues of the conception “code is law” are important. Software engineers have to regard limits of programming in regulation by code. Jurists have to regard limits of territorially based sovereigns.

\(^7\)“In real space, we recognize how laws regulate – through constitutions, statutes, and other legal codes. In cyberspace we must understand how a different “code” regulates – how the software and hardware (i.e., the “code” of cyberspace) that make cyberspace what it is also regulate cyberspace as it is.” [Lessig 2006, p. 5]
6. Conclusions

Virtual worlds are part of the real world. However legal and security virtual features need to be improved in order to guarantee safe and reliable virtual infrastructure. The paper reflects on this need and elaborates on possible solutions, including a legal framework. Formalisation issues are in the focus. Practical issues of virtual world law tackle governance in cyberspace and the concept of e-person, cf. [Schweighofer 2007].

In the approach worded “From rules in law to rules in artefact”, we elaborate on an issue of regulation by computer code. The artefacts can be of different kinds. Additionally to embedded systems, the artefacts comprise software systems like virtual world platforms and virtual nations. Hence, the approach builds a bridge between law and informatics.

We aim to build a bridge between an Ought-to-Be reality and Is reality in the domain of virtual worlds. Here software developers identify the rules and implement them in computer code. A part of rules can be explicitly represented – the ‘strong’ interpretation [Boella et al. 2007]. A part of rules cannot be represented explicitly. They are formulated explicitly in the system specification, namely, in the EULA text – the ‘weak’ interpretation.

Modelling virtual worlds and implementing rules is a challenging task; not only for software developers. Therefore, the people think in roles, not rules. The names of roles, such as an author, an owner, a NoCopy nation, serve to describe situations. Formalisations and operational implementation require further research. This paper attempts to formalise normative status and thus contributes to structuring legal semantics.

Law enforcement in virtual worlds remains a tough problem. Legal rules are unfeasible to be programmed in the full extent in software.

We claim that the rule of law is required for both human and artificial agent communities. Different persons can pursue conflicting interests. Therefore the need arises for governance by rules. Operating systems perform the role of “arbiters” in communities of computer programs that compete for common resources. Similarly, the behaviour of agents (avatars) in virtual worlds has to be regulated by rules that constitute the law of the virtual world.

7. References


