

Citation for published version:

Balke, T, De Vos, M, Padget, J & Fitzek, F 2011, Using a normative framework to explore the prototyping of wireless grids. in *Coordination, Organizations, Institutions, and Norms in Agent Systems VI - COIN 2010 International Workshops, COIN@MALLOW 2010, Revised Selected Papers*. vol. 6541 LNAI, Lecture Notes in Computer Science, Springer, Heidelberg, pp. 95-113, 6th International Workshops on Coordination, Organizations, Institutions, and Norms in Agent Systems VI, COIN@MALLOW 2010, August 30, 2010 - August 30, 2010, Lyon, France, 1/01/11. https://doi.org/10.1007/978-3-642-21268-0_6

DOI:

[10.1007/978-3-642-21268-0_6](https://doi.org/10.1007/978-3-642-21268-0_6)

Publication date:

2011

Document Version

Early version, also known as pre-print

[Link to publication](#)

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Using A Normative Framework to Explore the Prototyping of Wireless Grids

Tina Balke
Chair of Information Systems
Management
University of Bayreuth
tina.balke@uni-
bayreuth.de

Marina De Vos,
Julian Padget
Dept. of Computer Science
University of Bath
{mdv,jap}@cs.bath.ac.uk

Frank Fitzek
Multimedia Information and
Signal Processing
University of Aalborg
ff@es.aau.dk

ABSTRACT

The capacity for normative frameworks to capture the essential features of interactions between components in open architectures suggests they might also be of assistance in an early, rapid prototyping phase of system development, helping to refine concepts, identify actors, explore policies and evaluate feasibility. As an exercise to examine this thesis, we investigate the concept of the wireless grid. Wireless grids have been proposed to address the energy issues arising from a new generation of mobile phones, the idea being that local communication with other mobile phones, being cheaper, can be used in combination with network communication to achieve common goals while at the same time extending the battery duty cycle. This results in a social dilemma, as it is advantageous for rational users to benefit from the energy savings without any contributing to the cooperation, as every commitment has its price. We present a necessarily simplified model, whose purpose is to provide us with the foundation to explore issues in the management of such a framework, policies to encourage collaborative behaviour and the means to evaluate the effects on energy consumption.

Categories and Subject Descriptors

I.2.4 [Computing Methodologies]: Artificial Intelligence—*Knowledge Representation Formalisms and Methods, Distributed Artificial Intelligence*

Keywords

Wireless Grids, Answer Set Programming, Norms, Reciprocity

1. INTRODUCTION

This paper reports on a feasibility study into how and whether institutional models can help in evaluating the concept of wireless grids. While that is the specific topic of the paper, the broader contribution is that of asking the question of how such normative model building can be of use in an early design phase, long before hardware or software is available, in order to evaluate both principles and alternative policies — that might have significant consequences subsequently.

In technology neutral terms, the problem we consider is of some digital content to be distributed to a collection of nodes that support an expensive (in terms of power and money) connection via a structured network and a cheaper connection via an ad-hoc network. The task is to minimise the cost of the distribution of this digital content by using a combination of the structured and ad-hoc networks. The model can essentially be parameterised by the cost functions for the (un)structured network technology. The particular

case that interests us is the forthcoming 4G mobile phone network where the structured network uses a traditional cellular link and the ad-hoc network uses IEEE 802.11 (wireless LAN) with the ethernet transport protocol. The motivation for the idea of such a “wireless grid” is that local communication over (wireless) ethernet uses significantly less power per unit of data than communicating with the network base-station and that duration of the battery duty cycle is a major usability factor for users.

The deployment of third generation (3G) of mobile network systems is in progress, but a quite different next generation network (called Fourth Generation or 4G) is under development that is intended to cause a paradigm shift in the cooperation architecture of wireless communication [14]. While for 3G the industry focused on technology for enabling voice and basic data communications (technology-centric-view), the emphasis in 4G is more user-centric [24]. Consequently, studies to find possible drivers for consumer demand for mobile devices, such as the one by TNS [21] across 15 countries in mid-2004, have been conducted. This study revealed that it was not high performance that was attractive to consumers, but rather useful, convenient and enjoyable services coupled with ubiquitous infrastructures for constant connection. In addition, “two days of battery life during active use” topped the wish list of key features in 14 of the 15 countries surveyed.

Batteries have fixed capacity that puts limits on the operational time for a device in one charge cycle. The increasing sophistication of mobile phones and their evolution into smart phones offering Internet access, imaging (still and video), audio and access to new services, has had a significant impact on power consumption, leading to shorter stand-by times, as well as the problem of rising battery temperature unless there is active cooling [19].

Fitzek and Katz [9] have proposed a way around some of these issues with the concept of a wireless grid, in which users share resources in a peer-to-peer fashion that uses less power but this requires a difficult to obtain collaboration between the users. The contribution of this paper is to build an institutional model of the interactions between handsets and base-station and between handsets in order to provide a foundational model from which to be able to explore policies, identify suitable sanctions and evaluate potential gains from reduced power consumption.

The remainder of the paper is structured as follows: in the next section (2) we cover three aspects of the background, namely (i) normative frameworks, (ii) a detailed discussion of the wireless grid scenario, and (iii) the energy model: what different agent actions cost in terms of power consumption. Then, in section 3 we describe the action model—what the agents may do—before presenting some results from its analysis. We conclude in section 4 with a discussion of the related work, results and future directions.

2. TECHNICAL CONTEXT

The first section here serves to provide a brief description of the event-based normative framework that is used later for the model. The second provides a detailed description of some technical issues surrounding the wireless grid idea, highlighting in particular, actual energy costs and the risk of free-loading, which latter has some elements that echo issues with common-pool resource problems.

2.1 Normative Frameworks

The concept of the normative framework—sometimes also called an institution, sometimes a virtual organisation—has become firmly embedded in the agent community as a necessary foil to the essential autonomy of agents, in just the same way as societal conventions and legal frameworks have been developed to constrain people. In both the physical and the virtual worlds—and the emerging combination of the two—the arguments in favour centre on the minimization of disruptive behaviour and supporting the achievement of the goals for which the normative framework has been conceived and thus also the motivation for submission to its governance by the participants.

While the concept remains attractive, its realization in a computational setting remains a subject for research, with a wide range of logics [1, 4, 6] and tools [20, 22, 12], to cite but a few. We do not include an extensive and detailed case for the purpose and value of normative frameworks here—this can be found in [23, 5], for example.

2.1.1 Formal Model

To provide context for this paper, we give an outline of a formal event-based model for the specification of normative frameworks that captures all the essential properties, namely empowerment, permission, obligation and violation. Extended presentations can be found in the citations above.

The essential elements of our normative framework are:

1. Events ($\mathcal{E}$), that bring about changes in state, and
2. Fluents ($\mathcal{F}$), that characterise the state at a given instant.

The function of the framework is to define the interplay between these concepts over time, in order to capture the evolution of a particular framework through the interaction of its participants. We distinguish two kinds of event: normative events ($\mathcal{E}_{inst}$), that are the events defined by the framework and exogenous ($\mathcal{E}_{ex}$), that are outside its scope, but whose occurrence triggers normative events in a direct reflection of the “counts-as” principle [13]. We further partition normative events into normative actions ($\mathcal{E}_{act}$) that denote changes in normative state and violation events ($\mathcal{E}_{viol}$), that signal the occurrence of violations. Violations may arise either from explicit generation, from the occurrence of a non-permitted event, or from the failure to fulfil an obligation. We also distinguish two kinds of fluents: *normative fluents* that denote normative properties of the state such as permissions, powers and obligations, and *domain fluents* that correspond to properties specific to the normative framework itself.

The evolution of the state of the framework is achieved through the definition of two relations:

1. The generation relation: this implements counts-as, in that it specifies how the occurrence of one (exogenous or normative) event generates another (normative) event, subject to the empowerment of the actor. Formally, this can be expressed as $\mathcal{G} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{E}_{inst}}$, where $\mathcal{X}$ denotes a formula over the (normative) state and $\mathcal{E}$ an event, whose confluence results in an institutional event, and

2. The consequence relation, that specifies the initiation and termination of fluents subject to the performance of some action in a state matching some expression, or formally $\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}} \times 2^{\mathcal{F}}$.

Again, for the sake of context, we summarize the semantics of our framework and cite [7] for an in-depth discussion. The semantics are defined over a sequence, called a trace, of exogenous events. Starting from the initial state, each exogenous event is responsible for a state change, through initiation and termination of fluents, that is achieved by a three-step process:

1. The transitive closure of $\mathcal{G}$ with respect to a given exogenous event determines all the (normative) events that result
2. To this we add all violations of events not permitted and all obligations not fulfilled, giving the set of all events whose consequences determine the new state, so that
3. The application of $\mathcal{C}$ to this set of events, identifies all fluents to initiate and terminate with respect to the current state in order to obtain the next state.

So for each trace, we can obtain a sequence of states that constitutes the model of the normative framework. As with human regulatory settings, normative frameworks become useful when it is possible to *verify* that particular properties are satisfied for all possible scenarios. In order to do so, we need to incorporate a computational model in our formal representation.

2.1.2 Implementation

This formalisation is realized as a computational model through Answer Set Programming [3, 11] and it is this representation that is the subject of the evaluation process described in Section 3.2. In [7] it was shown that the formal model of an normative framework could be translated to an *AnsProlog* program—a logic program under answer set semantics—such that the answer sets of the program correspond to the traces of the framework. A detailed description of the mapping can be found there.

AnsProlog is a declarative knowledge representation language that allows the programmer to describe a problem and the requirements on the solutions. Answer set solvers like CLASP [10] or SMOLENS [16] process the *AnsProlog* specification and return the solutions, in this case the traces, as answer sets. Answer set programming, a logic programming paradigm, permits, in contrast to related techniques like the event calculus [15] and C+ [8], the specification of both problem and query as an executable program, thus eliminating the gap between specification and verification language. But perhaps more importantly, both languages are identical, allowing for more straightforward verification and validation.

A level of abstraction can be added using a domain-specific action, InstAL [7], and query language, InstQL [12], which can be both translated into *AnsProlog* in order to specify not only the valid traces, but those that exhibit features of interest. We use InstAL to describe our scenario in Section 3. The action language uses semi-natural language to describe the various components of the normative framework and allows type definitions to avoid grounding problems when translating to *AnsProlog*. For example, events are defined by `typeOfEvent event nameOfEvent;` with type being one of exogenous, create, inst or violation, while fluents are defined by `fluent nameOfFluent(ParameterType, ...);`. Generation of normative events from exogenous events is specified using the `generates` statement, while `initiates` and `terminates` define the two parts of the consequence relation. Conditions on the state are expressed using `if`. The `initially` statement serves to specify the set of fluents that characterise the initial

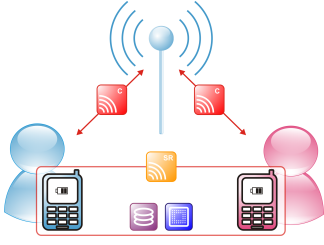


Figure 1: Wireless Grid Communication Architecture

state after the normative framework is created. For our model we are interested in all traces that lead to success, so we do not require the additional facilities of the query language InstQL. Instead we specify the fluents or events we want to show or hide directly in *AnsProlog* using the directives `#show` and `#hide`.

2.2 The Wireless Grid Scenario

2.2.1 The Wireless Grid Architecture

As described in the introduction, to overcome the energy problems of 4th generation mobile phones, Fitzek and Katz [9] proposed the establishment of *wireless grids* as shown in Figure 1 [9].

In these wireless grids, ubiquitous mobile devices with potentially different capabilities are expected to create ad-hoc connections and to cooperate and share their limited resources for the benefit of the community. Cooperation between mobile devices is achieved by short range communication link technologies, such as WLAN or Bluetooth. Compared to the traditional cellular 3G communication with the base-station, the advantage of the short-range communication is much higher bandwidth while using much less power, which we quantify in section 2.2.2. Thus, the battery and CPU power needed on the short link is significantly lower than it would be needed on the cellular one [19]. In this paper we will focus on the IEEE802.11 WLAN specification, that allows mobile devices to communicate directly with each other and according to Perrucci et al. [19] has the highest energy saving potential.

For a better understanding of the wireless grid idea we briefly present a scenario that we can refer back to later. This scenario is set in a football stadium: while watching one game, the fans are very likely to be interested in games that take place at the same time at another place. As they cannot watch two games live at the same time, they might use mobile phones in order to get information about other games. A likely problem for the infrastructure provider is that once a goal has been scored in another game, fans want to watch the other goal on their mobile phones and all try to stream the video file from the base station at the same time, thus overloading it. The bandwidth of the base station connection is divided into several channels that are sent out sequentially within one time frame. Thereby — up to a certain technical maximum — each mobile phone is allocated one slot. As the total bandwidth of a base station is fixed, the more mobile phone users are given a slot, the smaller the bandwidth that can be assigned to each channel gets. As a result the download times increase, leading both to more battery consumption and lower quality in the streaming service.

In contrast to the normal “non-cooperative” scenario in which a single mobile phone user would need to receive all sub-streams over the cellular link resulting in the above mentioned problems, using the cooperation envisioned in the wireless grid scenario, users could share the task by receiving a subset of the multicast channels over the cellular link from the base station and exchanging the missing pieces over the short range link.

2.2.2 The Energy Advantage in IEEE802.11

To understand the IEEE802.11 WLAN wireless grid scenario and its energy implications better, this section examines the technical aspects of WLAN transmission in more detail. We use A to denote the set of agents in the scenario. In considering the energy implications of the wireless grid scenario, we observe the following basic definition of energy $[E]$, that states that energy consumption in terms of battery depends on two factors: the power $[P]$ consumed per connection type and the time $[t]$ needed for the actual transmission:

$$Energy = Power * Time [Joules] \quad (1)$$

So what is the energy consumption in this scenario? The total energy consumption is the energy consumed over the traditional cellular 3G connection (E_{3G}) plus that over the short link (i.e. WLAN) connection (E_{WLAN}) plus as the idle time for both links (E_{idle}). In case of no cooperation the latter costs 0, i.e. it is assumed that the WLAN connection is turned off and the football fan has to stream the complete video using the 3G connection. In case of wireless grid cooperation it is assumed that both connections (WLAN and 3G) are turned on and the devices help one another in a peer-to-peer-like fashion. Assuming $|A_{Coop}|$ cooperating agents in the scenarios for example, each agent only needs to stream only a part of the total video from the base station (i.e. $\frac{1}{|A_{Coop}|}$ in an ideal scenario) and obtain the missing chunks from the other cooperation partners using the short link connection. Therefore the energy consumption in the cooperation case (E_{Coop}) comprises that consumed for:

1. **Streaming** part of the video from the base station using the 3G link ($E_{3G,rx}$) (plus the energy consumed while the 3G connection is idle ($E_{3G,i}$)),
2. **Receiving** the remaining chunks of the video on the WLAN connection ($E_{WLAN,rx}$),
3. **Sending** the own chunks to the other participants via the WLAN connection ($E_{WLAN,tx}$), and
4. **Idling** (i.e. when not transmitting or receiving anything but waiting for the next interaction) ($E_{WLAN,i}$).

With reference to equation 1, by replacing the E with the respective $P * t$ -values, one can analyse the power consumption as well as the transmission times for the scenario in the cooperative and non-cooperative case in detail. Representative power and time values for the transmission in the different states using 3G and WLAN connection can be found in [19, p.D10] for example, which are based on measurements from a Nokia N95. These numbers indicate that although the power needed for the WLAN and the 3G state are about the same, for a point-to-point communication, the data rate for the 3G link (0.193 Mbit/s for the receiving state) is significantly lower than that of WLAN (5.115 Mbit/s, receiving state, 30m distance) leading to significantly worse transmission times and consequently a much worse energy per bit ratio for the 3G link. The energy consumed in the idle states is of secondary importance and therefore neglected here.

This suggests that the cooperation scenario has a significant potential advantage in energy consumption, compared to the conventional cellular communication architecture, especially if the number of cooperating mobile phones is high and a large proportion of the data transmission can be done via the short-link connection.

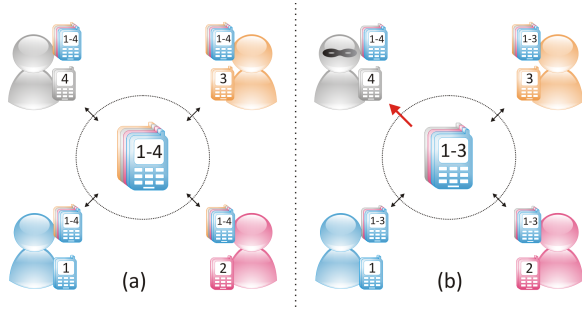


Figure 2: The Reciprocity Problem in Wireless Grids

2.2.3 The Reciprocity Problem in Wireless Grids

Although the wireless grid may have a huge advantage with regard to the battery consumption, it also has the intrinsic weakness of distributed cooperative architectures: it relies on cooperation to succeed. The cooperation idea in the wireless grid, as shown in figure 2(a), is as follows:

1. The participants volunteer their resources, forming a common pool which can be used by all of them in order to achieve a common goal, such as file streaming. The utility which users can obtain from the pooled resources is much higher than they can obtain on their own. For example, in the football stadium scenario, both download time and battery consumption are reduced. However, the problem is that commitment comes at a cost, in the form of battery consumption for sending file chunks, i.e. $E_{WLAN,tx}$. As a consequence, (bounded) rational users would prefer to access the resources without any commitment of their own, as shown in figure 2.
2. Thus, as shown in (b), the grey agent in the top left corner (with blindfold) can enjoy the full benefits from the common pool without committing anything itself, hence cheating on the three other agents.

However, if a substantial number of users follows this selfish strategy, the network itself would be at stake, depriving all users of the benefits [17]. The reason for this is straightforward: network users can have strategic behaviour and are not necessarily obediently cooperating by making their resources available without the prospect of rewards for their good behaviour. Unreciprocated, there is no inherent value to cooperation for a user. A lone cooperating user draws no benefit from its cooperation, even if the rest of the network does. Guaranteed cost paired with uncertainty or even lack of any resulting benefit does not induce cooperation in a (bounded) rational, utility-maximising user. Without any further incentives, rational users therefore would not cooperate in such an environment and all be worse off than if they cooperated [2].

2.2.4 The Energy Model

Utility quantification is being used by the (bounded rational) agents (i.e. agents that only have partial information about their environment, including other agents) to determine the utility of the different possible actions and choose their actions in such a way that maximises their utility. Concerning the knowledge that they can rely on when calculating utilities, we assume the agents not to have knowledge of the whole system, but only the small part of it in their vicinity.

We now explain how the agents determine the utility of an action, using the football stadium scenario described earlier. However, to

keep the example simple, for the utility considerations we consider the interaction of two agents only and formulate the costs in such a way that they can easily be expanded to any number of agents.

The two agents both want to stream the same file G in the stadium. In order to get the complete file, they can cooperate and thereby reduce their energy consumption or stream the file themselves using a cellular link connection. The exchange is done in chunks ($g \in G$).

As described above, the issue in the particular wireless grid scenario that we consider here is that the different agents have different subsets of G (i.e. parts of the file) already and each is trying to obtain the full set by exchanging parts of their subsets of G with one another. Thus, looking at a potential exchange, from the perspective of an agent a_i , for each chunk only two mutually exclusive situations can occur: either the agent does, or does not, have a given chunk. This can be expressed in terms of the set H_{a_i} (the set of chunks agent a_i has; $H_{a_i} \subseteq G$) and the corresponding complement set (with respect to G) H'_{a_i} that represents the set of chunks agent a_i has not.

In an exchange, an agent a_1 will try to obtain the set of the missing chunks H'_1 and in turn can potentially provide the set H_1 . Let H_2 being the chunks agent a_2 possesses and let agent a_1 and a_2 enter an exchange process ($H_1 \cup H_2 \subseteq G$). In order to reflect the local connectivity properties, we write $\bar{A}_{a_1} (\subseteq A)$ to denote those $a_j \in A, j \neq i$ that are within communication range of a_i . The local radius of each agent is determined by the transaction protocol dependent signal radius of its mobile phone.

What is important to the agent now are the utilities of the different action alternatives. Thus, an agent needs to consider the utility of using the short-link cooperation (including the costs for searching short-link cooperation partners in the first place) compared to the cellular link as well as the utility of reciprocating in contrast to cheating on other agents.

The search costs are those that accumulate as a result of the agents searching for the missing chunks. We assume that the costs of sending out a request message (RM) for cooperation using WLAN transmission are fixed and independent of the number of chunks requested. However, the number of messages an agent has to send before it finds an agent that is willing to cooperate and one that can supply at least one missing chunks depends on the success probability $p = f(|\bar{A}|, H'); p \in [0, 1]$ for a single message. We define “success” to mean finding a cooperation partner with at least one missing chunk. As stated above, the probability p is a result of the function of the number of agents in the neighbourhood $|\bar{A}|$ and of the number of chunks missing H' . As yet, we have no measure of how these two quantities are related, but we can make some general observations about their correlation. Thus, for the missing chunks, we contend, without evidence at this point, that p has a proportional relation with the missing chunks of the form $H'_{a_i} \propto p$. Our rationale starts from the assumption that the chunks are distributed uniformly over all agents. Thus, if missing many chunks an agent is more likely to find another agent that can offer any of the missing chunks, whereas the probability is lower if it is only missing a small number of specific chunks. Besides the number of missing chunks, p is furthermore dependent on the number of agents in the neighbourhood, i.e. the number of other agents $|\bar{A}_{a_i}|$ an agent a_i can see locally¹. The probability p is proportional to $|\bar{A}|$ as well. The intuition is that the higher the agent population density, the higher the probability of finding an agents that responds positively to the request when searching for the chunks.

¹For reasons of simplicity it is assumed that the number of agents in the neighbourhood has no volatility, but remains the same throughout the process.

To give an example for p , in a football-stadium where many people are in one place and want to download the same file (e.g. a replay of a goal), it will approach 1 as there are many people searching for and offering the same chunks, while it tends to approach 0 when there are fewer people searching for and offering the same chunks. Once an agent has found a transaction partner, they can exchange chunks. Thus the maximum number of chunks available for exchange is the intersection of the set an agent can offer to the transaction partner (i.e. all the chunks it has) and that the transaction partner needs; and vice versa, i.e. $H_1 \cap H'_2$ & $H_2 \cap H'_1$.

Returning to the example, in the course of the exchange both agents have the option to cooperate (i.e. deliver what they promised) or defect and not send their chunks. As a consequence of this, two different utility situations can occur. Thus, in the cooperation case, based on opportunity cost considerations, the utility is calculated by taking into account what it would have cost for an agent to download the chunks from the base station using the 3G connection ($E_{3G,rx}$) reduced by the costs of receiving the chunks on a short range WLAN link from another agent ($E_{WLAN,rx}$) minus the costs for sending its own chunks ($E_{WLAN,tx}$). The latter cost can be saved by the agent if it defects. However, assuming that the transaction partner stops the transaction if being cheated and no further chunks are exchanged (tit-for-tat), in this case the agent will have search for a new transaction partner for the remaining missing chunks. This results in search costs that could otherwise have been saved. The specific energy cost $E_{a,b}$ where $a \in WLAN, 3G; b \in tx, rx, idle$ have already been determined by Perrucci et al. [18] for single bits. As a first approximation, using a constant bpc (i.e. bits per chunk) these could be mapped to the chunks in the model.

Using the bpc mapping and the figures by Perrucci et al. and substituting them with the variables of our utility considerations an agent is able to compute an utility for all the actions available and decide on the action to take as a consequence.

3. FORMALIZING THE WIRELESS GRID SCENARIO

Now that we have explained the wireless grid scenario in some detail from the technological perspective, we now shift focus to the normative framework.

We observe three perspectives to the wireless grid scenario:

1. The actions that agents may take, as prescribed by the normative framework,
2. The utility functions that quantify battery costs for a given action, and
3. The agents that populate the normative frameworks and choose which action to take, informed by the utility functions.

In this paper, our focus is on the (normative) actions and the utility functions (see section 2.2.4): we will address their integration through the agents that participate in the normative framework in future work.

3.1 The Normative Framework

The model is preliminary in that it focusses on the essential interactions and the communication costs that arise from those interactions. Although a more elaborate model is desirable from a realistic point of view, more details would also distract and complicate while not adding to the presentation.

The features of the the prototypical scenario are:

- $1 \times$ base-station: B

- $m \times$ agents: $A = \{a_1, \dots, a_m\}$
- $1 \times$ digital good: G divided into
- $n \times$ chunks: $\{g_1, \dots, g_n\}$

We further assume that $n|m$, which is to say the number of chunks is a multiple of the number of agents.

3.1.1 Negotiation, obtaining and sharing

We identify three phases to the interactions for handset to base-station and handset to handset:

- **Negotiation:** assign g_i to a_j s.t. $f : G \rightarrow A$ and

$$f^{-1} : A \rightarrow G^{m|m} \text{ s.t. } f^{-1}(a_i) = \{g_j, f(g_j) = a_i\}$$

- **Obtaining:** agent a_i receives chunks $f^{-1}(a_i)$ from B
- **Sharing:** agent a_i sends chunks $f^{-1}(a_i)$ to and receives chunks $G \setminus f^{-1}(a_i)$ from other agents.

These three phases are distinct, but although negotiation must come first, obtaining and sharing can be interleaved as soon as downloading has commenced. In the following paragraphs we discuss each phase in more detail and how each is encoded in InstAL.

Each InstAL specification starts with the identification of the normative framework, the different types of variables it will use (their values can be specified in a domain file) and the fluents and events it will recognise. The full definition can be seen in Figure 3. The meaning of the various elements is explained as we progress through the different phases.

Negotiation Phase:

We are not particularly concerned with the technicalities of the negotiation phase—any off-the-shelf protocol could be employed—as long as the post-condition is satisfied: that each chunk is assigned to exactly one agent and that each agent is assigned the same number of chunks—although these conditions can readily be relaxed at the cost of a lengthier specification. An allocation satisfying these conditions is given in the initial state of the model (see Figure 6, lines 104–105) via the `obtainChunk` fluents indicating which agents are tasked with obtaining which blocks from the base-station. Together with their chunk assignment the agents receive the necessary permission to do so (lines 102–103).

Obtaining Phase:

This is where each agent downloads its assigned chunks from the base-station. This process should result in each agent holding $n|m$ distinct chunks. Because the base-station uses several different frequencies (frequency division multiplexing), many agents may download chunks simultaneously. We refer to a frequency division in the model as a channel. Of course, there is a physical limit to the number of frequency divisions and hence the number of simultaneous agent connections. The full specification of this phase can be seen in Figure 4. Each agent can only physically obtain one chunk at a time from the base station, while each channel can only be used to obtain one chunk. This is modelled by the fluent `cbusy`. The first InstAL rule (lines 34–36) indicates that a request to obtain a chunk is granted (`intObtain`) whenever there is an available channel and the agent is not busy obtaining another chunk. When a block is obtained the agent and the channel will become busy for a fixed amount of time — 2 time steps in this case (lines 42–43). From the first instant of the agent interacting with the base station, it is deemed to have obtained the block, so parts

can be shared (line 41). As soon as a channel and an agent become engaged, the framework takes away the power from the agent and from the channel to engage in any other interactions (lines 53–54), stops the agent from needing the chunk and cancels the permission to obtain the chunk again later on (lines 55 and 56, respectively).

Each exogenous event generates a transition to mark the passing of time (lines 38–39). The `clock` event indicates that no agent was interacting with the normative framework. The transition event reduces the duration of the interaction between the channel and agent (line 46). When the interaction comes to an end, transition restores the power for agents to obtain chunks via the channel and for the agent to obtain more chunks (lines 48–51). The event also terminates any busy fluents that are no longer needed (line 58).

Sharing Phase:.

In this phase each agent shares its chunks with another agent, with the goal that at the end of the process, each agent has a complete set of the chunks. The full specification can be found in Figure 5. The principle here is more or less the same as with obtaining blocks, only that we build in a mechanism to encourage agents to share their chunks with others rather than just downloading them. To be able to monitor the different costs of obtaining a chunk from the base-station or from a peer, we introduced the fluent `abusy`. When a chunk is downloaded from a peer, the agent loses permission to download another chunk until it has shared a chunk with another agent (lines 85 and 73 respectively). Continuous downloading without sharing (no permission is granted to download) results in a violation event named `misuse` (line 70). The penalty we chose to implement in our model is that the violation agent loses the power to `intDownload` (Line 91), which means that for all intents and purposes it has been expelled from the peer group. Initially, agents are given the permission and power to download one chunk (Figure 6 lines 112–114).

Figures 3 to 6 give the complete characterisation of our wireless grid scenario. When translated to *AnsProlog* and combined with the non-framework-dependent program components, we obtain all the possible traces over a specified number of time instances. A successful trace makes sure that at the end all agents have all chunks and are no longer engaged. Figure 7 shows a graphical representation of a successful trace for a scenario with two agents (bob and alice), four chunks (x1, x2, x3 and x4) and a base-station with two channels (c1 and c2). The circles indicate the time steps. Light grey fill means the device is `cbusy` while dark grey indicates `abusy`. The arrows indicate which block goes to which agent. The labels on the left-hand side indicate the exogenous event and the current distribution of chunks. The observed event `clock` is not shown to avoid cluttering the diagram.

3.1.2 Sanctioning

The model as presented in Figure 5 takes a rather harsh position on sanctioning, in that the violating agent is expelled—the power to get chunks from other agents is rescinded. In fact, this is both harsh and counter-productive, because given the initial state shown in Figure 6, the chunk assignment is not 1-resilient—meaning the distribution cannot be achieved following the expulsion of one agent, unless in the very special case where the expulsion occurs after the other agent no longer requires any chunks from this agent. Full 1-resilient assignment can be achieved with two chunks for each of three agents, in which each chunk is assigned to two agents and of course, n -resilience can be achieved by each agent downloading all the chunks from the base-station. In terms of the effect on the group goal, the ejection scenario is equivalent to one of the agents

```

1 institution grid;
2
3 type Agent;
4 type Chunk;
5 type Time;
6 type Channel;
7 type ConnectionPoint;
8
9 exogenous event clock;
10 exogenous event obtain(Agent,Chunk,Channel);
11 exogenous event download(Agent,Agent,Chunk);
12
13 create event creategrid;
14
15 inst event intObtain(Agent,Chunk,Channel);
16 inst event intShare(Agent);
17 inst event intDownload(Agent,Chunk);
18 inst event transition;
19
20 violation event misuse(Agent);
21
22 fluent obtainChunk(Agent,Chunk);
23 fluent hasChunk(Agent,Chunk);
24 fluent abusy(Agent,Time);
25 fluent cbusy(ConnectionPoint,Time);
26
27 fluent previous(Time,Time);
28 fluent matchA(Agent,ConnectionPoint);
29 fluent matchC(Channel,ConnectionPoint);

```

Figure 3: Declaration of types and events in the model

```

34 obtain(A,X,C) generates intObtain(A,X,C)
35   if not cbusy(C1,T1), not cbusy(A1,T2),
36     matchA(A,A1), matchC(C,C1);
37
38 obtain(A,X,C) generates transition;
39 clock generates transition;
40
41 intObtain(A,X,C) initiates hasChunk(A,X);
42 intObtain(A,X,C) initiates
43   cbusy(A1,2), cbusy(C1,2)
44   if matchA(A,A1), matchC(C,C1);
45
46 transition initiates cbusy(A,T2)
47   if cbusy(A,T1), previous(T1,T2);
48 transition initiates pow(intObtain(A,X,C))
49   if cbusy(A1,1), matchA(A,A1);
50 transition initiates pow(intObtain(A,X,C))
51   if cbusy(C1,1), matchC(C,C1);
52
53 intObtain(A,X,C) terminates pow(intObtain(A,X1,C1));
54 intObtain(A,X,C) terminates pow(intObtain(B,X1,C));
55 intObtain(A,X,C) terminates obtainChunk(A,X);
56 intObtain(A,X,C) terminates perm(obtain(A,X,C1));
57
58 transition terminates cbusy(A,Time);

```

Figure 4: Generation and consequence relations for obtaining

```

63 download(A,B,X) generates
64   intDownload(A,X), intShare(B)
65   if hasChunk(B,X), not abusy(A,T1), not abusy(B,T2);
66
67 download(A,B,X) generates transition;
68 clock generates transition;
69
70 viol(intDownload(A,X)) generates misuse(A);
71
72 intDownload(A,X) initiates hasChunk(A,X);
73 intShare(B) initiates perm(intDownload(B,X));
74 intDownload(A,X) initiates abusy(A,3);
75 intShare(B) initiates abusy(B,3);
76
77
78 transition initiates abusy(A,T2)
79   if abusy(A,T1), previous(T1,T2);
80 transition initiates pow(intDownload(A,X))
81   if abusy(A,1);
82 transition initiates pow(intShare(B))
83   if abusy(B,1);
84
85 intDownload(A,X) terminates perm(intDownload(A,X));
86 intDownload(A,X) terminates pow(intDownload(A,X));
87 intDownload(A,X) terminates pow(intShare(A));
88 intShare(B) terminates pow(intDownload(B,X));
89 intShare(B) terminates pow(intShare(B));
90
91 misuse(A) terminates pow(intDownload(A,X)),abusy(A,T);
92 intDownload(A,X) terminates perm(intDownload(A,Y));
93
94 transition terminates abusy(A,Time);

```

Figure 5: Generation and consequence relations for sharing

leaving the ad-hoc network. In either case, for an a-priori solution there is a trade-off to be explored in delivering *i*-resilience, based on the estimated number agent failures and on the additional cost of replicated base-station downloads. Alternatively, some agents may engage autonomously in additional base-station downloads for the sake of the group goal.

A more practical sanction may be to lock the offending agent out of the sharing process for a number of time steps, but as with the above scenario, this is only effective if it does not impact the group goal.

3.2 Evaluation

Now that we have set out the normative framework and how to quantify communication costs for the particular situation of a 3G structured network and an ethernet ad-hoc network (see section 2.2.4), we can use the model to examine the traces for expected, but also unexpected behaviour and, simply by counting the number of `cbusy` and `abusy` states, get an estimate for battery consumption under different initial conditions.

Each of the models of our framework contains information about the energy consumption of each of agents in the form of the messages they have been passing signalled by the exogenous events `obtain` and `download` and the amount of time they have been spending communicating with the base-station, by the number of occurrences of `cbusy`, and communicating with the other agent by the number of times `abusy` occurs.

The model is presently being used as an off-line tool and generates all possible traces. The likelihood of a high proportion of these trace occurring in practice, depends on the relative intelligence and (bounded) rationality of the agents participating in the normative framework, e.g. continuously trying the download a chunk when you are busy. Our model purposely avoids modelling handset behaviour—we believe that is responsibility of the handset designer—because our objective is the exploration of the design of the space in which the handsets interact. However, these unsuccessful or unnatural traces can easily be filtered out by adding the filter displayed in Figures 8 and 9 to the *AnsProlog* specification. The first filter only admits those traces that lead to success: all agents have all the chunks and are no longer busy. When adding the filter, we obtain the first traces after nine time steps. To be more precise we obtain 142368 different traces satisfying the criteria. CLINGO returns these in 22.96 second, excluding printing, on a standard laptop. When the second filter is added to the first, we only obtain successful traces that contain no violations and where each exogenous event leads to its corresponding normative event. This reduces the number of traces significantly. Traces are only returned after fifteen time steps, after which 5280 of them are returned in 3.58 seconds. If we do not constrain each `download` and `obtain` to be followed by its normative equivalent, we get over three million traces.

By changing the durations for obtaining and sharing chunks and altering the penalties imposed on agents not conforming to the norms, we are able to study a variety of situations and finding the most appropriate enforcement mechanisms.

Furthermore some model assumption need to be reconsidered. The model at the moment demands that sending and receiving alternate. In reality this is might not always be the case. Handsets should be allowed to take advantages of chunks being sent even when the same number of chunks have not yet been shared. Thus, it would be more realistic to evaluate a handset’s willingness to collaborate over a larger time period.

4. DISCUSSION

In this paper, we have presented a normative framework as a

mechanism to help understand and to model the economic challenges that might arise in the context of a wireless grid. We have developed a model for the actions of the agents that participate in such a grid and hence provide us with a basic energy model, that may be used by the agents as part of a utility maximization decision-making process.

This was the first time we had modelled a complete, but simplified, realistic scenario of a normative framework. While InstAL is very intuitive and makes the task significantly more approachable it still lacks certain features that would make the modelling process easier. To model that channels and handsets were busy during a given period we had to resort to introducing the fluent `previous`, as InstAL does not allow arithmetic in its rules (which the underlying *AnsProlog* does allow). The current version of InstAL also does not allow hierarchies in its type structure or polymorphic propositions. Ideally we would have liked `Device` to be a superset of both `Channel` and `Agent`, such that we would not have had to resort to the `matchC` and `matchA` fluents, which are a technical artifice to overcome a linguistic weakness. The answer sets representing the traces contain significant numbers of atoms, making debugging difficult. Neither InstAL nor InstQL have built-in mechanisms to filter the output. So for debugging purposes we often referred to the underlying *AnsProlog* program and its `#show` and `#hide` functionalities, although those are not very flexible. Thus, the exercise has identified a number of practical issues that need to be addressed to make InstAL more usable.

The modelling of the wireless grid scenario gave us also a good insight into our formal model. The model does not allow us to expel an agent completely from a normative framework, as all the observed events are automatically empowered. While this can be partially remedied by removing the empowerment of consequent normative events, as we have done in the sharing phase, it raises interesting issues on how membership of a normative framework should be handled.

The traces of the normative framework give an indication of how much energy each of the handsets will be using when the trace would be executed. It also allows us to test different sanctioning techniques and compare their efficiency. However, a number of the traces that are produced by our simple model, while valid, stand very little chance of being executed by rational agents. Agents are not going to repeatedly downloading the same block, trying to download/obtain a block when they were busy. While this can be easily added to the model, we believe that this should be encoded by the agents rather than the normative framework. From a normative perspective we are only interested in correct, valid traces.

A particularly intriguing line of research, arising from the capacity to compute such traces, is to explore those (economic) mechanisms that might alleviate the effects of free-loading, in a more subtle, and less draconian way, than the simple sanction of expelling, that has been applied here.

Both the wireless grid scenario and the energy model are necessarily simplified and demand expansion. As stated earlier in the paper some functions such as the one defining p , i.e. the probability of finding a cooperation partner that has the right chunks, have to be specified. Further aspects of interest to be included in the model are error rates on the different communication links as well as the aspect that agents are moving within the environment and as a consequence the neighbourhood of an agent is constantly changing.

Furthermore, our current model has very simple penalty mechanisms for violating agents. It does not allow for more elaborate forms of sanctions as was demonstrated in Section 3.1.2 when we blocked agents that obtained unallocated chunks from the base-station. However, with regard to future work we plan to develop

several enforcement mechanisms in order to address the reciprocity problem in more detail. The idea thereby is to take the existing model as a reference point and analyse the additional benefits and costs resulting from different normative mechanisms.

Acknowledgements: Tina Balke is partially supported by a grant from the German Academic Exchange Service (DAAD).

5. REFERENCES

- [1] A. Artikis, M. Sergot, and J. Pitt. Specifying electronic societies with the Causal Calculator. In F. Giunchiglia, J. Odell, and G. Weiss, editors, *Proceedings of AOSE*, LNCS 2585. Springer, 2003.
- [2] R. Axelrod. The emergence of cooperation among egoists. *The American Political Science Review*, 75(2):306–318, 1981.
- [3] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge Press, 2003.
- [4] G. Boella and L. van der Torre. Constitutive Norms in the Design of Normative Multiagent Systems, City College. In *Proceedings of CLIMA-VI*, 2005.
- [5] O. Cliffe. *Specifying and Analysing Institutions in Multi-Agent Systems using Answer Set Programming*. PhD thesis, University of Bath, 2007.
- [6] O. Cliffe, M. De Vos, and J. Padget. Answer set programming for representing and reasoning about virtual institutions. In K. Inoue, S. Ken, and F. Toni, editors, *CLIMA VII*, volume 4371 of *LNAI*, pages 60–79, 2006. Springer.
- [7] O. Cliffe, M. De Vos, and J. Padget. Specifying and reasoning about multiple institutions. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organization, Institutions and Norms in Agent Systems II - AAMAS 2006 and ECAI 2006 International Workshops, COIN 2006*, volume 4386 of *LNCS*, pages 67–85. Springer Berlin / Heidelberg, 2007.
- [8] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. Nonmonotonic causal theories. *Artificial Intelligence, Vol. 153*, pp. 49-104, 2004.
- [9] F. H. P. Fitzek and M. D. Katz. Cellular controlled peer to peer communications: Overview and potentials. In F. H. P. Fitzek and M. D. Katz, editors, *Cognitive Wireless Networks*, pages 31–59. Springer, 2007.
- [10] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. Conflict-Driven Answer Set Solving. In *Proceeding of IJCAI07*, pages 386–392, 2007.
- [11] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3-4):365–386, 1991.
- [12] L. Hopton, O. Cliffe, M. De Vos, and J. Padget. Instql: A query language for virtual institutions using answer set programming. In J. Dix, M. Fisher, and P. Novák, editors, *Proceedings of the 10th International Workshop on Computational Logic in Multi-Agent Systems (ClimaX)*, IfI Technical Report Series, pages 87–104, 2009. Institut für Informatik, Technische Universität Clausthal.
- [13] A. J. Jones and M. Sergot. A Formal Characterisation of Institutionalised Power. *ACM Computing Surveys*, 28(4es):121, 1996.
- [14] M. D. Katz and F. H. P. Fitzek. Cooperation in 4g networks - cooperation in a heterogenous wireless world. In F. H. P. Fitzek and M. D. Katz, editors, *Cooperation in Wireless Networks: Principles and Applications*, pages 463–496. Springer, 2006.
- [15] R. A. Kowalski and F. Sadri. Reconciling the event calculus with the situation calculus. *Journal of Logic Programming*, 31(1–3):39–58, Apr. 1997.
- [16] I. Niemelä and P. Simons. Smodels: An implementation of the stable model and well-founded semantics for normal LP. In J. Dix, U. Furbach, and A. Nerode, editors, *Proceedings of LPNMR’97*, volume 1265 of *LNAI*, pages 420–429, 1997. Springer.
- [17] E. Ostrom. Coping with tragedies of the commons. *Annual Review of Political Science*, 2:493–535, June 1999. Workshop in Political Theory and Policy Analysis; Center for the Study of Institutions, Population, and Environmental Change.
- [18] G. P. Perrucci and F. Fitzek. Measurements campaign for energy consumption on mobile phones. Technical report, Aalborg University, 2009.
- [19] G. P. Perrucci, F. H. Fitzek, and M. V. Petersen. Energy saving aspects for mobile device exploiting heterogeneous wireless networks. In *Heterogeneous Wireless Access Networks*. Springer US, 2009.
- [20] M. Sergot. (C+)+: An Action Language for Representing Norms and Institutions. Technical report, Imperial College, London, 2004.
- [21] TNS. Two-day batter life tops wish list for future all-in-one phone device. Technical report, Taylor Nelson Sofres, 004.
- [22] J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing multiagent systems. *AAMAS*, 11(3):307–360, 2005.
- [23] Virginia Dignum. *A Model for Organizational Interaction Based on Agents, Founded in Logic*. PhD thesis, Utrecht University, 2004.
- [24] K. Wrona and P. Mähönen. Analytical model of cooperation in ad hoc networks. *Telecommunication Systems*, 27(2–4):347–369, 2004.

```

98 initially
99   pow(transition), perm(transition),
100   perm(clock),
101   pow(intObtain(A,B,C)), perm(intObtain(A,B,C)),
102   perm(obtain(alice,x1,C)), perm(obtain(alice,x3,C)),
103   perm(obtain(bob,x2,C)), perm(obtain(bob,x4,C)),
104   obtainChunk(alice,x1), obtainChunk(alice,x3),
105   obtainChunk(bob,x2), obtainChunk(bob,x4);

109 initially
110   pow(transition), perm(transition),
111   perm(clock),
112   pow(intDownload(Agent,Chunk)), pow(intShare(Agent)),
113   perm(download(Agent,Agent1,Chunk)),
114   perm(intDownload(Agent,Chunk)), perm(intShare(Agent));

```

Figure 6: Initial state of the model, post negotiation

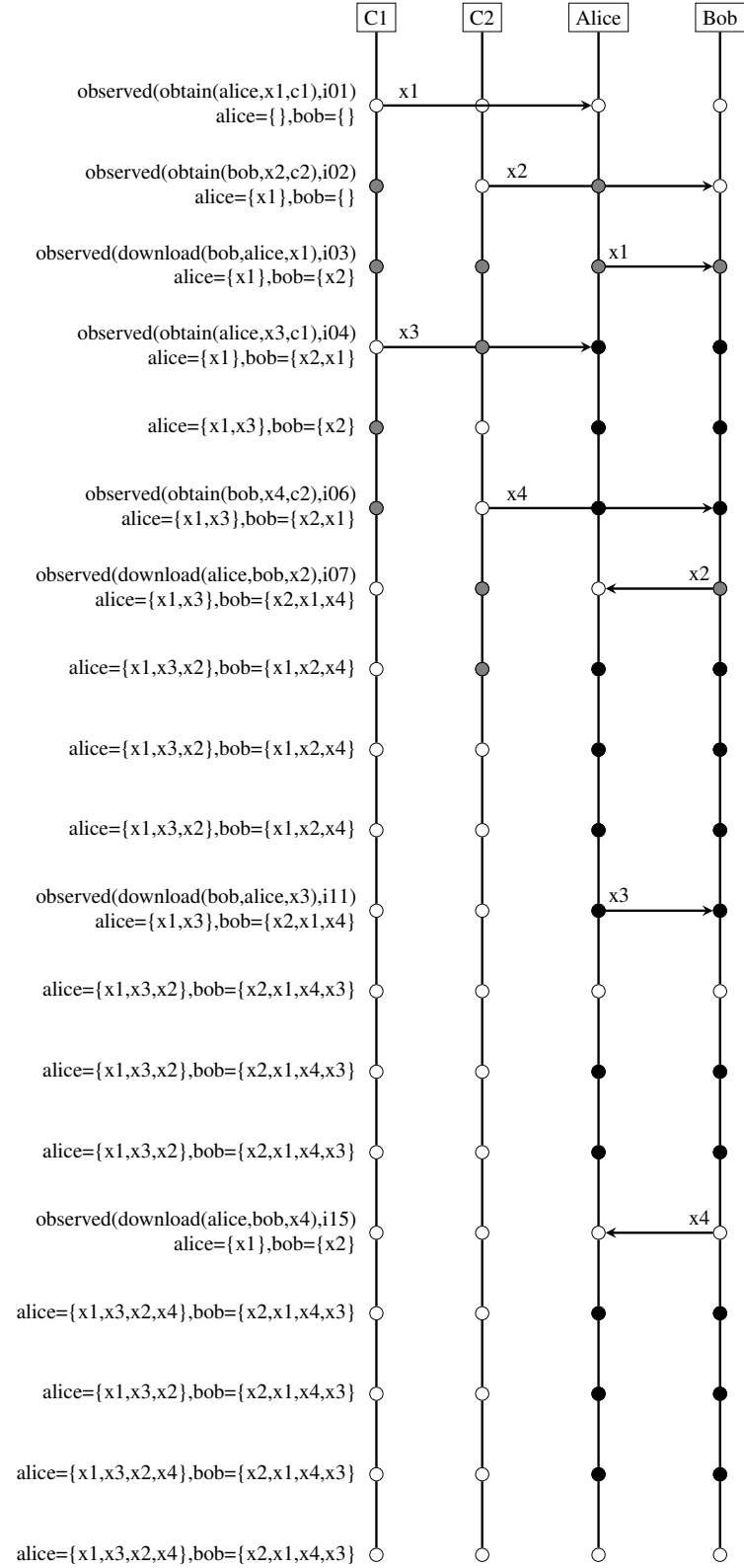


Figure 7: One trace of the interaction between alice, bob and the channels of base-station

```

1  % success criteria
2  success :- holdsat(hasChunk(alice,x1),T),holdsat(hasChunk(alice,x2),T),
3             holdsat(hasChunk(alice,x3),T),holdsat(hasChunk(alice,x4),T),
4             holdsat(hasChunk(bob,x1),T),holdsat(hasChunk(bob,x2),T),
5             holdsat(hasChunk(bob,x3),T),holdsat(hasChunk(bob,x4),T),
6             not holdsat(cbusy(dbob,T),F),not holdsat(cbusy(dalice,T),F),
7             not holdsat(cbusy(dc1,T),F), not holdsat(cbusy(dc2,T),F),
8             not holdsat(abusy(alice,T),F),not holdsat(abusy(bob,T),F),final(F).

```

Figure 8: A filter to remove unsuccessful traces

```

10 % only interested in successful traces
11 :- not success.
12
13 % indication that a violation has occurred
14 viol :- occured(viol(X),I).
15
16 :- viol.
17
18 % exogenous event should be follow by corresponding normative event
19 :- occured(download(H1,H2,Chunk),T), not occured(intDownload(H1,Chunk),T).
20 :- occured(obtain(Handset,Chunk,Channel),T),
21     not occured(intObtain(Handset,Chunk,Channel),T).

```

Figure 9: A filter to remove unsuccessful violation traces with unintuitive events