UNIVERSITY OF BATH

Citation for published version:
De Vos, M, Balke, T & Satoh, K 2013, Modelling legitimate expectations. in Y Motomura, A Butler & D Bekki (eds), New Frontiers in Artificial Intelligence: JSAI-isAI 2012 Workshops, LENLS, JURISIN, MiMI, Miyazaki, Japan, November 30 and December 1, 2012, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7856, Springer, Berlin, pp. 141-156, 4th JSAI International Symposia on Artificial Intelligence, JSAI-isAI 2012, Miyazaki , Japan, 30/11/12. https://doi.org/10.1007/978-3-642-39931-2_11

DOI:
10.1007/978-3-642-39931-2_11

Publication date:
2013

Document Version
Peer reviewed version

Modelling Legitimate Expectations

Marina De Vos[1], Tina Balke[2,1], and Ken Satoh[3]

[1] University of Bath, Department of Computer Science, UK
mdv@cs.bath.ac.uk
[2] University of Surrey, Centre for Research in Social Simulation, UK
t.balke@surrey.ac.uk
[3] National Institute of Informatics, Principles of Informatics Research Division, Japan
ksatoh@nii.ac.jp

Abstract. Legitimate expectation in the context of *culpa in contrahendo* is an important legal concept for the study of good faith and the duty to negotiate with good care. However when wanting to model it and reason about it, one finds that most existing legal formalisations do not directly account for the concept. In this paper we present a formal model that can explicitly model and reason about legitimate expectations by extending the formal INSTAL legal framework. We demonstrate our extensions with the help of a private law case study which has gained wide popularity in Japanese law.

1 Legal Frameworks and Legitimate Expectations

In the legal reasoning community, logical formalisations have been used for a considerable period of time for modelling and reasoning about legal concepts. [9] provides a detailed overview and discussion of the various approaches. Nevertheless these do not capture all legal concepts. *Legitimate expectations* are one example of a concept which is not explicitly accounted for in existing approaches.

Legitimate expectations are a legal concept which is typically mentioned in connection with the long standing legal doctrine of *culpa in contrahendo* (Latin for "fault in conclusion of a contract"). The concept goes back to an article by von Jhering, published in 1861, entitled "Culpa in contrahendo, oder Schadensersatz bei nichtigen oder nicht zur Perfektion gelangten Verträgen" [14]. The idea described in this article is the duty to negotiate with care. This duty includes not falsely leading a negotiation partner to a legitimate expectation that might result in him acting to his detriment before a contract is concluded. In case of a breach of this duty, the party to blame can be liable for damages suffered by the negotiating party relying on the conclusion of a firm contract [10].

The degree by which *culpa in contrahendo* is applied differs between countries. For example, in German contract law for example *culpa in contrahendo* is explicitly accounted for (§311 BGB specifies a number of steps by which an obligation to pay damages may be created). In contrast, the majority of common law jurisdictions are conservative with respect to the *culpa in contrahendo* doctrine and only apply it if consideration can be proven by the claimant. For example, in the US and the UK, the doctrine of *promissory estoppel* works as a model for *culpa in contrahendo*, whereas in Japan it is categorized as a problem of the good faith principle.

However the general idea of its application remains the same: if the claimant has acted with the appropriate consideration and as a consequence of the actions of the negotiation partner had *legitimate expectations* that the contract would be firmly concluded, *culpa in contrahendo* can be applied. In this paper we understand the term legitimate expectations in the broader sense of fairness and pre-contractual liability based on the abuse of rights outline above, i.e. not only applying to public law (as done in English law).

In this paper we propose, to our knowledge, the first model for representing and reasoning about legitimate expectations as a basis for *culpa in contrahendo*. We use an extension of Cliffe et al's [3] formal legal framework. Its formal model is solely based on mathematical constructs (i.e. functions and relations), thereby avoiding formalism specific side-effects. In detail, it allows us to specify the concept of legitimate expectations as well as the components required for it independently of the afterwards chosen implementation language.

To demonstrate our approach we use a private law case study which has gained wide popularity in Japanese law, in particular w.r.t. the principle of good faith[4]. This case study was first portrayed in a Workshop on the Sales Convention by Professor Shigeru Kagayama of Nagoya University (Japan)[5]as follows:

> A dentist (buyer and defendant) wanted to open a clinic, and, therefore, entered into negotiations to conclude a contract for the purchase of space in a suitable building. During the negotiations, the buyer specified the space needed for the dental clinic, gave the seller plans for the layout of the space, pointed out that the existing space lacked the electrical capacity required for the clinic, and implicitly authorized the owner to change the design and construct facilities suitable for the clinic. After six months, however, the buyer broke off negotiations because he had decided that the space available in the seller's building was too small.

The Japanese Supreme Court (Decision of September 18, 1984, Conf. Hanrei Jiho No. 1137, p. 51.) decided the case by applying legitimate expectations principles. The court held that – despite no contract having been signed by the two parties – the buyer was liable to the seller for losses caused when the seller changed the design of the space and incurred construction costs, because the buyer had not acted in good faith in negotiating the contract. The *culpa in contrahendo* principle was applied.

The paper is structured as follows. In Sec. 2 we provide a formal model for reasoning about legitimate expectations, based on the concept of legal frameworks, by first explaining the existing INSTAL legal framework and afterwards highlighting the extensions made. Both the syntax and the semantics of the framework are explained in detail. In Sec. 3 we demonstrate our approach with the help of the private law promissory Estoppel case study described earlier The paper finishes with a description of related work (Sec. 4), conclusions and an outline for future work (Sec. 5).

---

[4] In the UK and the US, the case study provided in this paper is typically considered as an example of promissory estoppel.

[5] The transcript of the workshop discussion including the case presented here were reproduced in the Journal of Law & Commerce [7].

2 Formal Model

Normative systems, also referred to as institutions or organisations, have been studied extensively in the area of multi-agent systems as a mechanism to regulate and evaluate the behaviour of the participating agents. Constructs like permission, institutional power (i.e. being able to affect a change), obligation, prohibition are used to describe the desired behaviour of the system and its participants. The formal representation of legal system by normative systems has been a subject of research for several decades: a comprehensive discussion appears in [9]. In [4] it was demonstrated that the formal INSTAL model of Cliffe et [3] was an appropriate model for legal reasoning. In this paper, we extend INSTAL to be able to explicitly represent and reason about legitimate expectations. Before we discuss its extension, we briefly discuss the original model.

2.1 Original Formal Model

The INSTAL framework's purpose is the formalisation of the effects of traces (i.e. sequences) of individuals' actions within a legal context. We refer to the formal model of (parts of) a legal system as a legal framework.

The individuals' actions that a legal framework is able to recognise are referred to as its exogenous events, $\mathcal{E}_{ex}$. These exogenous events need to be interpreted in the (legal) context that is being modelled. For example, raising your hand in class has a different meaning than raising your hand in an auction with the latter signifying that you wish to bid for the item on offer and that you intend to pay for the item if you win the auction. This interpretation is taken care of by the generate function. This function, $\mathcal{G}$, maps an individual's action, subject to conditions on the legal state, to its corresponding legal action $\mathcal{E}_{act}$. When required, it may also generates further events from any legal action to make the consequences of events more fine-grained. Apart from exogenous events ($\mathcal{E}_{ex}$) and their legal interpretations ($\mathcal{E}_{act}$), the framework recognises one more type of events, namely violation events ($\mathcal{E}_{viol}$) which indicate that one of the laws of the system was broken. Together, legal actions and violations comprise the legal events ($\mathcal{E}_{legal}$) of a legal framework.

The consequences of events are formalised by the $\mathcal{C}$ relation, which indicates which fluents needs to be initiated or terminated in the current legal state as a result of the occurrence of an event.

The legal state is denoted as a set of fluents ($\mathcal{F}$) that are true at that particular point of time. If a fluent is currently not part of the state, the fluent is considered false. The set of available fluents ($\mathcal{F}$) of a given legal framework consists of four disjoint subsets of fluents. The domain fluents, $\mathcal{D} \subseteq \mathcal{F}$, describe the properties of the environment, e.g. ownership, contract, recorded data. Permission fluents, $\mathcal{P} \subseteq \mathcal{F}$, denote that an event is permitted. The absence of the permission fluent for an event in the current state, indicates that the event is prohibited. Legal power fluents, $\mathcal{W} \subseteq \mathcal{F}$ indicate whether an event has the legal power to affect the legal state; for example whether an individual has the legal power to witness a signature. An event that is not empowered has no (legal) effect. The last subset contains the obligation fluents, $\mathcal{O} \subseteq \mathcal{F}$, which denote that a certain event has to take place before a deadline event. Failing this a specified violation event will occur. For example, you need to ask for planning permission before you start renovation. Once the obligation is satisfied or violated the obligation is removed from the state.

State conditions ($\mathcal{X}$) are expressed over $\mathcal{X} = 2^{\mathcal{F} \cup \neg \mathcal{F}}$. The initial state, $\Delta \subseteq \mathcal{F}$, is the set of fluents that are true at the start of legal framework. Putting the foregoing together, we have Cliffe et al's legal framework, INSTAL, as the quintuple $L = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \Delta \rangle$.

A summary of the framework can be found in Fig. 1. The original model appears as plain text. Our extension components have a box around them.

The semantics of the legal framework is defined over a sequence, called a trace, of exogenous events. Starting from the initial state of the legal framework, for each exogenous event in the trace, we take (i) the transitive closure of the $\mathcal{G}$ function augmented with a violation detection for unfulfilled obligations and non-permitted events to generate all events taking place in the framework, and (ii) for each of these events, the $\mathcal{C}$ relation is used to determine the fluents that need to be initiated and terminated in order to derive the next state. We also terminate obligations that were met or violated. We will discuss the semantics of the formal model in more detail once we extended the model to deal with legitimate expectations.

2.2 Legitimate Expectations

Having discussed the model that we wish to extend, we can now start with modelling legitimate expectations.

Syntax We start by rephrasing the earlier definition of legal expectations in terms of components of our legal framework.

A legitimate expectation indicates that sufficient grounds exist to be certain that a certain event will or will not take place or that a specific state will or will not be reached in the future. This implies that the system and its participants are obliged to (not) bring the event or state about. Any deviation from the legitimate expectation results in a violation. Participants in the legal system (and its formal representation) are entitled to use these expectations to influence their actions. So they need to be granted the normative capabilities to do so. For example, using our case-study, if the seller has the legitimate expectation that the dentist will buy the property, he is given the implicit permission or authorisation to make the necessary adjustments to the property at the expense of the buyer, i.e. the dentist.

To denote the legitimate expectations, we introduce a new set of fluents, $\mathcal{Z} \subseteq \mathcal{F}$. To specify the conditions for a legitimate expectation to be initiated or terminated, we provide the functions $\mathcal{ZC}^{\uparrow} :: \mathcal{X} \to \mathcal{Z}$ and $\mathcal{ZC}^{\downarrow} :: \mathcal{X} \to \mathcal{Z}$ respectively. Both functions map a state condition (i.e. a set of fluents to be true or false) to the corresponding legitimate expectation.

Legitimate expectations result in obligations for the system and/or its participants. At the moment, we can model that an event is expected to take place but not that a state with particular properties will be reached. The INSTAL model of Cliffe et al is solely event driven, implying that all normative behaviour is expressed in terms of events. The original model does not allow to state, for example, that the system is obliged to

$\mathcal{L} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \mathcal{LC}, \mathcal{ZC}^{\uparrow}, \mathcal{ZC}^{\downarrow}, \mathcal{ZO}, \Delta^* \rangle$, where

1. $\mathcal{E} = \mathcal{E}_{ex} \cup \mathcal{E}_{legal}$
   with $\mathcal{E}_{legal} = \mathcal{E}_{act} \cup \mathcal{E}_{viol}$ with
   $\mathcal{E}_{viol} \supseteq \{viol(perm(e)) \mid e \in \mathcal{E}_{ex} \cup \mathcal{E}_{legal}\} \cup \{viol(pow(e)) \mid e \in \mathcal{E}_{act}\}$
2. $\mathcal{E}_{dis} \in \mathcal{E}$
3. $\mathcal{F} = \mathcal{F}_{ev} \cup \mathcal{F}_{st}$ with
   (a) $\mathcal{F}_{ev} = \mathcal{F}_{rn} \cup \mathcal{D} \cup \mathcal{V} \cup \mathcal{NO}$ with $\mathcal{F}_{rn} = \mathcal{W} \cup \mathcal{P} \cup \mathcal{O}$:
      (i) $\mathcal{P} = \{\mathrm{perm(e)} \mid e \in \mathcal{E}\}$
      (ii) $\mathcal{W} = \{\mathrm{pow(e)} \mid e \in \mathcal{E}_{act}\}$
      (iii) $\mathcal{O} = \{\mathrm{obl(e, d, v)}, obl(e, v)\}$ with
         – $e, d \in \mathcal{E}, v \in \mathcal{E}_{viol}$
         – $e, d \in \mathcal{L}, v \in \mathcal{V}$
         – $e \in \mathcal{L}, d \in \mathcal{E}, v \in \mathcal{V}$
         – $e \in \mathcal{E}, d \in \mathcal{L}, v \in \mathcal{V}$
      (iv) $\mathcal{NO} = \{nobl(e, d, v), nobl(e, v)\}$
      (v) $\mathcal{V} \supseteq \{viol(e) \mid e \in \mathcal{E}_{viol}\}$
      (vi) $\mathcal{F}_{st} = \mathcal{Z} \cup \mathcal{L}$
4. $\mathcal{G} :: \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{E}_{legal}}$
5. $\mathcal{C} :: \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}_{ev}} \times 2^{\mathcal{F}_{ev}}$ with $C(X, e) = (\mathcal{C}^{\uparrow}(X, e), \mathcal{C}^{\downarrow}(X, e))$ where
   (i) $\mathcal{C}^{\uparrow}(X, e)$ initiates an $event$ fluent
   (ii) $\mathcal{C}^{\downarrow}(X, e)$ terminates an $event$ fluent
6. $\mathcal{LC} :: \mathcal{X} \rightarrow \mathcal{L}$
7. $\mathcal{ZC}^{\uparrow} :: \mathcal{X} \rightarrow \mathcal{Z}$
8. $\mathcal{ZC}^{\downarrow} :: \mathcal{X} \rightarrow \mathcal{Z}$
9. $\mathcal{ZO} :: \mathcal{Z} \rightarrow 2^{\mathcal{O} \cup \mathcal{NO}} \times 2^{\mathcal{F}_{rn}} \times 2^{\mathcal{F}_{rn}}$
10. $\Delta^* = \Delta \cup \{live\}$
11. State Formula: $\mathcal{X} = 2^{\mathcal{F} \cup \neg \mathcal{F}}$
12. States: $\Sigma = 2^{\mathcal{F}}$

Fig. 1. Formal specification of the legal framework

reach a state where a particular condition holds. It is not the aim of this paper to extend INSTAL to also be fully state driven. Instead, we only add the necessary state features to deal with legitimate expectations and leave the remainder for future work.

Using the normative specification language OPERA [5,12] as inspiration, we introduce the concept of landmark to INSTAL. OPERA is normative specification framework for multi-agent systems that takes an organisational view. Their focus is agents as a community achieving and avoiding certain normative states rather than the normative behaviour of the individual agents. OPERA introduces the concept of landmarks, representing a formula of constructed of state fluents. They are used to determine acceptable states that agents are encouraged or obliged to reach in order to achieve system goals and unacceptable state that should trigger a normative reaction when reached.

The fulfilment of a landmark is decided based on whether the current state satisfies its corresponding state formula or not. When a landmark is fulfilled, this is recorded in the state of the legal framework. Using this information, we introduce a new set of fluents, $\mathcal{L} \subseteq \mathcal{F}$, to represent the state conditions that we wish to reason about, and a function $\mathcal{LC} :: \mathcal{X} \rightarrow \mathcal{L}$ to map the state condition to the landmark fluents. During the initiation of a new state, the landmark function is used to determine if new landmarks have been reached.

Landmarks and legitimate expectations are initiated and terminated based on the provisional state provided by the event generation and consequence relation before the state becomes final. Since the conditions of both concepts can rely on newly initiated/terminated landmarks or legitimate expectations, we need to repeatedly call these functions to obtain the final set of landmarks and expectation that will form part of the new state. To avoid infinite repetition, we impose restrictions on the $\mathcal{LC}$, $\mathcal{ZC}^{\uparrow}$ and $\mathcal{ZC}^{\downarrow}$ functions. Informally, no cycle of dependencies, positive or negative, should exist between landmarks and expectations. Formally, the directed graph with $\mathcal{Z} \cup \mathcal{L}$ as nodes and edges between nodes a and b if $\exists(X, b) \in \mathcal{LC}$, $(X, b) \in \mathcal{ZC}^{\uparrow}$ or $(X, b) \in \mathcal{ZC}^{\downarrow}$ such that $a \in X$ or $\neg X \in X$ needs to be acyclic.

The consequence relation is responsible for initiating and terminating fluents as the consequence of an event. Since legitimate expectations and landmarks are not necessarily event driven, the consequence relation should not be concerned with them. To differentiate, we subdivide the set of fluents into two subsets $\mathcal{F}_{st} \subseteq \mathcal{F}$, called state fluents, which include legitimate expectations and landmarks and $\mathcal{F}_{ev} \subseteq \mathcal{F}$, named event fluents, which include all other fluents. The latter can be directly affected by events, hence the name, while the former can only be influenced by the current state. The consequence operator can only influence the event fluents, hence we change its image to $\mathcal{F}_{ev}$ instead of fluents. Within the event fluents, we distinguish between domain fluents and the normative fluents ($\mathcal{F}_{rn} \subseteq \mathcal{F}_{ev}$). The initiation and termination of state fluents is dealt with by other functions (i.e. $\mathcal{ZC}^{\uparrow}$, $\mathcal{ZC}^{\downarrow}$, $\mathcal{LC}$). Note that landmarks once achieved cannot be terminated.

In Cliffe et al's INSTAL framework, breaking the normative rules of the legal framework results in a violation event. While this is an appropriate response for a pure event-based model, it becomes a problem when one wants to introduce state based norms as well as event-based ones. State-based norms are evaluated after the effects of the events have been determined. A violation event for these norms would have to be evaluated separately, possible causing further violations and changes to state. To avoid this, we propose the use of violation fluents $\mathcal{V} \subseteq \mathcal{F}$. This also fits better with the concept of state-based norms. For completeness, whenever a violation event occurs a corresponding violation fluent is initiated. Using violation fluents has a further advantage: norm-aware participants can easily query the current state of the system to see if any violations have occurred.

Current obligations require a deadline event by which the obligation has to be fulfilled. This can be rather limited for the system and its designer. In some cases, the designer might not be able to state that something needs to happen or some landmark needs to be satisfied. To make this possible, we introduce dissolution events $\mathcal{E}_{dis} \subseteq \mathcal{E}$. These are events that indicate the end of the legal framework. At the start of a legal framework, we introduce an extra fluent `live` to the state. Events will only have an effect for as long as this fluent is part of the state. When a dissolution event occurs this `live` fluent is terminated. Obligations without a deadline event will automatically be assumed to have a dissolution event as their deadline event.

Having defined landmarks, violation fluents and dissolution events, we can extend our available obligations to obligations with an event or landmark as their target; an event, landmark or an implicit dissolution event as deadline; and a violation fluent or violation event as consequence for not satisfying the obligation. However, it should be noted that a violation fluent needs to be specified when a landmark is used as one the arguments. In order to model legitimate expectations, we also need, apart from these so-called positive obligations, to be able to expresses that system and its participants are obliged not to reach a certain landmark or perform a certain action for a certain period of time. While the latter is akin to permission it is not entirely the same as permissions do not have a deadline. These negative obligations ($\mathcal{NO} \subseteq \mathcal{F}$) operate in the opposite way from their positive counterparts. The violation occurs or is initiated when the target event occurs or the landmark is reached.

With obligations extended to cope with landmarks and the introduction of negative obligations, we can finalise the modelling of legitimate expectations. To map each legitimate expectation to its corresponding obligations and consequences, we provide the function $\mathcal{ZO} :: \mathcal{Z} \rightarrow 2^{(\mathcal{O} \cup \mathcal{NO})} \times 2^{\mathcal{F}_{rn}} \times 2^{\mathcal{F}_{rn}}$. The first $\mathcal{F}_{ev}$ refers to normative fluents that need to be initiated while the second one indicates the normative fluents that need terminating as a response to the initiation of the legitimate expectation. When the legitimate expectation is terminated the reverse is applied.

A full overview of the syntax of our extended legal framework can be found in Figure 1. The additions to the original INSTAL model are surrounded by a box.

Semantics The semantics of the extended INSTAL model consists of three phases. The first phase corresponds to the state transformation of the original model. It takes into account the events being generated and their consequences. This intermediate state is used for the initiation and termination of landmarks and legitimate expectations that need to be initiated and terminated resulting in a second intermediate state. In the third and final phase, this second intermediate state is used to initiate and terminate the obligations and consequences resulting from legitimate expectations and to deal with state-based obligations.

Event Generation The event generation of the extended version of INSTAL remains mostly unchanged. We only have to accommodate for negative obligations and obligations without a deadline event when a dissolution event occurs.

The generation of all events in a given state is specified by the function $\mathrm{GR} : \Sigma \times 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}}$. In some state S, subject to a set of events E, $\mathrm{GR}(S, E)$ returns all the events generated by the occurrence of events in E occurring in state S. It is defined as follows[6]:

[6] We use $S \models f$ if $f \in S$ and $S \not\models f$ if $f \notin S$.

$$
\begin{aligned}
\mathrm{GR}(S,E) = \{e \in \mathcal{E} \mid\ & e \in E & \textbf{or}\\
& \exists\, e' \in E, x \in \mathcal{X}, e \in G(x,e') \cdot S \models \mathrm{pow}(e) \wedge S \models x & \textbf{or}\\
& \exists\, e' \in E, x \in \mathcal{X}, e \in G(x,e') \cdot e \in \mathcal{E}_{viol} \wedge S \models x & \textbf{or}\\
& \exists\, e' \in E \cdot e = viol(e'), S \not\models \mathrm{perm}(e') & \textbf{or}\\
& \exists\, e' \in \mathcal{E}, d \in E \cdot S \models \mathrm{obl}(e',d,e) & \textbf{or}\\
& \exists\, e' \in \mathcal{E}, E \cap \mathcal{E}_{dis} \neq \emptyset \cdot S \models \mathrm{obl}(e',e) & \textbf{or}\\
& \exists\, e' \in E, d \in \mathcal{E}, \cdot S \models \mathrm{nobl}(e',d,e) & \textbf{or}\\
& \exists\, e' \in E \cdot S \models \mathrm{nobl}(e',e)\}
\end{aligned}
$$

- The first condition ensures that events remain generated (inertia).
- The second condition defines event generation to be explicitly specified by the relation G. One event generates another event in a given state, when (i) the generation was specified by the framework (ii) the generated event is empowered and (iii) the current state satisfies the conditions for the generation
- The third condition deals with violations generated as specified by the framework rather than violations resulting from events that were not permitted. Violations do not require empowerment.
- The fourth condition considers the generation of violation events as the result of the occurrence of non-permitted events.
- The fifth and sixth condition deals with the generation of violation events as a result of the failure to bring about an obliged event. For all asserted obligation fluents, the occurrence of the deadline event d or a dissolution event generates the corresponding violation event e
- The final two conditions deal with the violation of negative obligations. This occurs when the forbidden event of the obligation occurs.

It is easy to see that $\mathrm{GR}(S,E)$ is a monotonic function. This implies that for any given state and a set of events, we can obtain a fixpoint $\mathrm{GR}^{\omega}(S,E)$. In our legal framework, we are interested in all the events generated from a single exogenous event $e_{ex} \in \mathcal{E}_{ex}$ occurring in a certain state. So, we need $\mathrm{GR}^{\omega}(S,\{e_{ex}\})$.

Using these generated events, we can determine the fluents that need initiating and terminating as a consequence of the occurrence of these events. The main difference with the original INSTAL model lies in the positive and negative obligations and the introduction of dissolution events.

A fluent will be initiated as a result of the consequence relation in response to a generated event generated. Alternatively, a violation fluent is initiated as a consequence of its violation event. Or more formally: the set of all initiated fluents $\text{E-INIT}(S,e_{ex}) \subseteq \mathcal{F}$ for some state $S \in \Sigma$ and an exogenous event $e_{ex} \in \mathcal{E}_{ex}$ is defined as:

$$
\begin{aligned}
&\text{E-INIT}(S,e_{ex}) =\\
&\quad \{f \in \mathcal{F} \mid \exists\, e \in \mathrm{GR}^{\omega}(S,\{e_{ex}\}), X \in \mathcal{X} \cdot f \in \mathcal{C}^{\uparrow}(X,e) \wedge S \models X \text{ or}\\
&\quad\quad \exists\, e \in \mathrm{GR}^{\omega}(S,\{e_{ex}\}) \cdot e \in \mathcal{E}_{viol}, f \in \mathcal{V}, f = viol(e)\}
\end{aligned}
$$

A fluent is terminated if an event is generated in the current state for which $\mathcal{C}$ specifies that it needs terminating. Furthermore, an obligation fluent is terminated if either its (implicit) deadline or the (non) obliged event are in the set of generated events. In case of the occurrence of a dissolution event, the `live` fluent is terminated.

$$
\begin{array}{l}
\text{E-TERM}(S, e_{ex}) = \\
\qquad \{f \in S \mid \exists\, e \in \text{GR}^{\omega}(S, \{e_{ex}\}), X \in \mathcal{X} \cdot f \in \mathcal{C}^{\downarrow}(X, e), S \models X \ \textbf{or} \\
\qquad\qquad f = \text{obl}(e, d, v) \wedge e \in \text{GR}^{\omega}(S, \{e_{ex}\}) \quad \textbf{or} \\
\qquad\qquad f = \text{obl}(e, v) \wedge (e \vee d \in \text{GR}^{\omega}(S, \{e_{ex}\})) \quad \textbf{or} \\
\qquad\qquad f = \text{obl}(e, v) \wedge \text{GR}^{\omega}(S, \{e_{ex}\}) \cap \mathcal{E}_{dis} \neq \emptyset \quad \textbf{or} \\
\qquad\qquad f = \text{nobl}(e, d, v) \wedge (e \vee d \in \text{GR}^{\omega}(S, \{e_{ex}\})) \quad \textbf{or} \\
\qquad\qquad f = \text{nobl}(e, v) \wedge e \in \text{GR}^{\omega}(S, \{e_{ex}\}) \quad \textbf{or} \\
\qquad\qquad f = \text{nobl}(e, v) \wedge \text{GR}^{\omega}(S, \{e_{ex}\}) \cap \mathcal{E}_{dis} \neq \emptyset \quad \textbf{or} \\
\qquad\qquad f = \text{live} \wedge \text{GR}^{\omega}(S, \{e_{ex}\}) \cap \mathcal{E}_{dis} \neq \emptyset\}
\end{array}
$$

The first intermediate state is created from the current state by adding the initiated fluents and removing the terminated ones. This is done by the transition function $\text{E-TR} : \Sigma \times \mathcal{E}_{ex} \rightarrow \Sigma$:

$$\text{E-TR}(S, e_{ex}) = (S \cup \text{E-INIT}(S, e_{ex})) \setminus \text{E-TERM}(S, e_{ex})$$

Dealing with Legitimate Expectations and Landmarks Now that we have dealt with the event fluents, we can focus our attention on the initiation and termination of the state fluents. This is an iterative process in the same way as generating events is. For example, the initiation of a landmark fluent can result in the initiation of a legitimate expectation. During iteration we do have to be careful to avoid an infinite cycle due to negation-as-failure (something is false if it is not in the state). This is taken care of by using the fixpoint of initiation and termination rather than the fixpoint of each individually.

The initiation and termination steps are straightforward. If the state matches the conditions in the function then initiate/terminate the landmark or legitimate expectation.

$$
\begin{array}{l}
\text{S-INIT}(S) \quad = \{f \in \mathcal{F} \mid f = \mathcal{LC}(X), S \models X \text{ or } f = \mathcal{ZC}^{\uparrow}(X), S \models X\} \\
\text{S-TERM}(S) = \{f \in \mathcal{F} \mid f = \mathcal{ZC}^{\downarrow}(X), S \models X\}
\end{array}
$$

Combining these two to obtain a new intermediate state is harder. The relation TR_s operates over a pair of states. The first element is the original state, the second state is the future new state which will become the new intermediate state once a fixpoint is reached. The new state is obtained by iteratively removing the state fluents that need terminating from the original state and adding the ones marked for initiation. Termination and initiation is determined on the second state. Our acyclic condition on the dependency graph of the landmark and legitimate expectations guarantees termination.

$$\text{S-TR}(S_1, S_2) = (S_1, (S_1 \cup \text{S-INIT}(S_2)) \setminus \text{S-TERM}(S_2))$$

To find the new state of our legal framework after the occurrence of an exogenous event e_{ex} in the current state S, we are interested in the fixpoint $\text{S-TR}^{\omega}(\text{E-TR}(S, e_{ex}), S)$ in general and the second argument in particular. We denote this second argument of the fixpoint as $\text{S-TR}(S)$. Notice that legitimate expectations that are already part of the state do not get re-initiated.

Dealing with the remaining obligations and the consequences of legitimate expectations The last step in obtaining the complete state transition is dealing with obligations that rely on landmarks and the consequences of initiated and terminated legitimate expectations. It should be noted that by dealing with these separately from the initiation and termination of landmarks and legitimate expectations, we possibly introduce delay effects when state expressions use obligations. However this is in line with obligations that do not use landmarks.

On the initiation part, obligations and initiating consequences resulting from the added legitimated expectations are marked for initiation[7]. The same is done for violation fluents resulting from violated obligations that have a violation fluent as their sanction. Also, fluents marked for initiation after the termination of a legitimate expectation are added. To be able to do so, initiation takes the temporary state obtained after initiating and terminating event fluents, the temporary state after landmarks and legitimated expectations have been updated and the observed event as input.

$$
\begin{aligned}
&\text{O-INIT}(S_1, S_2, e_{ex}) = \{f \in \mathcal{F} \mid \\
&\quad \exists\, l \in (S_2 \setminus S_1) \cap \mathcal{Z} \cdot (f, I, T) \in \mathcal{ZO}(l) \wedge I, T \subseteq \mathcal{F}_{rn} && \text{or}\\
&\quad \exists\, l \in (S_2 \setminus S_1) \cap \mathcal{Z} \cdot (e, I, T) \in \mathcal{ZO}(l) \wedge f \in I \wedge I, T \subseteq \mathcal{F}_{rn} \wedge e \in \mathcal{Z} && \text{or}\\
&\quad \exists\, l \in (S_1 \setminus S_2) \cap \mathcal{Z} \cdot (e, D, V) \in \mathcal{ZO}(l) \wedge f \in Y \wedge D, V \subseteq \mathcal{F}_{rn} \wedge e \in \mathcal{Z} && \text{or}\\
&\quad \exists\, obl(o, d, f) \in S_2 \cdot d \in S_2 && \text{or}\\
&\quad \exists\, obl(o, d, f) \in S_2 \cdot d\, \text{GR}^{\omega}(S_1, \{e_{ex}\}) && \text{or}\\
&\quad \exists\, obl(o, f) \in S_2 \cdot \text{GR}^{\omega}(S_1, \{e_{ex}\}) \cap \mathcal{E}_{dis} \neq \emptyset && \text{or}\\
&\quad \exists\, nobl(o, d, f) \in S_2 \cdot o \in S_2 && \text{or}\\
&\quad \exists\, nobl(o, d, f) \in S_2 \cdot o \in \text{GR}^{\omega}(S_1, \{e_{ex}\}) && \text{or}\\
&\quad \exists\, nobl(o, f) \in S_2 \cdot \text{GR}^{\omega}(S_1, e_{ex}) \cap \mathcal{E}_{dis} \neq \emptyset\}
\end{aligned}
$$

Termination has the same arguments as initiation. It uses these to terminate obligations corresponding to legitimate expectations. It also terminates the initiating consequences of terminated expectations an terminate consequences of initiated expectation. Finally it terminates satisfied and violated obligations that are based on landmarks.

$$
\begin{aligned}
&\text{O-TERM}(S_1, S_2, e_{ex}) = \{f \in S_2 \mid \\
&\quad \exists\, l \in (S_1 \setminus S_2) \cap \mathcal{Z} \cdot (f, I, T) \in \mathcal{ZO}(l) \wedge I, T \subseteq \mathcal{F}_{rn} && \text{or}\\
&\quad \exists\, l \in (S_1 \setminus S_2) \cap \mathcal{Z} \cdot (e, I, T) \in \mathcal{ZO}(l) \wedge f \in I \wedge I, T \subseteq \mathcal{F}_{rn} \wedge e \in \mathcal{Z} && \text{or}\\
&\quad \exists\, l \in (S_2 \setminus S_1) \cap \mathcal{Z} \cdot (e, I, T) \in \mathcal{ZO}(l) \wedge f \in I \wedge I, T \subseteq \mathcal{F}_{rn} \wedge e \in \mathcal{Z} && \text{or}\\
&\quad \exists\, obl(o, d, f) \in S_2 \cdot (o \vee d \in S_2) && \text{or}\\
&\quad \exists\, obl(o, f) \in S_2 \cdot o \in S_2 && \text{or}\\
&\quad \exists\, obl(o, f) \in S_2 \cdot \text{GR}^{\omega}(S_1, \{e_{ex}\}) \cap \mathcal{E}_{dis} \neq \emptyset && \text{or}\\
&\quad \exists\, nobl(o, d, f) \in S_2 \cdot (o \vee d \in S_2) && \text{or}\\
&\quad \exists\, nobl(o, f) \in S_2 \cdot o \in S_2 && \text{or}\\
&\quad \exists\, nobl(o, f) \in S_2 \cdot \text{GR}^{\omega}(S_1, \{e_{ex}\}) \cap \mathcal{E}_{dis} \neq \emptyset\}
\end{aligned}
$$

[7] Note, that while the legitimate expectation is still valid, the consequences and obligations might change.

Now we combine all of this into a single state transition function, $\mathrm{TR} : \Sigma \times \mathcal{E}_{ex} \rightarrow \Sigma$. This function generates the new state from the current state and an exogenous event as follows:

$$
\begin{aligned}
\mathrm{TR}(S, e_{ex}) = {} & (\text{S-TR}(\text{E-TR}(S, e_{ex})) \cup \\
& \text{O-INIT}(\text{E-TR}(S, e_{ex}), \text{S-TR}(\text{E-TR}(S, e_{ex})), e_{ex})) \\
& \setminus \ \text{O-TERM}(\text{E-TR}(S, e_{ex}), \text{S-TR}(\text{E-TR}(S, e_{ex})), e_{ex})
\end{aligned}
$$

Traces and Models Using these final transformation function we can conclude this section by defining traces and their evaluation.

An ordered trace is defined as a sequence of exogenous events $\langle e_0, e_1, \ldots, e_n \rangle \;\; e_i \in \mathcal{E}_{ex}, 0 \leq i \leq n$. Its evaluation starting with the initial state of the legal framework is the sequence: $\langle S_0 = \Delta^*, S_1, \ldots S_{n+1} \rangle$ with:

$$
S_{i+1} = \begin{cases} \text{E-TR}(\Delta^*, e_i,) & \text{if } \texttt{live} \in S_i \\ S_i & \text{otherwise.} \end{cases}
$$

3 Case-Study

Having defined the syntax and the semantics for our extended INSTAL model, we now demonstrate the framework on the dentist case as explained in the introduction. We slightly extended the example to demonstrate all the features of our model. Fig 2 shows the formal model of the case-study. We used the same numbering as Fig 1 were we provided an overview of the syntax of INSTAL's formal model.

Here we give an informal description of the modelling process.

The dentist (buyer) can start the purchasing negotiations, provide details and plans, explicitly withdraw from the sale, buy the property and repay any costs the seller incurred to legitimately believing the dentist would purchase the property. The seller can make implicitly requested alterations. These are represented by the exogenous events (1.a): `startNeg`, `provideDetails`, `providePlans`, `makeChanges`, `withdraw`, `buy` and `repay`, and one dissolution event `endNegotiation` .

The occurrence of the dentist's exogenous event `startNeg` results in the occurrence of the institutional action `takeClient` (3.a). When the legitimate expectation `buyExp` (2.a.vii) is part of the state, the occurrence of `makeChanges` results in the generation of `buyerChanges` to indicate that the (expected) buyer is liable for the costs (3.b). The occurrence of `takeClient` initiates `client` and `interest` (4.i.i), to indicate that buyer is moderately interested in the property. The occurrence of the events `provideDetails` and `providePlans` result in domain fluents `detail` and `plans` to be initiated (4.i.ii-iii). When `interest`, `details`, `plans` are part of the state, the system reaches the landmark `commitment` (5.a). This triggers the legitimate expectation `buyExp` (6.a) to indicate that the seller has sufficient ground to assume that the buyer is going to proceed with the purchase. In turn, and this an extension of the original case, this creates a further legitimate expectation `noOtherBuyer` (6.b) on the seller not to look for other buyers, on the condition that the buyer has not `withdrawn` from the sale. Without this withdrawn condition, `noOtherBuyer` would be recreated once it was

DentistCase $= \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \mathcal{LC}, \mathcal{ZC}^{\uparrow}, \mathcal{ZC}^{\downarrow}, \mathcal{ZO}, \Delta^* \rangle$, where

1. $\mathcal{E} = \mathcal{E}_{ex} \cup \mathcal{E}_{legal}$ with $\mathcal{E}_{legal} = \mathcal{E}_{act} \cup \mathcal{E}_{viol}$ s.t.
   (a) $\mathcal{E}_{ex} = \{\texttt{endNegotiation}, \texttt{startNeg}, \texttt{provideDetails}, \texttt{providePlans}, \texttt{makeChanges}, \texttt{withdraw}, \texttt{buy}, \texttt{repay}, \texttt{endNegotiation}\}$
   (b) $\mathcal{E}_{act} = \{\texttt{takeClient}, \texttt{buyerChanges}\}$
   (c) $\mathcal{E}_{viol} = \{\texttt{implicitCD}\} \cup$
   $\{\texttt{viol}(\texttt{perm}(\texttt{e})) \mid e \in \mathcal{E}_{ex} \cup \mathcal{E}_{legal}\} \cup \{\texttt{viol}(\texttt{pow}(\texttt{e})) \mid e \in \mathcal{E}_{act}\}$
   (d) $\mathcal{E}_{dis} = \{\texttt{endNegotiation}\}$
2. $\mathcal{F} = \mathcal{F}_{ev} \cup \mathcal{F}_{st}$ with
   (a) $\mathcal{F}_{ev} = \mathcal{W} \cup \mathcal{P} \cup \mathcal{O} \cup \mathcal{D} \cup \mathcal{V} \cup \mathcal{NO}$ s.t.:
      (i) $\mathcal{W} = \{\mathrm{pow(e)} \mid e \in \mathcal{E}_{legal}\}$
      (ii) $\mathcal{P} = \{\mathrm{perm(e)} \mid e \in \mathcal{E}\}$
      (iii) $\mathcal{O} = \{\texttt{obl}(\texttt{bought}, \texttt{walkOut}), \texttt{obl}(\texttt{repay}, \texttt{debt})\}$
      (iv) $\mathcal{D} = \{\texttt{interest}, \texttt{details}, \texttt{client}, \texttt{plan}, \texttt{buyerCost}, \texttt{sold}, \texttt{withdrawn}\}$
      (v) $\mathcal{NO} = \{\texttt{nobl}(\texttt{withdraw}, \texttt{implicitCD})\}$
      (vi) $\mathcal{V} = \{\texttt{walkOut}, \texttt{debt}\} \cup \{viol(e) \mid e \in \mathcal{E}_{viol}\}$
      (vii) $\mathcal{F}_{st} = \mathcal{Z} \cup \mathcal{L}$ s.t.
         A. $\mathcal{L} = \{\texttt{commitment}, \texttt{bought}\}$
         B. $\mathcal{Z} = \{\texttt{buyExp}, \texttt{noOtherBuyer}\}$
3. $\mathcal{G} :: \mathcal{X} \times \mathcal{E} \to 2^{\mathcal{E}_{legal}}$
   (a) $\langle \emptyset, \texttt{startNeg} \rangle \to \{\texttt{takeClient}\}$
   (b) $\langle \{\texttt{buyExp}\}, \texttt{makeChanges} \rangle \to \{\texttt{buyerChanges}\}$
4. $\mathcal{C} :: \mathcal{X} \times \mathcal{E} \to 2^{\mathcal{F}} \times 2^{\mathcal{F}}$ s.t.
   (i) $\mathcal{C}^{\uparrow}(X, e)$
      i. $\langle \emptyset, \texttt{takeClient} \rangle \to \{\texttt{client}, \texttt{interest}\}$
      ii. $\langle \emptyset, \texttt{provideDetails} \rangle \to \{\texttt{details}\}$
      iii. $\langle \emptyset, \texttt{providePlans} \rangle \to \{\texttt{plans}\}$
      iv. $\langle \emptyset, \texttt{buyerChanges} \rangle \to \{\texttt{buyerCosts}\}$
      v. $\langle \emptyset, \texttt{buy} \rangle \to \{\texttt{sold}\}$
      vi. $\langle \emptyset, \texttt{withdraw} \rangle \to \{\texttt{withdrawn}\}$
      vii. $\langle \{\texttt{buyerCosts}\}, \texttt{withdraw} \rangle \to \{\texttt{obl}(\texttt{repay}, \texttt{debt})\}$
   (ii) $\mathcal{C}^{\downarrow}(X, e)$
      i. $\langle \emptyset, \texttt{repay} \rangle \to \{\texttt{buyerCosts}\}$
5. $\mathcal{LC} :: \mathcal{X} \to \mathcal{L}$
   (a) $\{\texttt{interest}, \texttt{details}, \texttt{plans}\} \to \texttt{commitment}$
   (b) $\{\texttt{sold}\} \to \texttt{bought}$
6. $\mathcal{ZC}^{\uparrow} :: \mathcal{X} \to \mathcal{Z}$
   (a) $\{\texttt{commitment}\} \to \texttt{buyExp}$
   (b) $\{\texttt{buyExp}, \neg\texttt{withdrawn}\} \to \texttt{noOtherBuyer}$
7. $\mathcal{ZC}^{\downarrow} :: \mathcal{X} \to \mathcal{Z}$
   (a) $\{\texttt{withdrawn}\} \to \texttt{noOtherBuyer}$
8. $\mathcal{ZO} :: \mathcal{Z} \to 2^{\mathcal{O}} \cup 2^{\mathcal{NO}} \times 2^{\mathcal{F}_{rn}} \times 2^{\mathcal{F}_{rn}}$
   (a) $\texttt{buyExp} \to \langle \{\texttt{obl}(\texttt{bought}, \texttt{walkOut}), \texttt{nobl}(\texttt{withdraw}, \texttt{implicitCD})\},$
   $\{\texttt{perm}(\texttt{buyerChanges}), \texttt{pow}(\texttt{buyerChanges})\}, \emptyset \rangle$
   (b) $\texttt{noOtherBuyer} \to \langle \{\texttt{nobl}(\texttt{takeClient}, \texttt{walkOut})\}, \emptyset, \emptyset \rangle$
9. $\Delta^* = \{\texttt{live}\} \cup \{\texttt{perm}(\texttt{e}) \mid e \in \mathcal{E}_{ex}\} \cup \{\texttt{pow}(\texttt{takeClient}), \texttt{perm}(\texttt{takeClient})\}$

Fig. 2. The formal model for the case-study discussed in Sec 3

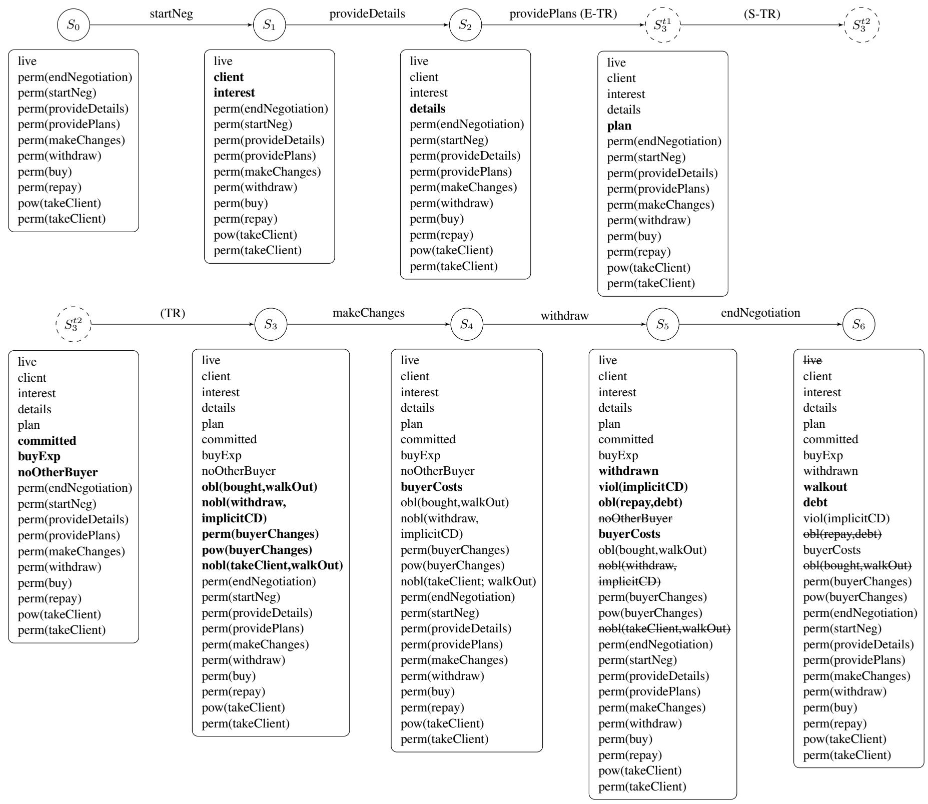

Fig. 3. A visualistion of the evaluation of a trace of the Dentist case

terminated. The initiation of the legitimate expectation `buyExp` results in the initiation of one obligation to reach the landmark `bought` and negative obligation to deter the buyer from withdrawing (8.a). It also initiates the power and permission for the seller to make changes to the property at the expense of the buyer. The obligations use a violation fluent `walkOut` and violation event `implicitCD` respectively. This is mainly done to demonstrate the different possibilities but also to be able to provide a course of action in case the buyer withdraws. The landmark `bought` is reached when `sold` becomes true in the state (5.b). This is the case after the occurrence of `buy` (4.i.v). The legitimate expectation `noOtherBuyer` creates the negative obligation on `takeClient`. The occurrence of the event results in the violation fluent `walkOut` if the obligation gets violated (8.b). The `noOtherBuyer` expectation is terminated once `withdrawn` is part of the state (7.a). The event `withdraw` makes this happen (4.i.vi). If the seller incurred any cost due to a legitimate expectation, the dentist will be obliged to `repay` these costs when he `withdraws` (4.i.vii) from the sale. In the initial state we give permission to all exogenous events. Power and permission is given to `takeClient` (9).

Figure 3 shows the evaluation of the trace: ⟨`startNeg`, `provideDetails`, `providePlans`, `makeChanges`, `withdraw`, `endNegotiation`⟩. For the transition from state S_2 to S_3 we have also displayed the intermediate states (dashed circles). Additions to the state are marked in bold while deletions are struck-out. The trace is marked above the arrows.

Let us highlight the transition from state S_2 to S_3 when providePlan occurs. Applying the first step of the transformation function we obtain:
$S_3^{t1} = \text{E-TR}(S_2, \texttt{providePlan}) = S_2 \cup \{\texttt{plan}\}$. Following this up with determining landmarks and legitimate expectations, we obtain $S_3^{t2} = \text{S-TR}(S_3^{t1}) = S_3^{t1} \cup \{\texttt{committed}, \texttt{buyExp}, \texttt{noOtherBuyer}\}$. Finally, by adding the necessary obligation and initiation and terminating the expectations' consequences, we obtain $S_3 = S_3^{t2} \cup \{\texttt{obl}(\texttt{bought}, \texttt{walkOut}), \texttt{nobl}(\texttt{withdraw}, \texttt{implicitCD}),$
$\texttt{perm}(\texttt{buyerChanges}), \texttt{pow}(\texttt{buyerChanges}), \texttt{nobl}(\texttt{takeClient}, \texttt{walkOut})\}$.

4 Related Work

To the best of our knowledge no other formal model of legitimate expectations can be found. Widening this search to models of expectation and related concepts, more research was found. Two major directions for research can be distinguished.

The first of these is concerned with the formal modelling of expectations. Castelfranchi [1] for example approaches expectations from a cognitive science perspective and suggests a formal model in which expectations are modelled as a mental objects of agents. His model demonstrates how expectations alter desires and intentions in the decision making process of agents. Whereas Castelfranchi mainly focuses on the effects of existing expectations on the agent reasoning, Trân et al. [13] take a step back and formulate a formal model which allows the analysis of the generation of expectations as mental objects of agents based on different perceptions from the environment. Although these works lay important foundations for the approach presented here (e.g. proposing to view expectations as results from observed actions) neither of them transfers their ideas to the legal domain. One paper which takes a step in this direction is [8]. Gärdenfors uses non-monotonic logic to model expectations as explicit premises in logical arguments. He applies them to argumentation theory and places them in a legal context. Similar to [13] Gärdenfors stops with the generation of expectation, i.e. missing the component of legal consequences (e.g. legitimate expectations) resulting from these generated expectations.

The second main stream of research comprises formal models of concepts similar or related to legitimate expectations. Here we want to highlight [6,2]. Feigenbaum et al. [6]. present a formal model of accountability in which they deduce information about the responsibility for faults in security systems. In contrast to our work which focuses on reasoning about legitimate expectations resulting from the actions of entities in a live system, Feigenbaum et al. focus on an a priory analysis of their systems. They use automatic enforcement for all states which this analysis identified as undesired (i.e. security hazard), preventing any unexpected problems from occurring. Cavedon et al. [2] present a formal model of "social commitments". Similar to us, their social model includes the notion that from a legal perspective obligations can be created even if no explicit contract has been made. Rather than focusing on the evolution of states as we do, to them these obligations result from prior informal group negotiations in which a group goal was defined (i.e. from some form of prior agreement) which the agents are expected to follow. In our model, legitimate expectations (and the linked obligations) do not result from informal discussion and agreed upon goals, but rather from modelled interactions.

In our approach we have opted for modelling all the components of legal framework, e.g. different events, their consequences, the different types of fluents and the different state conditions, as individual components in our mathematical model. While it is possible to describe them as a logic program, first order logic set of formula or an event calculus [11] description, we believe that some of the granularity would be lost or would require the designer to provide extra formuli/rules for each framework. Some of these will be framework-dependent while others are not. While we believe this is acceptable for a computational tool, we feel this is undesirable for a formalisation.

5 Conclusion and Future Work

In this paper we presented an extension of the INSTAL framework allowing us to model and reason about legitimate expectations. We presented a detailed description of syntax and semantics of this extended formal model. We demonstrated our approach with the help of a Japanese private law case study. As pointed out before, the presented formal model is solely based on mathematical constructs, thereby avoiding formalism specific side-effects. This allowed us to specify the concept of legitimate expectations as well as the components required for it independently of the afterwards chosen implementation language. Thus, using our formal model, designers can pick a specification language of their choice to implement a corresponding computational model.

We identify several issues for future work. The first concerns the extension of the ideas set out in this paper. We focused on states and landmarks for reasoning about legitimate expectations. From a users perspective it might be of interest to reason about the history of events that lead to legitimate expectations. Our framework currently uses sequences of exogenous events as input for the state transition of the framework which could act as historic information.

Furthermore we plan to introduce the concept of scenes which define sequences of landmarks participants should or should not reach within the legal framework. This could for example be done with the help of obligations. The introduction of scenes would allow us to conceptually link several landmarks to reason about more complex processes and legal concepts.

Additionally we would like to further extend the concept of obligations presented in this paper, making it possible for them to consider that these could continue to exist even after their violation rather than automatically terminated.

In the future work we would like to develop a computational model based on the formal specification described in the paper. This computational model would allow the complete computational analysis of sample cases. Based on our formal specifications, this computational model could be implemented using, for example, the event calculus, situation calculus, logic programming or, like the original INSTALframework answer set programming. Irrespective of the computational back-end, providing a more dedicated language, like for example an extension the action language associated with INSTAL, for the specification of legal framework would be beneficial.

The final direction of future research is that of software development. We aim to build interfaces and tools more suited to a general audience. The idea is to provide user interfaces and visualisation that allow users without a background in legal modelling to use our framework to reason about sample cases they are interested in.

References

1. C. Castelfranchi. For a systematic theory of expectations. In S. Vosniadou, D. Kayser, and A. Protopapas, editors, *Proceedings of the European Cognitive Science Conference 2007*. Taylor & Francis, 2007.
2. L. Cavedon, A. Rao, and G. Tidhar. Social and individual commitment. In L. Cavedon, A. Rao, and W. Wobcke, editors, *Intelligent Agent Systems Theoretical and Practical Issues*, volume 1209 of *LNCS*. Springer, 1997.
3. 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, *Computational Logic for Multi-Agents*, volume 4371 of *LNAI*, pages 60–79. Springer, May 2006.
4. M. De Vos, J. Padget, and K. Satoh. Legal modelling and reasoning using institutions. In T. Onoda, D. Bekki, and E. McCready, editors, *New Frontiers in Artificial Intelligence (JSAI-isAI 2010 Workshops, selected papers)*, pages 129–140. Springer, 2011.
5. V. Dignum. *A Model for Organizational Interaction*. PhD thesis, Utrecht University, 2004.
6. J. Feigenbaum, A. D. Jaggard, and R. N. Wright. Towards a formal model of accountability. In *Proceedings of the 2011 workshop on New security paradigms workshop*, pages 45–56. ACM, 2011.
7. H. M. Flechtner. Transcript of a workshop on the sales convention: Leading cisg scholars discuss contract formation, validity, excuse for hardship, avoidance, nachfrist, contract interpretation, parol evidence, analogical application, and much more. *Journal of Law & Commerce*, 18:191–258, 1999.
8. P. Gärdenfors. The role of expectations in reasoning. In *International Conference Logic at Work on Knowledge Representation and Reasoning Under Uncertainty, Logic at Work*, pages 1–16. Springer, 1994.
9. A. J. I. Jones and M. Sergot. On the characterization of law and computer systems: the normative systems perspective. In *Deontic logic in computer science: normative system specification*, pages 275–307. John Wiley and Sons Ltd., 1993.
10. F. Kessler and E. Fine. Culpa in contrahendo, bargaining in good faith, and freedom of contract: A comparative study. *Harvard Law Review*, 77(3):401–449, 1964.
11. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95, 1986.
12. D. Okouya and V. Dignum. Operetta: a prototype tool for the design, analysis and development of multi-agent organizations. In *AAMAS (Demos)*, pages 1677–1678. IFAAMAS, 2008.
13. B. V. Trân, J. Harland, and M. Hamilton. A combined logic of expectation & observation. a generalisation of BDI logics. In J. A. Leite, A. Omicini, L. Sterling, and P. Torroni, editors, *DALT 2003, Revised Selected and Invited Papers*, volume 2990 of *LNCS*, pages 155–172. Springer, 2003.
14. R. von Jhering. Culpa in contrahendo, oder Schadensersatz bei nichtigen oder nicht zur Perfektion gelangten Verträgen. In *Jahrbücher für die Dogmatik der heutigen Römischen und Deutschen Privatrechts I*. 1861.