A Normative Agent-based Model for Sharing Data in Secure Trustworthy Digital Market Places

Deljoo, A.; van Engers, T.; van Doesburg, R.; Gommans, L.; de Laat, C.

Published in:
ICAART 2018

DOI:
10.5220/0006661602900296

A Normative Agent-based Model for Sharing Data in Secure Trustworthy Digital Market Places

Ameneh Deljoo[1], Tom van Engers[2], Robert van Doesburg[2], Leon Gommans[1,3] and Cees de Laat[1]
[1]*Institute of Informatics, University of Amsterdam, Amsterdam, The Netherlands*
[2]*Leibniz Center for Law, University of Amsterdam, The Netherlands*
[3]*Air France-KLM, The Netherlands*

Keywords: Normative Agent Based Model, Beliefs Desire and Intentions Agent, Simulation, Agent Based Modeling, Jadex.

Abstract: Norms are driving forces in social systems and governing many aspects of individual and group decision-making. Various scholars use agent based models for modeling such social systems, however, the normative component of these models is often neglected or relies on oversimplified probabilistic models. Within the multi-agent research community, the study of norm emergence, compliance and adoption has resulted in new architectures and standards for normative agents. We propose the N-BDI* architecture by extending the Belief-Desire and Intention (BDI) agents' control loop, for constructing normative agents to model social systems; the aim of our research to create a better basis for studying the effects of norms on a society of agents. In this paper, we focus on how norms can be used to create so-called Secure Trustworthy Digital Marketplaces (STDMPs). We also present a case study showing the usage of our architecture for monitoring the STDMP-members' behavior. As a concrete result, a preliminary implementation of the STDMP framework has been implemented in multi-agent systems based on Jadex.

1 INTRODUCTION

Norms[1] are an important key to understanding the function of societies of agents, such as human groups, teams, and communities; they are a ubiquitous but invisible force governing many societies. Bicchieri (Bicchieri, 2005) describes human norms as:"the language a society speaks, the embodiments of its values and collective desires, the secure guide in the uncertain lands we all traverse, the common practices that hold human groups together."

A normative agent refers to an autonomous agent who understands and demonstrates normative behavior. Such agents must be able to reason about the norms with which they should comply and occasionally violate them if they are in conflict with each other or with the agent's private goals (Luck et al., 2013). For individual agents, reasoning about social norms can easily be supported within many agent architectures. Dignum (Dignum, 1999) defines three layers of norms (private, contract, and convention) that can be used to model norms within the BDI framework. Creating realistic large-scale models of social systems is impaired by the lack of good general purpose computational models of social systems. These models help to analyze and reason about the actions and interactions of members' of such societies of agents that are bound by norms.

A real-world social scenario where these concerns clearly apply is in business relationships. In our research, we are focusing on environments in which agents may agree on cooperation efforts, involving specific interactions during a certain time frame. This way, agents compose any organizations (in this paper STDMPs), which is regulated by the specific norms agreed upon. Agents may represent different business units or enterprises, which come together to address new market opportunities by combining skills, resources, risks, and finances that no partner can alone fulfill (Dignum and Dignum, 2002). Any cooperation activity requires trust between the involved partners. When considering open environments, previous

[1]Norms play an important role in open artificial agent systems; they have been said to improve coordination and cooperation. As in real-world societies, norms provide us a way to achieve social order and give raise for expectations thus controlling the environment and making it more stable and predictable.

Deljoo, A., Engers, T., Doesburg, R., Gommans, L. and Laat, C.
A Normative Agent-based Model for Sharing Data in Secure Trustworthy Digital Market Places.
DOI: 10.5220/0006661602900296
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 1*, pages 290-296
ISBN: 978-989-758-275-2

performance records of potential partners may not be assessable. In this paper a Trusted Electronic Institutions agent (TEI) propose to mimic real-world institutions, by regulating the interactions between agents. The TEI agent concept is a coordination framework that facilitating the establishment of contracts and providing a level of trust by offering an enforceable normative environment. The TEI agent encompasses a set of norms regulating the environment.

This normative environment evolves as a consequence of the establishment of agents' agreements formalized in contractual norms. Therefore, an important role of the TEI agent is to monitor and enforce, through appropriate services, both predefined institutional norms and those formalizing contracts that result from a negotiation process. Agents rely on the TEI agent to monitor their contractual commitments.

Previously, we presented the elements of a normative architecture and the extension of the BDI agents for sharing data case studies (Deljoo et al., 2016; Deljoo et al., 2017). This paper describes an extended BDI architecture for constructing and simulating normative effects on a social system such as STDMPs.

The aim of our research is to create a general purpose Agent-Based Modeling (ABM) and simulation system for studying how norms can be used to create STDMPs and how we can monitor the effects of norms on such system where each member's of society are self-governed autonomous entities and pursue their individual goals based only on their beliefs and capabilities (Gouaich, 2003).

This paper presents a study showing the relative contribution of social norms on creating STDMPs and predicting the impact of norms on the member's of the STDMPs. Our proposed model to simulate STDMPs members' behaviors and a detailed description is provided in Section 2. The norm description in the STDMPs case study, accepting the partners' request to share acceptable data with the partners after checking compliancy with the General Data Protection Regulation (GDPR), presented in Section 4. Section 3 presents the STDMP scenario and primarily implementation of our model.

Although this paper focuses on STDMPs, we believe our architecture is sufficiently general to study a variety of social scenarios. We conclude the paper with the related work on normative agents and different normative architectures.

2 N-BDI*

In a previous paper, we have presented an extension of the BDI agents framework (Deljoo et al., 2017). In the previous extension, agents need to select the most appropriate plan when they have partial observations. To enable this we extended the agent planner component by integrating probabilities and utility into the BDI agent's planner component. Also in our further extension described in this paper which we have called the normative BDI* (N-BDI*) framework, the agents have the ability to select the most appropriate plans based on the highest expected utility that fulfills the expectations of agents as well (see Algorithm 2).

We complete our extension in this paper by introducing N-BDI* architecture. The N-BDI* framework is inspired by the nBDI framework from (Criado et al., 2010). Their framework, like ours, is an extension of the basic BDI agents. The nBDI framework consists of two functional contexts: the Recognition Context (RC), which is responsible for the norm identification process; and the Normative Context (NC), which allows agents to consider norms in their decision making processes. One of the differences of our N-BDI* framework compared to the nBDI is the way the agents select the most appropriate plan. In our framework we assign utility to each plan and select the one that maximizes the utility. In the nBDI framework, the authors did not consider the utility in the agent's planner.

In the nBDI framework the agent's intention is equal to the action of agent but in reality, which is reflected in our agent framework,the intention is the different component from the action component. After selecting a plan, the agent's intention becomes to execute that plan. In our architecture, before executing, the agent checks its (institutional) *Power* to execute the selected plan. The *ability* to actually execute the plan is in social reality, and can be checked by the agent by monitoring the effects of the selected action(s), even in case they fail, and comparing these effects with the intended effects. The explicit notion of institutional powers and social abilities are not addressed in the nBDI.

Belief revision in nBDI is based on the received feedback from the environment while in our extended N-BDI* framework the belief revision happens with a higher frequency, for example when the agent's sensor received data or when the most appropriate plan has been selected.

Also, they considered two types of norms (Constitutive and Deontic norms). In our work, we have a norm representation based upon the work of Hohfeldian (Doesburg and Engers, 2016). The GDPR norms described in this paper are represented in this way. In Figure 1 we depict our N-BDI* architecture. The deliberation cycle of the N-BDI* agent model is presented in Algorithm 1.In our terminology, beliefs

Algorithm 1: Modified control loop for the extended BDI agent (N-BDI*), where O= observation, B= Belief set, G= Goal set, P= Plan set, and A_p= Actions.

Given an agent $\{O, B, G, P, A_p, Norms\}$
repeat
  $O := Observe(O + Norms);$
  $B := Revise(B, O);$
  $G := Generate\ G\ (B);$
  $P := \forall g \in G \rightarrow generate\ P(B, G);$
  $P := Calculte\ U_P\ \forall\ p \in P(B, G, P);$
  $PrefP := Update\ P\ to\ PrefP(B, G, A_p, P);$
  $B := revise(B, Pref);$
  $A_p := (norms(Power), Allowed?);$
  $take\ (A_p);$
until *forever*;

Algorithm 2: Select Plan.

input : (sub)Goal, Set of plans ($p \in P$), the Probability of each plan, Value, Norms
output: Selected P, Plan that has the best utility.

$SelectedPlan(P) := null;$
for $p \in P$ **do**
  $CalculateRiskViolation(p) = Value \times Pr(p);$
  $CalculateCost(p);$
  $U(p) := Pr(p) \times U(s);$
  $PU(P) := setofPU(p);$
end
$PrefP := argmaxPU(P);$
$SelectedPlan(P) := PrefP;$
return $SelectedPlan(P)$

encode the agent's knowledge about the world or its mental states, which the agent holds to be true (that is, the agent will act upon them while they continue to hold). Goals are equated with"desires" and commitments to plans with "intentions". We view intentions as commitments to new beliefs or to carrying out certain plans or pursuing new goals and actions in the future. As stated above (Algorithm 1), the agent has a set of plans (P), where each is primarily characterized by the goals (G) and a set of possible actions(A_p). In other words, each plan consists of an invocation which is the event that the plan responds to and may contribute to the G.
The N-BDI* agent belief set (B) contains the norms and observations. Agent sets up the agent's G based on these two factors. The N-BDI* agent after producing the set of plan and select the best plan based on the utility, for each plan will calculate a risk of violation and cost corresponding to the selected plan (see Algorithm 2). The selected plan becomes the current intention of the agent. Before selecting the appropriate plan, the agent calculates the Risk of violation and cost for each plan, and remove the plans that have the high association risk or cost from the plan sets[2]. Then, inspecting A_p to find all the action recipes which have among their effects a goal in G. Then, the agent will examine a power of itself to execute the A_p. As we mentioned earlier, our goal to use N-BDI* framework to model and simulate the effect of different norms on STDMPs. Our architecture contains three phases: recognition, adoption and compliance. In the first recognition part, the beliefs of an agent revise and develop. This step equals to the

[2]In some cases, the agent selects the plan which has a low violation risk value and calculates the violation penalty as well but in this paper we only consider that the agent eliminates the violating plans from the plan sets before selecting the appropriate plan

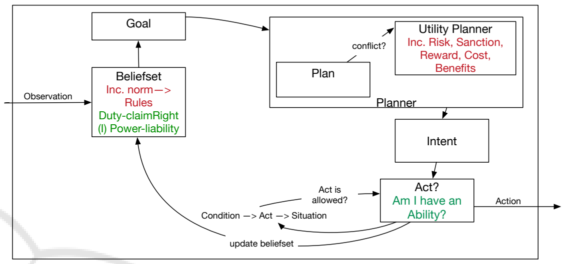

Figure 1: N-BDI* architecture.

RC in nBDI. During the adoption (equals to the NC in nBDI architecture), the agent commences actions. The norm violation can happen during the adoption phase (Luck et al., 2013). The compliance phase is used to simulate the situation when the agent really starts executing the action. We add another part to the normative phase in our architecture that called monitoring. In monitoring phase, the agent will reason about the action and consequence of the actions on the society.

3 SECURE TRUSTWORTHY DIGITAL MARKETPLACES (STDMPS)

Secure Trustworthy Digital Marketplaces (STDMPs) a concept developed for data sharing in an open world, while protecting the interests of the subjects whose data is exchanged, the controllers of their data and the rights of the subjects who created the data transformations and the subjects that have an interest in applying those transformations to that data. To reduce the complexity of case study, we only consider three types of STDMP agents:
Agents:

- LH: license holding agents who hold data and can provide data to the market (the STDMPs);

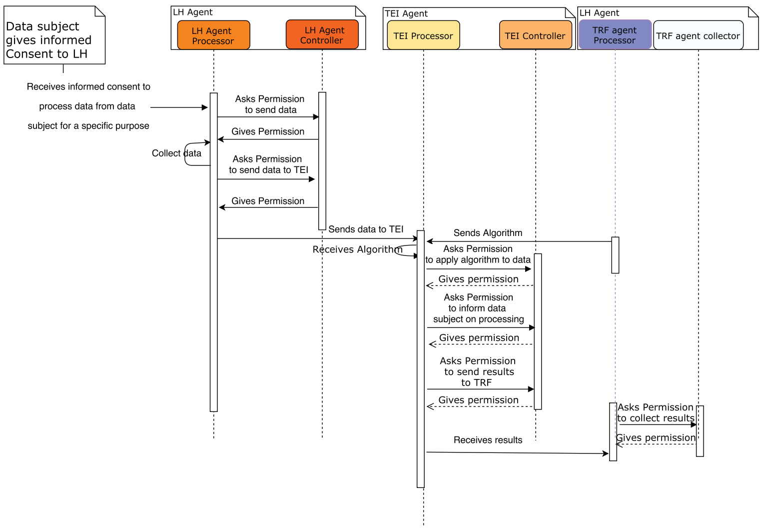

Figure 2: Message sequences diagram among STDMP's agent.

- TEI agents who monitor the members' behavior;
- TRF: Transformation agents who hold the algorithms, have a need for the LH's data that can be provide through the STDMPs.

The STDMPs society is a regulated environment which includes the expression and use of regulations of different sorts: from actual laws and regulations issued by governments, to policies and local regulations issued by managers, and to social norms that prevail in a given community of users.

3.1 Scenario

As we mentioned before, STDMP consists of three main agents (LH, TEI, TRF). Each of these agents can take the role of processor and controller. These are two of the roles distinguished by the GDPR; the data subject, the controller and the processor. Following our architectural principle that one actor role comes with its own belief set and plan operators, we have components for each individual role. The processor component is responsible for processing data on behalf of the controller, which includes making data available, while the controller component is responsible to check requests for data against the GDPR and the informed consent given by the data subjects. A secure and trustworthy data-sharing platform among hospitals, third parties and data analyst looking for the most effective interventions based on patient data is one good example of where STDMP protecting the interests of the stakeholders and preventing data protection infringements can contribute.

In order to explain how STDMPs help to implement GDPR and other requirements derived from norms, we present a simplified scenario. The LH's agent in its role as controller receives informed consent for processing personal data from a data subject for a specific (set of) purpose(s). The LH's in his processor role asks permission from its controller to collect data and send it to the TEI. After giving the permission, the LH's processor collects and sends data to the TEI. The TEI agent asks the TRF agent to send its algorithm to the TEI agent to analyze data. Be aware that since these data transformation functions, e.g. data-mining algorithms, may be the protected norms e.g. copy right law, the algorithms do not contain personal data and therefore are not subjected to the GDPR, although in other cases we may have to apply norms regulating access as well. The TRF agent sends the algorithm after getting the permission from its controller. The TEI agent combines the data with the algorithm and sends the result to the TRF. The details of this scenario are visualized in a UML sequence diagram, see Figure 2.

In the scenario, depicted data processing requires protection of the interests of the stakeholders involved and compliance to the GDPR. Because the liabilities that may follow from not meeting the demands from each of the parties involved, such data processing infrastructure will depend on trust between parties. The TEI acts, as its name suggests as a trusted third party. The behavior of this agent should be completely determinant and transparent, and no human interference is part of the actions of that agent. In the scenario presented, the purpose of using the requested data must be fitted into the LH's interest and the request must be adhered to the GDPR rules.

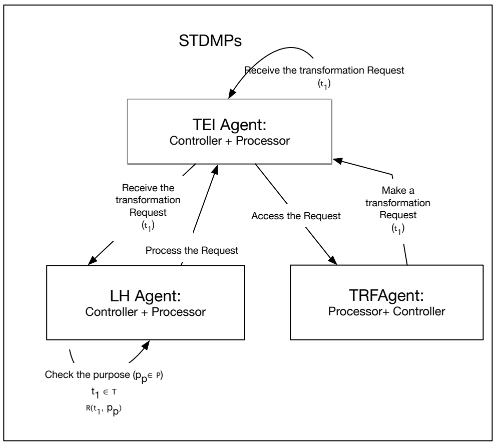

Figure 3: STDMPs Scenario.

We formalized the mentioned scenario as follow:
Context= $(A_1 \in LH, A_2 \in TRF, Contract)$;
and,
Contract=Set of Permissions.
The TEI agent receives the A_1's transformation request(t_1) and check the eligibility of (t_1) by checking the condition of $t_1 \in T$, where T denotes a set of transformations. Note that, in the STDMP, the LH's controller defined the set of licenses for each data set. Licenses have a defined a set of conditions on using data. We visualize the scenario in Figure 3. In the following, we present a norm definition to express the contract among STDMPs' members.

4 NORM

In this section, we present a recent general model of norms (Oren et al., 2009) that distinguishes some general normative concepts shared by some existing work on norms and normative systems (Farrell et al., 2005). In (Oren et al., 2009), a norm n is modeled as a tuple:
Definition 1 (norm). A norm is defined as a tuple
$n = (role, normtype, conditions, action)$
such that:

- role: indicates the organizational position;
- norm type is one of the four modal verbs "can" (which we formalize as a power), "can not" (disability), "must" (duty) and "must not" (which is not the same as a no-right, but the obligation to not do something!); In this paper, we define the contract as a set of permissions that when acted upon may result in other normative relations, including duties. Permissions are equal to the Hohfeldian concept *power*.
- condition[3]: describes when and where the norm holds (norm adoption);
- action:action specifies the particular action to which the normative relation is assigned (norm adoption);

Example: Consider the norm, **NormCollectData** that describes the permission to collect ***personal data*** from the ***data subject***, where the collector is the LH agent consisting of two sub-agents (LH's controller, LH's processor).

1. [Role: LH's controller][Normative relation: **Power**][condition: "iff " legitimate purpose of collecting data is specified explicit **"AND"** the LH's controller has provided the data subject with the information on the processing of his personal data'[4]] [action: collecting data].

2. [Role: LH's processor][Normative relation: **Power**] [condition: iff "processing of data is compatible with the purposes for which data was collected' AND 'controller took appropriate measures to provide information relating to processing to the data subject**' AND '** the LH's controller has provided the data subject with the information on the processing of his personal data"][action: process data].

5 PRELIMINARY IMPLEMENTATION

To implement the STDMP we used the Java Agent Development Framework based on BDI (Jadex) (Pokahr et al., 2005) platform. The Jadex, is an object-oriented software framework for the creation of goal-oriented agents following the BDI model. The Jadex reasoning engine tries to overcome traditional limitations of the BDI agents by introducing an explicit representation of goals and a systematic way for the integration of goal deliberation mechanisms. The Jadex agent framework is built on the top of the JADE platform and provides an execution environment and an application programming interface (API) to develop the BDI agents. In this paper, we propose to implement the STDMPs system using Jadex. As example of synthesis, we are now

[3]Pre-condition and Post-condition have been extracted from the GDPR.

[4]Providing information to the data subject can be done before the collection of data (then it is part of the pre-condition, and the providing of information was part of a different action), or during the action of collecting data (then the result is part of the post-condition)

able to implement the N-BDI* model illustrated in Algorithm 1. This is an excerpt of the code of the LH's controller agent where the agent checks the request and gives a permission (presented in the below code 5).

Code: A part of the LH's plan after receiving the order to process data

```
@Plan(trigger=@Trigger(goals=ExecuteTask.class))
....
    if(order!=null)
    {
        double time_span = order.getDeadline().getTime() -
              order.getStartTime();
        double elapsed_time = getTime() - order.getStartTime();
        String start = order.getCheckRequest(); //will check
              the request against GDPR
        if(start == "allowed")
        {
            // Save successful transaction data.
            order.setState(Order.DONE);
            order.setExecutionNorm(norm);
            order.setExecutionData(data);
            order.setExecutionDate(new Date(getTime()));

            String report = "Applied for: "+ Data;
            NegotiationReport nr = new NegotiationReport(order,
                  report, getTime());
            reports.add(nr);
        }
```

6 RELATED WORK

Various normative architectures have been presented by researchers for different purposes. One of the pioneering architectures in the area of normative multi-agent systems was the deliberative normative agents' architecture (Castelfranchi et al., 1999). According to this architecture, violating norms can be considered as acceptable as following them. Agents deliberate about the norms that are explicitly implemented in the model. Panagiotidi et. al presented a norm-oriented agent (Panagiotidi et al., 2012); this agent takes into consideration operationalized norms during the plan generation phase, using as guidelines for the agent's future action path. Boella and van der Torre (Boella and van der Torre, 2003) introduced a defender and controller agent in their normative multi-agent system. In their models, defenders agents should behave based on the current norms. Controllers monitor the behaviors of other agents and sanction violators, who can also change norms as needed. Garcia et al.(Criado et al., 2010) proposed a method to specify and explicitly manage the normative positions of agents (permissions, prohibitions and obligations), with which distinct deontic notions and their relationships can be captured. Another architecture that uses logical representation is presented by Sadri et al. (Sadri et al., 2006). The logical model of agency known as the KGP model was extended in this work, to support agents with normative concepts, based on the roles an agent plays and the obligations and prohibitions that result from playing these roles.

The EMIL (Lotzmann et al., 2013)architecture is one of the most elaborate normative architectures described in the literature. This architecture defines two sets of components for each agent:

1. Epistemic, which is responsible for recognizing norms;
2. Pragmatic, which is responsible for guaranteeing that the institution creates some (usually normative) agent's behavior.

) Applying the EMIL architecture in real scenarios can be challenging due to the elaborate design of its' cognitive mechanisms. Many existing normative architectures are based on the BDI (belief, desire and intention) structure. BOID (Belief Obligation Intention Desire) architecture extends the classic BDI agent's architecture to include the notion of obligation. Burgemeestre et al. (Burgemeestre et al., 2010)propose a combined approach to identify objectives for an architecture for self-regulating agents.

7 CONCLUSION

The regulation of multi-agent systems in environments with no control mechanisms, is gaining much attention in the research community. Normative multi-agent systems address this issue by introducing incentives to cooperate (or discouraging deviation). In our case, we used Calculamus, a knowledge representation formalism based on Hohfeld's normative relations, to express the norms that govern real-world data-sharing actions, essential for contract monitoring purposes. .

The STDMPs society is a regulated environment which includes the expression and use of regulations of different sorts: from actual laws and regulations issued by governments, to policies and local regulations issued by managers, and to social norms that prevail in a given community of users. For these reasons, we consider the secure data sharing problem is a representative example of a societal problem where norms impact the autonomous agents involved. Hence our case study, which we also used for evaluating the performance of the N-BDI* agent architecture. The agents' behavior in our STDMP model are affected by different sorts of norms which are controlled by different mechanisms such as regimentation, enforcement and grievance and arbitration processes. We identify the main goals of the TEI agent as being twofold. First, it aims at supporting agent interaction as a coordination framework, making the

establishment of business agreements more efficient. Furthermore, it serves the purpose of providing a level of trust by offering an enforceable normative environment. Our research is focused on modeling normative reasoning in a completely distributed environment. In particular, we are interested in how norms affect the STDMPs,, which monitoring activities enable detection of (non-)compliance in networked societies of agents, and what enforcement activities would enhance compliance.
In order to support this, we are working on the implementation of a prototype of the N-BDI* architecture. Our aim is to empirically evaluate our proposed solution through the design and implementation of scenarios belonging to the STDMPs case study. In future work, we will describe some experiments concerning the flexibility and performance of the N-BDI* agent model compared to simple BDI agents, using the STDMPs case study.

ACKNOWLEDGEMENTS

This work is funded by the Dutch Science Foundation project SARNET (grant no: CYB-SEC.14.003/618.001.016). Special thanks go to our research partner KLM. The authors would also like to thank anonymous reviewers for their comments.

REFERENCES

Bicchieri, C. (2005). *The grammar of society: The nature and dynamics of social norms*. Cambridge University Press.

Boella, G. and van der Torre, L. (2003). Norm governed multiagent systems: The delegation of control to autonomous agents. In *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, pages 329–335. IEEE.

Burgemeestre, B., Hulstijn, J., and Tan, Y.-H. (2010). Towards an architecture for self-regulating agents: a case study in international trade. *Lecture Notes in Computer Science*, 6069:320–333.

Castelfranchi, C., Dignum, F., Jonker, C. M., and Treur, J. (1999). Deliberative normative agents: Principles and architecture. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 364–378. Springer.

Criado, N., Argente, E., and Botti, V. (2010). A bdi architecture for normative decision making. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1383–1384. International Foundation for Autonomous Agents and Multiagent Systems.

Deljoo, A., Gommans, L., van Engers, T., and de Laat, C. (2016). An agent-based framework for multi-domain service networks: Eduroam case study. In *The 8th International Conference on Agents and Artificial Intelligence (ICAART'16)*, pages 275–280.

Deljoo, A., Gommans, L., van Engers, T., and de Laat, C. (2017). What is going on: Utility-based plan selection in bdi agents. In *The AAAI-17 Workshop on Knowledge-Based Techniques for Problem Solving and Reasoning WS-17-12*.

Dignum, F. (1999). Autonomous agents with norms. *Artificial Intelligence and Law*, 7(1):69–79.

Dignum, V. and Dignum, F. (2002). Towards an agent-based infrastructure to support virtual organisations. In *Collaborative business ecosystems and virtual enterprises*, pages 363–370. Springer.

Doesburg, R. v. and Engers, T. v. (2016). Perspectives on the formal representation of the interpretation of norms. In *Legal Knowledge and Information Systems: JURIX 2016: The Twenty-Ninth Annual Conference*, volume 294, page 183. IOS Press.

Farrell, A. D., Sergot, M. J., Sallé, M., and Bartolini, C. (2005). Using the event calculus for tracking the normative state of contracts. *International Journal of Cooperative Information Systems*, 14(02n03):99–129.

Gouaich, A. (2003). Requirements for achieving software agents autonomy and defining their responsibility. In *International Workshop on Computational Autonomy*, pages 128–139. Springer.

Lotzmann, U., Möhring, M., and Troitzsch, K. G. (2013). Simulating the emergence of norms in different scenarios. *Artificial intelligence and law*, 21(1):109–138.

Luck, M., Mahmoud, S., Meneguzzi, F., Kollingbaum, M., Norman, T. J., Criado, N., and Fagundes, M. S. (2013). *Normative Agents*, pages 209–220. Springer Netherlands, Dordrecht.

Oren, N., Panagiotidi, S., Vázquez-Salceda, J., Modgil, S., Luck, M., and Miles, S. (2009). Towards a formalisation of electronic contracting environments. In *Coordination, organizations, institutions and norms in agent systems IV*, pages 156–171. Springer.

Panagiotidi, S., Vázquez-Salceda, J., and Dignum, F. (2012). Reasoning over norm compliance via planning. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 35–52. Springer.

Pokahr, A., Braubach, L., and Lamersdorf, W. (2005). Jadex: A bdi reasoning engine. *Multi-agent programming*, pages 149–174.

Sadri, F., Stathis, K., and Toni, F. (2006). Normative kgp agents. *Computational & Mathematical Organization Theory*, 12(2-3):101.