# Negative Norms Detection Technique in Open Normative Multi-agent Systems

Muhsen Hammoud[1], Alicia Y. C. Tang[2] and Azhana Ahmad[2]
[1]College of Graduate Studies, Universiti Tenaga Nasional, Selangor, Malaysia
[2]College of Information Technology, Universiti Tenaga Nasional, Selangor, Malaysia

Keywords: Social Norms, Normative Multi-agent Systems, Negative Norms, Norms Detection.

Abstract: Social norms main objective is to regulate autonomous agents' behaviour in an open normative multi-agent system. Norms in these societies are dynamically created and disappeared according to the society's needs. Consequently, norms effects on agents or on the environment are not observable at the moment of creation. Norms practicing consequences might be either positive, like increasing the educational level of a society by conducting social discussions. Or negative, like causing money loss in gambling. Or the norm might have neutral consequences. In this paper, we propose a technique to detect negative norms in an open normative multi-agent system. Our technique has two main stages: i) Observation and ii) Analysis. The observation stage relies on the overhearing approach of monitoring where the messages that are exchanged between agents are observable. All observations are then analysed in order to detect negative norms. Negativity of a norm is based on its effect on agents or on the environment. In this technique, we adopted ATN concept to represent norms. This technique is implemented using Java and JADE. Testing results of this technique shows that it works properly, and detects negative norms according to the defined negativity threshold.

## 1 INTRODUCTION

A remarkable growing interest in regulating and coordinating agents' behaviour using the concept of social norms has been witnessed in the recent years (Hammoud, Ahmad et al. 2014). Norms usage in multi-agent systems lead to achieve the overall objectives of creating such systems (Modgil, Faci et al. 2009). There are mainly two approaches to create open normative multi-agent systems which are: regimentation approach and enforcement approach. In the regimentation approach (Jones and Sergot 1993), norms totally constrain agents behaviours. This means that agents are not allowed to behave autonomously. Consequently, agent's autonomy is drastically curtailed. This approach make the multi-agent system less flexible, and only specified agents can join. Regimented systems are adopted by electronic institutions for example. In contrast of regimentation approach, the enforcement approach (Conte, Falcone et al. 1999; y López, Luck et al. 2006; Grossi 2007; Dastani, Grossi et al. 2009; Oren, Panagiotidi et al. 2009) allows agents to use their autonomy. In these systems, norms are created dynamically according to the system needs. Besides, allowing autonomous agents to join this type of systems raise the possibility of creating new norms without knowing the long run consequences of practicing the newly created norms. Some of the created norms might cause negative consequences. The negative consequences might not be critical at the norm creation time. In order to discover these negative norms, agents' actions should be monitored, and the consequences of practicing such norms should be monitored also.

In this paper, we present a negative norms detection technique in open normative multi-agent systems. This technique builds on the overhearing approaches to monitoring, as in (Kaminka, Pynadath et al. 2011). The overhearing approach assumes that the messages that are exchanged between agents are observed. Consequently, agents' behaviours are inferred. At the same time, agents' mental state is not available for inspection. There is another approach for monitoring, called intrusive approach (Jennings 1995; Tambe 1997; Mazouzi, Seghrouchni et al. 2002), which assumes, in contrast with the overhearing approach, that agents' mental state is available for inspection.



Hammoud, M., Tang, A. and Ahmad, A.
Negative Norms Detection Technique in Open Normative Multi-agent Systems.
DOI: 10.5220/0005654502410249
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 2*, pages 241-249
ISBN: 978-989-758-172-4

Our detection technique relies mainly on monitoring agent's behaviours and their effect on the environment. This is done by using norms monitoring system, which is adopted from (Modgil, Faci et al. 2009). The monitoring system consists of a set of trusted observers in which any norm activity is observed and reported to the authority agent. At the same time, this monitoring system is able to observe any type of effects in a society and report about it to the authority agent. The authority agent then analyse these reports and, consequently, decides whether a norm is negative or not.

In the proposed technique, observers agents are assumed to be trusted by the authority. Therefore, their observations are not suspected. Trusted observers can be any part of service facilities, like a cashier. A trusted observer has the ability to observe the messages that are exchanges between society's agents and at the same time monitor their actions and report to the authority agent.

In this paper, we represented norms using Augmented Transition Network approach (ATN) (Woods 1986). This approach allows monitoring norms practicing by representing a norm as a set of nodes and arcs. The nodes represent norms state, while the arcs represent the necessary conditions in order to move from one state to another. Whenever the set of conditions that are represented as arcs are satisfied, the norm state changes.

The features of ATN representation of norms are as follow:

1. Norms are represented generally as an abstract model.
2. Ability to represent complex behaviours as a set of conditions.
3. Norms are represented independently using ATN. Hence, a norms might be created and removed at runtime

Our contribution in this paper can be summarized as follow:

1. We propose a negative norms detection technique in open normative multi-agent system, which is based on agents' actions observation. This technique relies on a set of trusted observers and an authority agent.
2. ATN approach is used to represent individual norms in this technique. This approach provides the ability of monitoring complex behaviours and actions.
3. The previous two contributions provide a generic negative norms detection technique which can work in any open normative multi-agent system.

The next section dwells upon the related work on norms monitoring and agents' actions observation. Section 3 presents a description of the adopted norms representation in open normative multi-agent systems... Section 4 details out the proposed observation architecture along with the monitoring algorithm and ATN representation of norms. Section 5 presents the proposed technique of negative norms detection in details. Section 6 details out the implementation and testing of the proposed technique, and finally Section 7 concludes the paper.

## 2 RELATED WORK

In this section, we describe an architecture for monitoring agents behaviour in a normative multi-agent system (Modgil, Faci et al. 2009). The normative system consists of agents who practice the available norms. Norms practicing affects agents and the environment as well. This monitoring architecture suggests that there is a set of trusted observers who are able to observe all the messages that are passed between the society's agents. These messages are then analysed in order to detect norms violation and compliance. Norms are represented using Augmented Transition Network (ATN) approach (Woods 1986).

The monitoring architecture that is proposed in (Modgil, Faci et al. 2009) is dedicated to detect agents violation and compliance to norms. Therefore, the authority can take an appropriate action by either sanction or reward society's agents. Figure 1 illustrates the proposed monitoring architecture. The researcher adopts overhearing approach in monitoring in which agents as black boxes. Black boxes mean that agents' internal state transitions are invisible. When an agent takes an action that is a part of a norm, the observer agent maps an instant of the abstract representation of the practiced norm. As the observer agent receives more messages about agents' actions, the analysis process continues and new instances of different norms are created. The state of each norm changes according to the actions of the agent who is practicing it. Whenever a norm expires, this means that the agent who is practicing it fulfilled all the conditions or requirements of this norm, the observer agent reports the practicing result to the authority. Norms practicing result might be either a compliance or violation.

In this paper, we adopt part of this monitoring architecture and adapt it to be able to detect negative norms.

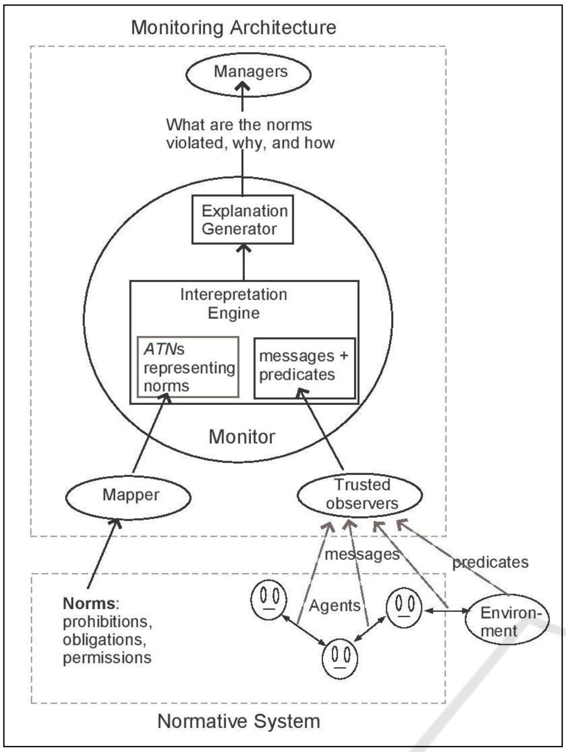


Figure 1: Agents' behaviour monitoring architecture.

# 3 NORMS REPRESENTATION

This section contains a review about general model of norms (Farrell, Sergot et al. 2005; Kollingbaum 2005; Oren, Panagiotidi et al. 2009). A norm *N* is modelled as a tuple: (NormType, NormActivation, NormCondition, NormExpiration, NormTarget, NormConsequences). A NormType might be one of three types: i) Obligation, ii) Permission and iii) Prohibition. A norm *N* is activated whenever the set of conditions that are described in NormActivation are satisfied. Whenever a norm *N* comes into force, or activated, an agent from observation group must track its state to detect the consequences of its practicing. The NormCondition set should be satisfied in order to say that a norm is practiced. The observer agent is the one who should be responsible of detecting and monitoring the satisfaction of NormCondition set. NormExpiration state refers to the state in which a norm is not in active any more. The described three components: i) NormActivation, ii) NormCondition and iii) NormExpiration, are called norm components.

NormTarget refers to the agents that are involved in practicing a specific norm *N*. For example: in a bank, the customer and the bank employee are the two agents who practice a norm of loan payment arrangement. Lastly, NormConsequences refers to the effect of practicing this norm on the society or agents, norm consequences are realized immediately when the norm expires. We illustrate the tuple of norm modelling in an example below.

Grocery pricing in a supermarket is an example for a norm that is practiced frequently in a society. Normally, customers need to price their grocery at a special place inside a supermarket before going to the cashier point. This norm is noticed in several countries like Malaysia. The norm state becomes active when a customer prices his grocery, and expires when the same customer pays his bill. In case the customer returned the grocery before reaching the cashier point, the norm is deactivated. The norm consequences are either positive in the case of completing the purchasing process, or negative in the case of returning the grocery. Negativity comes from the fact that the supermarket loses money if a customer returned his grocery.

# 4 OBSERVATION PROCESS

Agents' actions observation process represents the most important part of negative norms detection technique. Trusted observers are responsible of detecting norms practicing actions, and consequently report to the authority. In the following sections, we describe the observation process along with the monitoring architecture and ATN norms representation.

## 4.1 Description

Normally, agents' actions observation process is used to recognize their compliance and violation of norms (Modgil, Faci et al. 2009). Hence, agents' action is observed by a set of other agents, the observers, in order to recognise the cases of norm compliance or violation. Consequently, to apply a sanction in the case of violation, and give reward in the case of compliance.

In this paper, we adopt the norms compliance observation technique and adapt it to be able to detect norms consequences, as the main interest of this research is to detect negative norms which, in turn, relies on norms practicing consequences. Norms practicing monitoring requires detecting the fulfilment the conditions described in Norm-Activation, NormCondition and Norm-Expiration. We adopt the overhearing approach in monitoring in order to detect the fulfilment of these sets of conditions (Kaminka, Pynadath et al. 2011). This approach proposes that the messages that are exchanged between society's agents are observable, while the internal state of all agents is reserved.

An example that illustrates norms monitoring is grocery pricing in a supermarket that is described in section 3. This norm might not be applied in some countries, in Malaysia it is widely spread while in Brazil it doesn't exist. In this example, the supermarket, staff and equipments like casher computers are considered as trusted observers. This norm is activated whenever a customer wants to price the grocery he bought. The norm condition is fulfilled whenever the customer pays his final bill. This norm expires when the customer leaves the supermarket and never complains in the allowed complaint period, which might be 2 days for example. Since the pricing place is not at casher point, some customers might forget to price their grocery before coming to the casher point. Consequently, they either leave the grocery, or go back to price it. If the customer left the grocery, the supermarket doesn't get a benefit and one staff should return the grocery back to its place. It also might be damaged, therefore the supermarket loses. If the customer went back to pricing label place, customers behind him in the line are delayed, or he should take a new place in the line and he is delayed. The trusted observers, which are the supermarket staff and the equipments, gather the information about customers' actions and send it to specialized unit in order to analyse it, and therefore inform the authority agent.

## 4.2 The Monitoring Architecture

In this section, we describe the monitoring control loop which receives messages from the trusted observers and process them into ATN. Trusted observers send their messages to this monitoring architecture, all messages are stored in a message store for later processing. If a message contents satisfies specific arcs condition, then the respected norm's state is moved.

The monitoring algorithm is presented as follow:

***Require:*** *Message Queue* $Q_{Msg}$

***Require:*** *Message Store* $M_{St}$

***Require:*** *Set of Abstract Norms ATNs* $\mathcal{X}_{Abs}$

***Require:*** *Set of Instantiated Norms ATNs* $\mathcal{X}_{Inst}$

```
while Monitor is Active do
  while Q_Msg is not empty do
    Retrieve Msg from head of Q_Msg
    Add Msg to M_St
    for all A in X_Abs do
      for all Arcs α in A1_A do
        if satisfied(M_St, arc label α) then
          create norm ATN instance I of A
          add I to X_Inst
          move I to state S2
        end if
      end for
    end for
    for all I in X_Inst do
      for all Arcs α in A2_I do
        if satisfied(M_St, arc label α) then
          remove I from X_Inst
          move I to state S3
          notify authority about I consequences
        end if
      end for
    end for
  end while
end while
```

## 4.3 ATN Norms Representation

According to (Loritz 2013), ATNs are directed labelled graphs that were originally proposed for parsing complex natural languages. Basically, an ATN is composed of nodes that are connected with sets of arcs. Each arc has a label which should be processed in order to move from one node to another, the label contents might be a set of conditions or procedures. As mentioned in section 3, a norm has three components which are: Norm-Activation, NormCondition and NormExpiration. These components are represented as three nodes using ATN approach; these three nodes are connected with two sets of arcs. Based on that, a norm is represented using ATN approach as follow: *({S1, S2, S3}, A1, A2)*, where *S1, S2* and *S3* represents the components of a norm, while *A1* and *A2* represents the set of arcs that connects *S1→S2* and *S2→S3* respectively. Hence, norm activation corresponds to the fulfilment of the set *A1*, therefore changing the state of the norm from *S1* to *S2*. While norm expiration corresponds to the fulfilment of the set *A2*, therefore changing the state from *S2* to *S3*. The transition between *S1*, *S2* and *S3* happens based on the messages that are received from trusted observers.

The sets of arcs between the ATN nodes are labelled according the needed behaviour in order to move from one state to another, consider the example in section 3. For this purpose, we propose the following definitions:

Definition 1: Norm components mapping to ATN labels:

If $N_C$ is a norm component, then

$$N_C \in \{NormActivation, NormCondition, NormExpiration\} \quad (1)$$

Based on that:

$$N_C = \alpha_1 \vee \alpha_2 \vee \dots \vee \alpha_n$$
$$\alpha_i = \beta_1 \wedge \beta_2 \wedge \dots \wedge \beta_m \qquad for\ i = 1 \dots n$$

$$map(\beta_j) = (\{Ob_{\beta_j}, M_{\beta_j}, T_{\beta_j}\}) \quad for\ j = 1 \dots m \quad (2)$$

Where:

- $M_{\beta_j}$ represents a message
- $T_{\beta_j}$ represents the expression that is to be processed in conjunction with the processing of $M_{\beta_j}$
- $Ob_{\beta_j}$ represents the observer unique identifier.

Finally:

$$\forall \alpha_i: map\_conj(\alpha_i) = \bigcup_{j=1}^{m} map(\beta_j) \quad (3)$$

Definition 2: Norm representation into ATN approach:

If *N* is a norm, then:

*N={NormType, NormActivation, NormCondition, NormExpiration, NormTarget, NormConsequences}*, where:

- NormActivation = $\delta_1 \vee \delta_2 \vee \dots \vee \delta_m$
- NormCondition = $\epsilon_1 \vee \epsilon_2 \vee \dots \vee \epsilon_n$

*ATN* representation is a tuple *({S1, S2, S3}, A1, A2)* where:

- *A1* is a set of arcs $\{(S1,S2)_1, \dots, (S1,S2)_m\}$ $such\ that\ \forall i = 1 \dots m: map\_conj(\delta_i)$
- *A2* is a set of arcs $\{(S2,S3)_1, \dots, (S2,S3)_n\}$ $such\ that\ \forall i = 1 \dots n: map\_conj(\delta_i)$

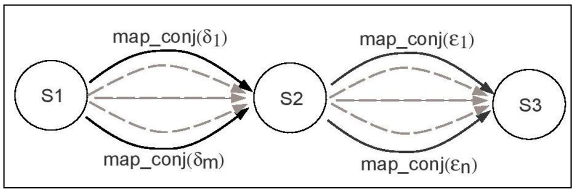


Figure 2: Augmented Transition Network.

# 5 THE DETECTION ARCHITECTURE

Negative norms detection model relies basically on monitoring control loop output. In the following sections we present the detection model along with its algorithm.

## 5.1 The Detection Model

The normative multi-agent system consists of two main parts: the society, and the authority. The society consists of three parts which are: society agents, society norms and the environment. These three components interact among each other. Society agents practice the available norms (1) and might change or create new norms according to their needs. When norms are practiced, they affect both agents and the environment (2). Agents are affected by norms either by gaining a benefit, being rewarded or sanctioned. While the environment is affected by the consequences of norms practicing, like increasing the pollution because of using more vehicles in transportation. The authority is normally represented in one agent with high level of power to control the society. This authority monitors the whole society through a set of trusted observers (3). In our model, those observers can be any services facility like banks, supermarkets, or any other facility. The trusted observers has the ability to capture all actions that are carried out by society agents, they also are able to detect norms practicing according to agents actions. Besides, trusted observers are able to detect any change in the environment and the cause of this change. These observers arrange their monitoring input into messages with special format, and then send these messages to the authority agent (4).

In our detection model, we care about the consequences of norms practicing only, so the first step of negative norms detection technique is to filter messages that arrived from trusted observers (5). These messages are stored in the cognitive structure of the authority agent. Whenever a new message is received from trusted observers, an analysis process is carried out in order to detect if a norm is causing negative effects or not (6). The analysis is done on the whole set of received messages that are related to a specific norm. After finishing analysis process, the result is stored in the cognitive structure again. Lastly, if a negative norm is detected, a report is generated about this norm (7). Then this report is sent to another unit in order to be handled properly (8).

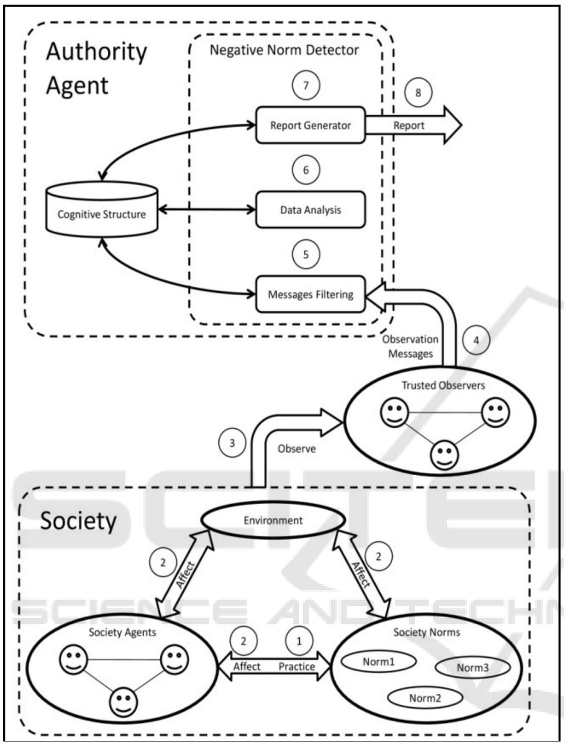


Figure 3: Negative Norms Detection Model.

## 5.2 The Detection Algorithm

The core idea of negative norms detection algorithm is to detect the norms that have negative consequences that exceed the allowed threshold for negativity. Norm negativity is calculated by dividing the number of negative consequences of this norm by the total number of practicing.

The detection algorithm is presented as follow:

***Define:*** *Norm Practicing Message Queue* $Q_N^P$

***Define:*** *Norm Effect Message Queue* $Q_N^E$

***Define:*** *Received Message* $M_R$

***Define:*** *Norm Practice Threshold* $N_{Th}^P$

***Define:*** *Norm Negativity Threshold* $N_{Th}^N$

***if*** *Message Received* ***then***
  $M_R$ ←*Received Message*
  ***if*** $M_R$ *is Norm Practicing Message* ***then***
    *Add* $M_R$ *to* $Q_N^P$
  ***else if*** $M_R$ *is Norm Effect Message* ***then***
    *Add* $M_R$ *to* $Q_N^E$
  ***end if***
  $C_N^P \leftarrow Count(Q_N^P)$
  ***if*** $C_N^P > N_{Th}^P$ ***then***
    $C_N^{NE} \leftarrow Count\_Negative(Q_N^E)$
    $N_L^N \leftarrow C_N^{NE}/C_N^P$
    ***If*** $N_L^N > N_{Th}^N$ ***then***
      *Mark N as Negative*
    ***end if***
  ***end if***
***end if***

## 6 IMPLEMENTATION AND TESTING

We implemented the proposed model using Java programming language in integration with JADE agent programming platform.

### 6.1 Java Agent Development Framework (JADE)



JADE is a middleware that is dedicated to develop distributed multi-agent applications which is based on peer-to-peer, or agent-to-agent where a peer in JADE is an agent, communication architecture. JADE provides the ability of distributing the intelligence, the information, the initiative, the resources and the control on either mobile or computer terminals in a fixed network. An environment that is created using JADE has the ability to evolve dynamically as agents can appear and disappear during run time according to the application requirements. Agents are able to communicate with other agents in the environment and at the same time make internal decisions.

JADE is a pure Java platform. It has the following principles:

- Interoperability: JADE is compliant with the FIPA specifications. As a consequence, JADE agents can interoperate with other agents, provided that they comply with the same standard.

- Uniformity and portability: JADE provides a homogeneous set of APIs that are independent from the underlying network and Java version.
- Easy to use: The complexity of the middleware is hidden behind a simple and intuitive set of APIs.
- Pay-as-you-go philosophy: Programmers do not need to use all the features provided by the middleware. Features that are not used do not require programmers to know anything about them, neither adds any computational overhead.

Figure 4 illustrates the architecture if JADE (Caire, Poggi et al. 2003).

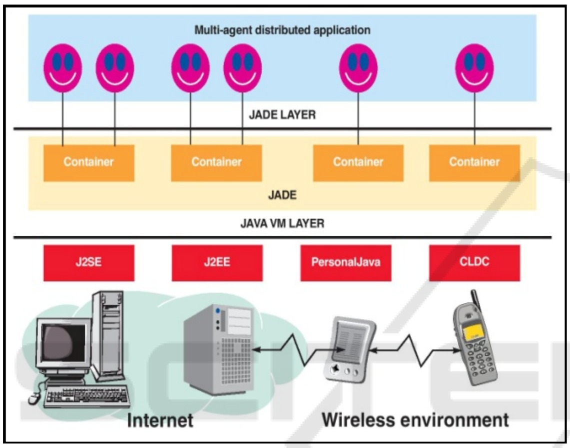


Figure 4: JADE architecture.

## 6.2 Implementation and Testing

The implemented simulation has two stages. The first stage requires entering the norms that are to be practiced, with some important information about them. Also the user should enter some information about the society, like the number of agents and observers. This interface is shown in Figure 5.

After that, JADE starts working. Figure 6 shows an experiment with 10 agents in a society and 3 norms and 4 observers. Society's agents exchange messages among each other; JADE platform handles the delivery process. Based on the exchanged messages, observers create instances of the practiced norms and track them. Figure 7 shows the interface of sniffer agent. Sniffer agent tracks the sent messages between all agents and shows them in a timeline depending on sending time.

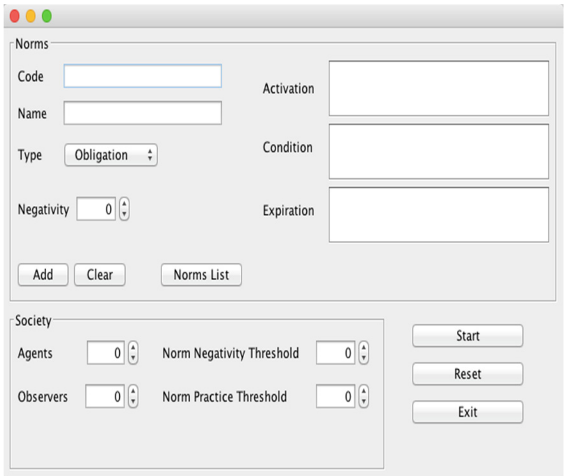


Figure 5: Simulation main interface.

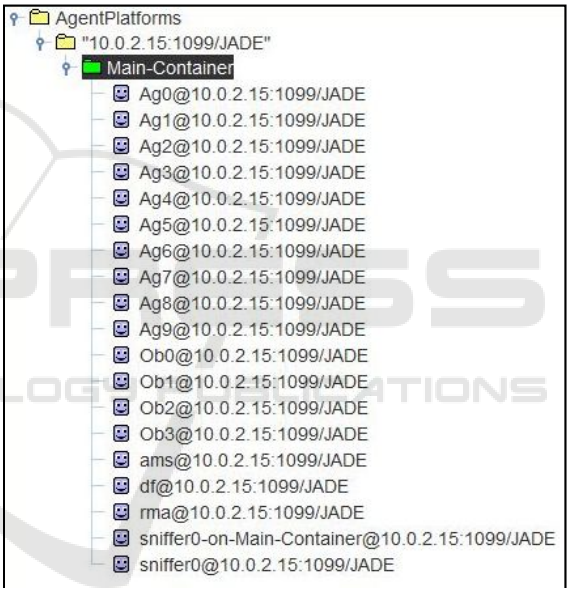


Figure 6: Experiment with 10 agents, 4 observers and 3 norms.

Figure 7: Sniffer agent interface.

The testing of this technique is done by analysing the messages that are exchanges between society's agents and, consequently, determine the negative norm among the available norms set. Testing results shows that the proposed technique is working properly. Negative norms are detected according to the determined negativity threshold. The smaller the threshold, the more norms are marked as negative. If the negativity threshold is set to the value 0, all society norms are marked as negative. On the contrast, if the negativity threshold is set to a high value, none of society's norms is marked as negative.

# 7 CONCLUSIONS

In this paper, we presented our negative norms detection technique in open normative multi-agent systems. This technique consists of two main parts: i) Agents' actions observation and ii) Analysis of observations. Agents' actions observation process is carried out by a set of trusted observers in which they are able to monitor all the messages that are exchanged between agents. They are also able to monitor and analyse agents' actions. We adopted the overhearing approach in which the internal mental state of agents is reserved, while the exchanged messages are observed. In this technique, norms are represented using ATN approach. This approach allows representing complex behaviours as a set of conditions and states. Besides, this representation is dynamic in which each norm has its own ATN abstract model. This feature allows the creation and removal of norms at run time.

This technique is a part of our work on formulating a theory of norms decay (Hammoud, Ahmad et al. 2014; Hammoud, Ahmad et al. 2014). The presented technique will be used in norms removal which is part of norms decay along with norms disappearance and norms collapse. The next step in this research work is to remove the negative norms in order to reach a stable society.

# ACKNOWLEDGEMENT

This work is supported by the Exploratory Research Grant Scheme (ERGS) by the Ministry of Education Malaysia under the Grant Ref. No. ERGS/1/2013/ICT01/UNITEN/02/02.

# REFERENCES


Caire, F. Bellifemine. G., A. Poggi, et al. (2003). JADE: A White Paper, Volume.

Conte, R., R. Falcone, et al. (1999). "Introduction: Agents and Norms: How to fill the gap?" Artificial Intelligence and Law 7(1): 1-15.

Dastani, M., D. Grossi, et al. (2009). Normative multi-agent programs and their logics. Knowledge Representation for Agents and Multi-Agent Systems, Springer: 16-31.

Farrell, A. D., M. J. Sergot, et al. (2005). "Using the event calculus for tracking the normative state of contracts." International Journal of Cooperative Information Systems 14(02n03): 99-129.

Grossi, D. (2007). Designing invisible handcuffs: Formal investigations in institutions and organizations for multi-agent systems.

Hammoud, M., A. Ahmad, et al. (2014). Norms Disappearance in Open Normative Multi-agent Communities: A Conceptual Framework. Agent and Multi-Agent Systems: Technologies and Applications, Springer: 233-243.

Hammoud, M., A. Ahmad, et al. (2014). Modeling norms removal in open normative multi-agent system. Computational Science and Technology (ICCST), 2014 International Conference on, IEEE.

Jennings, N. R. (1995). "Controlling cooperative problem solving in industrial multi-agent systems using joint intentions." Artificial intelligence 75(2): 195-240.

Jones, A. J. and M. Sergot (1993). "On the characterisation of law and computer systems: The normative systems perspective." Deontic logic in computer science: normative system specification: 275-307.

Kaminka, G. A., D. V. Pynadath, et al. (2011). "Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach." arXiv preprint arXiv:1106.1816.

Kollingbaum, M. J. (2005). Norm-governed practical reasoning agents, University of Aberdeen.

Loritz, D. (2013). "Generalized transition network parsing for language study: The GPARS system for English, Russian, Japanese and Chinese." Calico Journal 10(1): 5-22.

Mazouzi, H., A. E. F. Seghrouchni, et al. (2002). Open protocol design for complex interactions in multi-agent systems. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2, ACM.

Modgil, S., N. Faci, et al. (2009). A framework for monitoring agent-based normative systems. Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems.

Oren, N., S. Panagiotidi, et al. (2009). Towards a formalisation of electronic contracting environments. Coordination, organizations, institutions and norms in agent systems IV, Springer: 156-171.

Tambe, M. (1997). "Towards flexible teamwork." Journal of artificial intelligence research: 83-124.

Woods, W. A. (1986). Transition network grammars for natural language analysis. Readings in Natural Language Processing, Morgan Kaufmann Publishers Inc.

y López, F. L., M. Luck, et al. (2006). "A normative framework for agent-based systems." Computational & Mathematical Organization Theory 12(2-3): 227-250.