Extending the FIPA Interoperability to Prevent Cooperative Banking Frauds

Mauricio Paletta[1] and Pilar Herrero[2]

[1] Departamento de Ciencia y Tecnología.
Universidad Nacional Experimental de Guayana.
Av. Atlántico. Ciudad Guayana. Venezuela.
[2] Facultad de Informática. Universidad Politécnica de Madrid.
Campus de Montegancedo
S/N. 28.660 Boadilla del Monte. Madrid. Spain.

Abstract. Electronic bank transactions are very common today. Services given by an Automatic Teller Machine (ATM), for example, are very popular and widely used by bank clients. Unfortunately, in the same way as the use of these devices is increasing, the proliferation of different frauds to try to violate these systems to steal user's money is also increasing. Sometimes, the modus operandi used by the delinquents depends on different factors, such as the country or the city where fraud is committed or, as in the case of ATMs, the model or location of these devices. Since the detection of these modus operandi is not easy and they could be different from a bank institution to another, having both an environment capable of following up the swindler agents learning processes and a way to prevent the cooperation between these agents to share the learned knowledge, would be very useful to discover different modus operandi before crimes are committed. In this paper, a framework designed to follow up the swindlers' agents learning process and to share the knowledge between the agents is presented. This framework is based on the FIPA (Foundation for Intelligent Physical Agents) specifications and it emphasizes on the swindler agents learning process to fulfil the human-like agent behaviour and a realistic interaction with the environment.

1 Motivation

Electronic banking frauds have attracted significant international attention, since individuals and organizations have lost billions of dollars worldwide. Electronic banking speeds up transactions and creates new "promising" services, altering banking operations, and dramatically expanding the reach of financial institutions. Services given by an Automatic Teller Machine (ATM), for example, are very popular and widely used by bank clients. In fact, this kind of transaction is leading the current payment system.

Given the inherent nature of electronic banking in eliminating paper documentation and traditional identity verification processes, a new dimensional amount of risky situations have arisen. The proliferation of different frauds to try to violate these systems to steal user's money is also increasing.

Bank institutions are continuously receiving claims from victims of this type of crimes and the only thing they can do is to inform their employees and clients about one particular modus operandi once it is discovered.

Detection of procedures to commit these crimes is not easy and once it is discovered, criminals find another more skilful way to proceed. Sometimes, the modus operandi involves different people and techniques making them more difficult to detect. These also could depend on different factors: the country or city where the fraud is committed, the model or location of an ATM, etc. This hinders bank institutions to inform their employees and clients about crimes on time before they can be committed.

In this sense, having an environment capable of following up the swindler agents learning processes and to prevent the cooperation among these agents, with the purpose to share learned knowledge, would be very useful for bank institutions, discovering, for example, how criminals improve their modus operandi day by day and how they "create" new techniques and schemes to avoid the bank systems devoted to detect this kind of frauds.

If each bank institution could have one of these specialized agents which have learned the modus operandi from previous committed frauds, and all these agents could communicate among each other, then institutions can share learned knowledge and have more chance to avoid crimes. This is explained in more detail in this paper.

FIPAL (a FIPA agent-based framework designed to follow up the swindlers' intelligent agents Learning process) is also presented in this paper. It is based on the Foundation for Intelligent Physical Agents (FIPA)[1] specifications and it emphasizes on the swindler agents learning process to fulfil the human-like agent behaviour and a realistic interaction with the environment. In order to accomplish these necessities, the FIPAL-KBEL (Knowledge Base Experience Language for FIPAL), has been created as a new XML language based approach. This paper also describes how the FIPAL-KBEL language allows a flexible representation of the knowledge as well as a more effective learning process.

2 Related Work

The specifications of the Foundation for Intelligent Physical Agents (FIPA) constitute an interoperability model covering all elements from the agent architecture to the application domains. In this model, the agent system interoperability is based on the use of a common Agent Communication Language (ACL) [5] and supported by an Abstract Architecture [6] which can be used to abstract the internal architecture of each agent.

The FIPA abstract architecture provides some mechanisms that could be used to enact the communication process among heterogeneous agent systems

[1] http://www.fipa.org/

to achieving interoperability in message representation and transportation. However, the technological support provided by FIPA alone is not sufficient to achieve interoperability. Agents also need to have the knowledge to contextualize their acts of communication within the multiagent environment in which they take place [15].

Extensions of the FIPA specifications for intelligent agents could be seen in some research works, such as in [15] where the authors proposed an approach to extend the FIPA interoperability model to deal with agent social issues, like social requirements on agent conversations and communication languages. Examples of FIPA compliance, on the other hand, can be seen in other works, such as in [7] where Lynden et al introduced LEAF, a software toolkit for developing learning multiagent systems, Pokahr et al [12] present Jadex, a software framework for the creation of goal-oriented agents based on the FIPA specifications and following the BDI (Belief-Desire-Intention) [14].

By the other side, a no less important factor is the one related to the intelligent learning process. Learning plays a fundamental role in many of human activities since experience, including both achievements and errors [2]. It seems that this is the fundamental property that allows humans to adjust themselves to the different changes in the environment [1], [2], [3].

However, none of the previous authors has focused on the main objective from the perspective of this paper: to design and develop an open and flexible framework based on the learning process to be applied to the electronic banking fraud.

In the next section, the way in which an intelligent agent has been structured will be presented, by means of an open and flexible architecture using the FIPA specifications and based on the Learning and environment interaction processes (FIPAL). FIPAL is the evolution of a previous architecture called IVAL (An Open and Flexible Architecture based on the IVA Learning Process) which can be reviewed in detail in [10] and [11].

3 A FIPA extension designed to follow up the swindlers' agents learning process

3.1 The FIPAL structure

According to the FIPA in [6], the existence of a FIPA Abstract Architecture does not prohibit the introduction of elements useful to make a good agent system; it merely sets out the minimum required elements. Based on this affirmation and considering the BDI fundamentals, the FIPAL architecture has been designed as it can be seen in Fig. 1. There are five service modules:

1. The Service Directory Service (SDS): its basic role is to provide a consistent means by which agents and services can discover services [6].
2. The Social Issue Service (SIS): it contains those services needed to achieve the interaction with the environment based on stimuli reception and acting (see Sect. 3.2 for more detail).

3. The Learning Service (TLS): it contains a knowledge repository and those services needed to achieve the knowledge reasoning (see Sect. 3.3 for more detail).
4. The Communication Service (TCS): it contains the aspects of message communication between agents, as the message structure, message representation and message transport (see Sect. 4 for more detail).
5. The Control of Services (COS): it synchronizes the execution of the remainder elements and has the algorithms to react, deliberate and control the plans of the agent. It represents the FIPAL agent core since all the interaction among the other four elements passes through it. Related plans are represented using a kind of Teleo-Reactive (T-R) sequence [9].

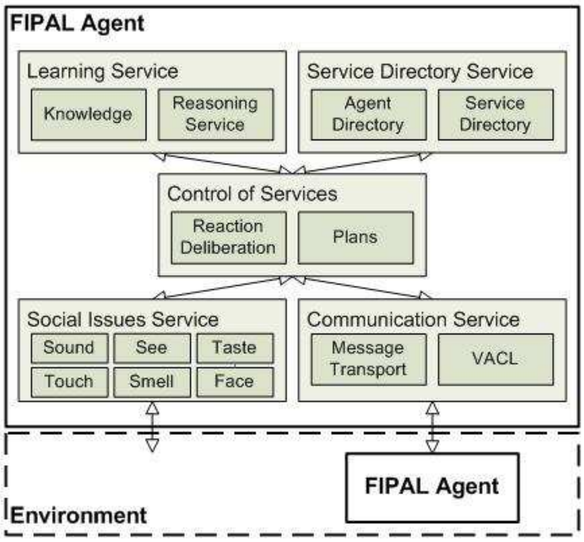

Fig. 1: The FIPAL structure

Following the FIPA specifications, the descriptions of all these services are registered in the Service Directory element inside the SDS module.

3.2 The FIPAL social issues

As it could be seen previously, the FIPAL structure is endowed with social issues abilities provided by the corresponding service module. One of the fundamental roles of this module is to interact with the environment. In this sense, a FIPAL agent has six components or set of services each of them with the capabilities to interchange information with the environment in order to receive a stimulus and give an answer according to the kind of interaction the agent is doing.

Five of the six kinds of interaction or social issues correspond to the human being senses. The other remaining social aspect corresponds with a service capable to represent facial expression. In this sense, a FIPAL agent tries to follow a social behaviour in the same way as the human beings do it. This six set of services are the following:

1. Sound, to interact with their surroundings by means of hearing (listening and speaking). The services associated to this component are the following:
   - Listening( ): start listening from the environment in order to receive auditory stimuli; with this service the FIPAL agent acquires the hearing ability.
   - StopListening( ): stop receiving auditory stimuli from the environment.
   - Talking(message): with this service the FIPAL agent can act saying an oral message to the environment.
2. See, to perceive the environment by means of the sight. The services associated are the following:
   - Seeing(): start seeing from the environment in order to receive visual stimuli; with this service the FIPAL agent acquires the visual ability.
   - StopSeeing(): stop receiving visual stimuli from the environment.
   - SeeFaceExpression(): with this service the FIPAL agent can act observing the facial expression of the most nearby agent.
3. Touch, to interact with their surroundings by means of the touch (touching and manipulating), with the following services:
   - Touching(): start touching objects into the environment in order to receive sense of touch stimuli; with this service the FIPAL agent acquires the sense of touch ability.
   - StopTouching(): stop receiving sense of touch stimuli from the environment.
   - Taking(): with this service the FIPAL agent can act taking the most nearby object.
   - Putting(object): with this service the FIPAL agent can act putting into the environment the object indicated.
4. Smell, to interact with their surroundings by means of the smell by using the following services:
   - Smelling(): start smelling from the environment in order to receive olfactory stimuli; with this service the FIPAL agent acquires the ability to smell.
   - StopSmelling(): stop receiving olfactory stimuli from the environment.
5. Taste, to interact with their surroundings by means of the taste by using the following services:
   - Tasting(): start tasting from the environment in order to receive taste stimuli; with this service the FIPAL agent acquires the taste ability.
   - StopTesting(): stop receiving taste stimuli from the environment.
6. Face, to represent and interpret the facial expressions such as happiness, fright, fears, etc.
   - Setting(expression): with this service the FIPAL agent can act by setting the indicated facial expression.

By the other side, FIPAL pays a lot of attention to the learning topic, including all an agent has to learn, how it should learn it and how knowledge should be handled. This is the main reason that has motivated the definition of a new XML-based approach for knowledge representation, known as FIPAL-KBEL. The following section describes how FIPAL-KBEL is used to represent the knowledge and how the learning process is carried out.

3.3 The knowledge representation and learning process in FIPAL

The FIPAL basic architecture is provided with learning abilities, thanks to the corresponding Learning Service (Fig. 1). One of the fundamental purposes of this service is to maintain the knowledge learned by the agent. In this sense, a knowledge representation technique was necessary to achieve this objective.

Due to its simplicity and flexibility, XML[2] (Extensible Markup Language) has been used as the basic representation language for covering the FIPAL functionalities, particularly knowledge representation. Since XML is a universal and web-based data format, it is appropriate for an independent platform and world wide use and has become a widely accepted standard data interchange technology, so its general usability is guaranteed for the next years [4].

Based on the previous statements, FIPAL-KBEL (Knowledge Base Experience Language for FIPAL) has been designed to fulfill the requirements related to knowledge representation. FIPAL-KBEL follows the rules representation [13] and, for instance, it defines the requested labels to structure a knowledge rule.

```
<?xml version="1.0"?>
<FIPAL-KBEL:Knowledge xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="FIPAL-KBEL.xsd"
 xmlns:FIPAL-KBEL="…">
 <FIPAL-KBEL:Experience id="…">
  <FIPAL-KBEL:Stimulus source="…">
   <FIPAL-KBEL:Data>…</FIPAL-KBEL:Data>
  </FIPAL-KBEL:Stimulus>
  <FIPAL-KBEL:Service name="…">
   <FIPAL-KBEL:Parameter name="…" type="…">…</FIPAL-KBEL:Parameter>
   <FIPAL-KBEL:Parameter name="…" type="…">…</FIPAL-KBEL:Parameter>
   … more parameters from this service
   <FIPAL-KBEL:Result type="…">…</FIPAL-KBEL:Result>
  </FIPAL-KBEL:Service>
  <FIPAL-KBEL:Service name="…">
   <FIPAL-KBEL:Parameter name="…" type="…">…</FIPAL-KBEL:Parameter>
   <FIPAL-KBEL:Parameter name="…" type="…">…</FIPAL-KBEL:Parameter>
   … more parameters from this service
   <FIPAL-KBEL:Result type="…">…</FIPAL-KBEL:Result>
  </FIPAL-KBEL:Service>
  … more services from this experience
 </FIPAL-KBEL:Experience>
 … more experience from this knowledge base
</FIPAL-KBEL:Knowledge>
```

Fig. 2: The FIPAL-KBEL document structure

Since the agent associated with the FIPAL architecture is acting according to the received stimuli, the relation between what it is saw, heard, touched, smelled or tasted and the given reactions needs to be represented and stored. Any stimulus consists on a tuple <data, type> that represents the information

[2] http://www.w3.org/XML/

coming from the environment. The reaction of the agent consists on many executed services with their corresponding parameters, if any; for example, in one service associated with the ability of talking, the parameter must be the string that the agent wants to say. Additionally, each service is accompanied by its corresponding executed result or reaction. In this way, each service is given by a pair $<service\_name(p_1,\ p_2,\ \cdots,\ p_n),\ result>$ where p_i $1 \leq i \leq n$, is the ith parameter.

Fig. 2 can be reviewed to detail the labels of the FIPAL-KBEL document structure. As it can be seen, FIPAL-KBEL represents each of this knowledge rules inside the label "<FIPAL-KBEL:Experience>". Any rule has a different identification number, "id" attribute, inside this label and it is used to indicate the corresponding sequence into the knowledge base. This is useful to follow exactly what happened before and after the stimulus arrived. It is important to highlight that the result of a service indicates whether not only the service was executed properly, but also if it was appropriate or still unknown.

The reasoning process occurs when a new stimulus arrives and the agent looks for an adequate answer; it consults the learning service component to know whether there is any experience with the present state or stimulus. If a knowledge rule associated with the received stimulus is found, all the services identified in the rule with the appropriate result value are executed in the same order they are executed when the same stimulus is received during the learning process. Fig. 3 shows in detail this algorithm executed by the Control of Services component.

```
Experience E
Stimulus I

For all Stimuli received:
  I = SocialIssuesService.StimulusReception()
  E = LearningService.Consulting(I)
    If E is not null
      For all E.Services: E.S_i
        ControlOfServices.Calling(E.S_i,p_1,…,p_n)
    For all new Services: S_i
      E = new Experience(I,S_i)
      LearningService.Learn(E)
```

Fig. 3: The FIPAL reasoning and learning general algorithm

If a different service is executed after the evaluation of the knowledge rule, this new information is considered to reinforce or give feedback to the knowledge according to what it was learned. This learning strategy continues until a new stimulus is received. This means that all the services executed after a stimulus is received and before a new stimulus arrives are taken in consideration to upgrade the knowledge rule associated with the first stimulus. To better understand the previous algorithm, Fig. 4 shows an UML collaborative diagram between the elements that conform the FIPAL structure (Fig. 6 in section 5 shows the UML class diagram related to these concepts).

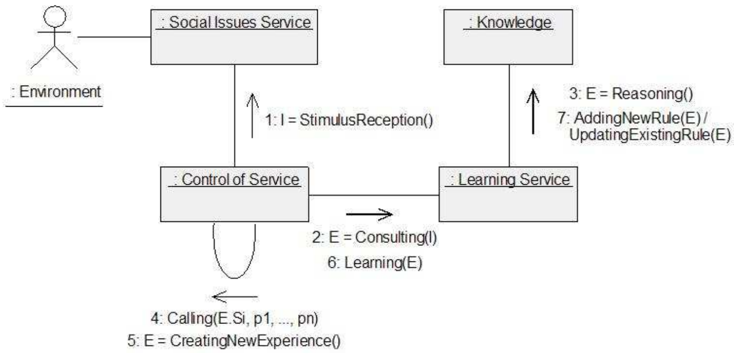

Fig. 4: UML collaboration diagram of the FIPAL learning and reasoning general process

Next section describes the way in which the FIPAL agents can interact each other to prevent banking frauds cooperatively.

4 The FIPAL interoperability to prevent cooperative banking frauds

Supposing there is at least one FIPAL agent in any bank institution which is learning the modus operandi used by the delinquents to commit the frauds in this institution, and supposing that these FIPAL agents are connected among each other (see Fig. 5), then the FIPAL agents can share the experience acquired with the aim to detect the possible fraud on time before it can be committed.

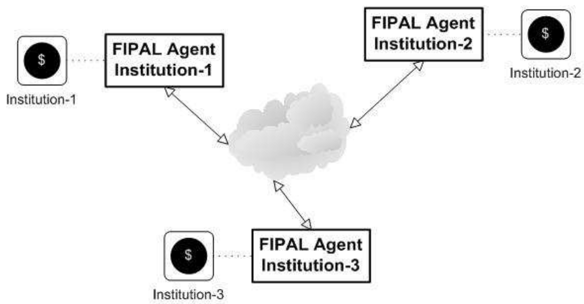

Fig. 5: Cooperative banking frauds

In order to cooperate with each other, any agent structured with FIPAL can respond to the service "Consulting(stimulus)" which is associated with the component Learning Service (see Fig. 1). The objective of this service is to find an experience based on the given stimulus and, as it occurs with the rest of services the agent can serve, the description of this service is registered in the Service Directory element of the FIPAL structure.

Once the Communication Service of the agent receives the request of executing the service "Consulting(stimulus)", the Learning Service element looks for an experience associated with the given stimulus by using the algorithm showed in Fig. 3. If there is a knowledge rule associated with that stimulus, the agent prepares an answer based on the services related to it. Only that services which have the corresponding results with a set value indicating if its execution is or not appropriate, will be considered for the answer. This is useful to inform the agent who sent the request of the service what it is or not appropriate to do in response to that stimulus.

A FIPAL agent can ask for help at anytime, particularly when it doesn't have any experience to an actual situation. Once the agent receives the answers from other agents, if any, it can execute the services with an appropriated value in their corresponding results and avoid executing services with an inappropriate value. All this information is used for this agent to learn how must act in presence of this stimulus in the future.

In order to achieve the communication among FIPAL agents, the FIPA ACL abstract message structure was adopted to define the XML based language FIPAL-VACL (Virtual Agent Communication Language for FIPAL). XML allows agent developers to extend sensor or effectors classes with their own ACL if they so desire, and the text basis of XML minimises functional coupling between sensors and effectors of different agents [8].

A scenario in which it is possible to review the way a couple of FIPAL agents can interact between each other to prevent banking frauds cooperatively is described in the next section.

5 A case study: Cooperative Banking Frauds

It should be pointed out that this work is currently in progress and therefore FIPAL will continue evolving. FIPAL is developing using C# programming language and, at this moment, efforts have been concentrated on having an appropriated graphical interface for the IVA. Fig 6 shows a UML class diagram with the concepts related to FIPAL and the relations among them. As it can be seen, these concepts and relations are related to the FIPAL structure showed in Fig. 1.

Several scenarios have been raised with the aim of validating the FIPAL interoperability described in the previous sections. The scenario presented in this section corresponds to a simulation of real situation which has been happening in Venezuela related to the Automatic Teller Machine (ATM) banking fraud.

The environment where the simulation is carried out is an ATM precinct with a couple of machines: ATM-1 and ATM-2. There are three agents: the delinquent D1, a second delinquent D2 and the client C1. The fraud is carried out because while the client C1 is initiating the transaction in ATM-2, the delinquent D1, who is using ATM-1, has programmed this ATM to intercept the transaction of ATM-2 acting itself as if it were ATM-2. Then, D1 prevents C1 from using ATM-2 pretending it does not work, while D2 is already finishing in ATM-1 the transaction that C1 initiated in ATM-2.

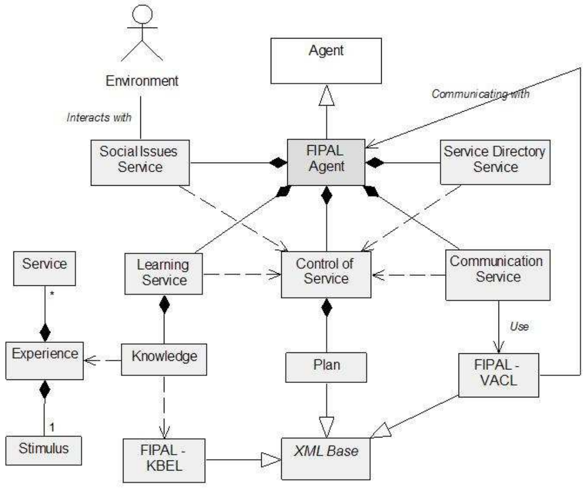

Fig. 6: UML class diagram of the FIPAL concepts

The scenario described previously can be summarized through the following steps:

1. C1 arrives to the ATM room while D1 is using the ATM-1 and the ATM-2 is free.
2. C1 is located in front of ATM-2. It introduces its card and its password while D2 enters in the ATM room.
3. D2 is in the ATM-1 queue, behind D1; D1 moves towards C1 with a serious facial gesture and tell him that the ATM he is intended to use (ATM-2) counts the money but it will not be able to get the cash from the ATM (see Fig. 7-a).
4. C1 reacts to D1's comment by cancelling the transaction which he was carrying on.
5. Once C1 has given the order of cancelling the transaction in the ATM-2 and it has waited for the confirmation of the cancellation, it decides to go to the ATM-1.
6. D1's transaction in ATM-1 is over and it moves towards the ATM-2; D2, which was waiting behind D1, starts using ATM-1 and C1 places behind D2 to use the ATM-1 as soon as possible.
7. D1, with a warning facial expression, tells C1 that the ATM-2 is making a transaction.

8. C1, with a worry expression, goes to the ATM-2 to cancel the transaction, while D1 gets out from the ATM room (see Fig. 7-b).
9. C1 can not observe any kind of transaction movement in the ATM-2 and therefore it decides to move again towards the ATM-1 where D2 is still finishing its transaction before leaving the ATM room (see Fig. 7-c).
10. Once C1 is located again in front of ATM-1, it introduces its card, password and the rest of the data to start a new transaction.
11. ATM-1 alerts to C1 that it has already taken out the maximum amount permitted by the bank; C1 reads this message and realises of the fraud (see Fig. 7-d).

With the aim of explaining better the simulation it is assumed that there are two bank institutions: B1 and B2. In B1 the FIPAL agent associated to the client C1 (for convenience, it will be used B1:C1 to identify this agent) has learned the modus operandi previously described, but in B2 the agent associated to C1 (B2:C1) doesn't have any experience yet. The simulation is going to be divided in two different parts: 1) the B1:C1's learning process and 2) the B2:C1's learning process using the B1:C1's experience. Both of parts are described by means of a series of steps each of them with a brief description of the simulated situation and the explanation of how the FIPAL architecture associated to C1 (B1:C1 or B2:C1) tackles this scene or step.

(a) Step 3

(b) Step 8

(c) Step 9

(d) Step 11

Fig. 7: Simulated situation of some steps of the scenario. The ATM-1 is the ATM on the left side being used by D1 in (a); the ATM-2 is the ATM on the right side being used by C1 in (a), (b) and (c); D2 is the woman

5.1 The B1:C1's learning process

Step 1: C1 arrives to the ATM room while another client (D1) is using the ATM-1 machine and the ATM-2 is free. **FIPAL:** The environment perception starts once the B1:C1 agent executes the service "Seeing()" from the See set of services. In response, B1:C1 receives a set of visual stimuli associated with the specific environment situation. In this case, the stimulus received would be <"ATM-1 busy", seen> and <"ATM-2 free", seen>.

Step 2: C1 is located in front of ATM-2. It introduces its card and its password while another client (D2) enters in the ATM room. **FIPAL:** Once received the stimuli and according to section 3.3, since B1:C1's knowledge about the fraud is initially null, B1:C1's reasoning about the fraud is null and therefore, it decides to ask for help by sending to the other FIPAL agents which are interconnecting into the cooperation banking network, a FIPAL-VACL message with the request of the service "Consulting(<"ATM-2 free", seen>)". Since there are not FIPAL agents with experience about this fraud, B1:C1 does not received any answer and therefore it didn't react to the delinquent intentions based on its experience; B1:C1 keeps its main purpose "to use the ATM machine as soon as possible" (based on its plans) and therefore it tries to go to the ATM-2 and to use this machine; B1:C1 executes the services "Putting(card)" and "Putting(password)" from the Touch set of services. In this moment, B1:C1 starts learning about what is happening, and as a consequence a knowledge rule is created based on the last stimulus received as well as on the actions executed; B1:C1 updates the knowledge with this new rule, modifying therefore the FIPAL-KBEL document.

Step 3: D2 is in the ATM-1 queue, behind D1; D1 moves towards C1 with a serious facial gesture and tell him that the ATM he is intended to use (ATM-2) counts the money but it will not be able to get the cash from the ATM. **FIPAL:** As D2 is in the ATM-1 queue, B1:C1 realises that the environment has changed and then it receives the stimulus: <"new client in ATM-1", seen>; B1:C1 doesn't react to this environment; moreover, the oral message that D1 sends to B1:C1 is captured by the Listen() service as an FIPAL-VACL message with the stimulus: <"ATM does not give money", listened>.

Step 4: C1 reacts to D1's comment by cancelling the transaction which he was carrying. **FIPAL:** Due to B1:C1 receives the stimulus <"ATM does not give money", listened> and it doesn't have experience on this respect, it decides to ask for help again following the same action as it is described in the Step2 and, in this case the sent FIPAL-VACL message contains this solicitude: "Consulting(<"ATM does not give money", listened>)". As it has no experience on this respect, B1:C1 reacts executing the service "Setting(worry)" from the Face set of services and C1's expression change to "worried". In the same way, B1:C1 reacts executing the service "Putting(cancel)" from the Touch set of services and the transaction is cancelled. B1:C1 updates its knowledge with this new rule, modifying therefore the IVAL-KBEL document.

Step 5: Once C1 has given the order of cancelling the transaction in the ATM-2 and it has waited for the confirmation of the cancellation, it decides to go to the ATM-1. **FIPAL:** The See module realises that the environment has changed and then it receives the stimulus: <"transaction cancelled", seen>; B1:C1 tries to makes a decision based on its experience but once more it has no experience on this respect and once more it doesn't receive any help from another FIPAL agent.

Step 6: D1's transaction in ATM-1 is over and it moves towards ATM-2; D2, which was waiting behind D1, starts using ATM-1 and C1 places behind D2 to use ATM-1 as soon as possible. **FIPAL:** B1:C1 realises the changes in the environment and it receives the corresponding stimulus without carrying out any reaction.

Step 7: D1, with a warning facial expression, says to C1 that the ATM-2 is making a transaction. **FIPAL:** The oral message that D1 sends to B1:C1 is captured by the Sound module with the stimulus: <"ATM-2 processing", listened>; B1:C1 tries to react with this new stimulus received.

Step 8: C1, with a worry expression, goes to the ATM-2 to cancel the transaction, while D1 gets out from the ATM room. **FIPAL:** B1:C1 reacts: 1) executing the "Setting(worry)" service from the Face module (with the aim of changing the face expression), 2) executing the "Putting(cancel)" service from the Touch set of services (with the aim of cancelling the transaction), 3) executing the "Seeing()" service from the See module (with the aim of observing the situation); B1:C1 introduces a new knowledge rule and updates its learning.

Step 9: C1 cannot observe any kind of transaction movement in ATM-2 and therefore it decides to move again towards ATM-1 where D2 is still finishing its transaction before leaving the ATM room. **FIPAL:** B1:C1 receives the stimulus <"Insert card", seen> from the See module and it reacts executing the service "Setting(normal)" from Face; the agent goes back to ATM-1; B1:C1 introduces a new knowledge rule and updates its learning.

Step 10: Once C1 is located again in front of ATM-1, it introduces its card, password and the rest of the data to start a new transaction. **FIPAL:** See receives the stimulus <"ATM-1 free", seen>; B1:C1 reacts executing the services: "Putting(card)", "Putting(password)" and "Putting(options)" from Touch; B1:C1 updates its learning with the new knowledge rule.

Step 11: ATM-1 alerts to C1 that it has already taken out the maximum amount permitted by the bank; C1 reads this message and realises of the fraud. **FIPAL:** See receives the stimulus <"it has withdrawn the maximum quantity of money permitted for day", seen>; B1:C1 reacts executing "Setting(worry)" from Face; B1:C1 introduces, again, a new knowledge rule and updates its learning. At this moment it is important to mention that, due to the fact that the final result for B1:C1 was negative, all the knowledge rules previous the last one and in which the services results are not setting yet, must be changed in order to set a non acceptable value into the corresponding services results.

5.2 The B2:C1's learning process using the B1:C1's experience

Step1: C1 arrives to the ATM room while D1 is using the ATM-1 and the ATM-2 is free. **FIPAL:** The environment perception starts once the B2:C1 agent executes the service "Seeing()" from the See set of services. In response, B2:C1 receives a set of visual stimuli associated with the specific environment situation. In this case, the stimulus received would be <"ATM-1 busy", seen> and <"ATM-2 free", seen>.

Step2: C1 (located in front of ATM-2) introduces its card and its password while D2 enters in the ATM room. **FIPAL:** Once received the stimuli and according to section 3.3, since B2:C1's knowledge about the fraud is initially null, B2:C1's reasoning about the fraud is null and therefore, it decides to ask for help by sending to the other FIPAL agents which are interconnecting into the cooperation banking network, a FIPAL-VACL message with the request of the service "Consulting(<"ATM-2 free", seen>)". Since B1:C1 has experience regarding to this stimulus, it responds to the solicitude sent by B2:C1 by sending a FIPAL-VACL message with the services "Putting(card)" and "Putting(password)", because this was what B1:C1 learned once it received this stimulus. In response to its solicitude, B2:C1 receives the message previously described and therefore it executes the services "Putting(card)" and "Putting(password)" from the Touch set of services. In this moment, B2:C1 starts learning about what is happening, and as a consequence a knowledge rule is created based on the last stimulus received as well as on the actions executed until now; B2:C1 uses the Learning Service component in order to update the knowledge with this new rule, modifying therefore the FIPAL-KBEL document.

Step3: D2 is in the ATM-1 queue, behind D1; D1 moves towards B2:C1 with a serious facial gesture and tell him that the ATM he is intended to use (ATM-2) counts the money but it will not be able to get the cash from the ATM. **FIPAL:** As D2 is in the ATM-1 queue, B2:C1 realises that the environment has changed and then it receives the stimulus: <"new client in ATM-1", seen>; B2:C1 doesn't react to this environment; moreover, the oral message that D1 sends to B2:C1 is captured by the "Listen()" service with the stimulus: <"ATM does not give money", listened>.

Step4: As B1:C1 has learned that cancelling the transaction due to a comment like D1 has done is not good, B2:C1 decides to continue with the transaction normally. **FIPAL:** Due to B2:C1 receives the stimulus <"ATM does not give money", listened> and it doesn't have experience on this respect, it decides to ask for help again following the same action as it is described in the Step2 and, in this case the sent FIPAL-VACL message contains this solicitude: "Consulting(<"ATM does not give money", listened>)". At this point it is important to mention that once B1:C1 found the rule that correspond with the stimulus <"ATM does not give money", listened>, it realises that this experience is associated with the services "Setting(worry)", "Putting(cancel)" and "Putting(options)" but, the results of the two first services are not acceptable. In this sense, B1:C1 answers with a message which only contains the execution of the service "Putting(options)" with the intention to inform to B2:C1 to continue with the transaction. Once B2:C1 receives the answer it reacts executing the service "Putting(options)" following the indications given by B1:C1. At this point, B2:C1 updates the knowledge with this new knowledge rule, modifying therefore the FIPAL-KBEL document.

Step5: The C1's transaction was successful and it receives the money from the ATM-2, so it takes its money and gets out from the ATM room. **FIPAL:** B2:C1 receives the stimulus: <"Taking money", seen>; B2:C1 reacts: 1) executing the service "Setting(happy)" from Face, and 2) executing the service "Taking(money)" from Touch; B2:C1 introduces a new knowledge rule and updates its learning. Due to the fact that the final result for C1 was positive, all the knowledge rules previous the last one and in which the services results are not setting yet, must be changed in order to set an acceptable value into the corresponding services results.

6 Discussion

In this paper it is presented FIPAL, a framework designing to follow up the swindlers' agents learning process. This framework is based on an open and flexible architecture, according to the FIPA specifications, that emphasizes on the swindlers' agents learning and environment interaction processes.

FIPAL is based on the FIPA architecture as well as on some previous research made on this direction, such as the design an implementation of a learning architecture for IVA named IVAL [10][11]. Both IVAL and FIPAL emphasize on the swindlers' agents learning process to fulfill not only more human-like agent behavior but also a more realistic interaction with the environment.

The implementation of the FIPAL architecture, complemented with the FIPAL-KBEL language, has already been evaluated in different simulations with successful results. Based on the evaluation of the model as well as on the results obtained, we can conclude that FIPAL, starting with an empty knowledge base, learns from the experience of interacting with the environment. This learning process is quite similar to human learning process since they are born.

In this paper we have concentrated in describing the way in which FIPAL agents can communicate or interact among each other to prevent banking frauds cooperatively. A complete scenario of a real ATM fraud situation was presented in this paper, in which was possible to verify that an agent, structured with FIPAL, not only is capable of learning the modus operandi used by the delinquents to commit the fraud but also cooperate with other agents which have not learned yet this particular modus operandi and therefore, both bankers and clients can be trained to react to this kind of banking frauds in almost real-time.

It is important to point out that FIPAL has not been implemented yet as final applications but it is part of the future work related to this research.

References

1. Bonasso R. P., Kortenkamp D.: An Intelligent Agent Architecture In Which to Pursue Robot Learning. In Proc. Workshop on Robot Learning, Rutgers University, New Brunswick, NJ (1994)
2. Buczak A. L., Cooper D. G., Hofmann M. O.: Evolutionary agent learning. International Journal of General Systems, Vol. 35 N 2 (2006) 231–254

3. Caicedo A., Thalmann D.: Virtual Humanoids: Let Them Be Autonomous without Losing Control. In Proc. Fourth International Conference on Computer Graphics and Artificial Intelligence, Limoges, France (2000)
4. de Vries A: XML framework for concept description and knowledge representation. Artificial Intelligence; Logic in Computer Science; ACM-class: I.7.2; E.2; H.1.1; G.2.3; arXiv:cs.AI/0404030 v1 (2004)
5. Foundation for Intelligent Physical Agents: FIPA ACL Message Structure Specification. SC00061, Geneva, Switzerland (2002) http://www.fipa.org/specs/fipa00061/index.html
6. Foundation for Intelligent Physical Agents: FIPA Abstract Architecture Specification. SC00001, Geneva, Switzerland (2002) http://www.fipa.org/specs/fipa00001/index.html
7. Lynden S., Rana O.: LEAF: A FIPA Compliant Software Toolkit for Learning based MAS. In Proc. First International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'02, Bologna, Italy, ACM 1-58113-480-0/02/0007 (2002)
8. Maher M. L., Smith G. J., Gero J. S.: Design Agents in 3D Virtual Worlds. In Proc. Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions (IJCAI) (2003) 92–100
9. Nilsson, N. J.: Teleo-Reactive Programs for Agent Control. Journal of Artificial Intelligence Research, 1 (1994) 139–158
10. Paletta M., Herrero P.: Learning from an Active Participation in the Battlefield: A New Web Service Human-based Approach. In Proc. International Workshop on Agents, Web Services and Ontologies Merging (AWeSOMe'06) in the Second OTM Federated Conferences and Workshops (OTM 2006), Montpellier, France. LNCS 4277, Springer (2006) 68–77
11. Paletta M., Herrero P.: Banking Frauds: An Agents-Based Training Framework to Follow-up the Swindlers Learning Process. Special Issue of International Transactions on Systems Science and Applications, Vol. 3, No. 2 (2007) (to be published)
12. Pokahr A., Braubach L., Lamersdorf W.: Jadex: A BDI Reasoning Engine. Multiagent Systems, Artificial Societies, and Simulated, Organizations International Book Series, Vol. 15, 10.1007/b137449, ISBN: 978-0-387-24568-3, Springer (2005) 149–174
13. Post E.: Formal reductions of the general Combinatorial Problems. American Journal of Mathematics 65 (1943) 197–268
14. Rao A. S., Georgeff M. P.: Modeling Rational Agents within a BDI-Architecture. In Proc. Second International Conference on Principles of Knowledge Representation and Reasoning, San Mateo, CA, USA, Morgan Kaufmann publishers Inc. (1991) 473–484
15. Soriano F. J., Reyes J., Gómez G., Amo F.: Extending the FIPA Interoperability Model to Deal with Agent Social Issues. In Proc. 6th. World Multiconference on Systemics, Cybernetics and Informatics, Orlando, USA (2002)