Non task oriented agents in dialogue

Sansonnet Jean-Paul
LIMSI
91403 Orsay Cedex
FRANCE
jps@limsi.fr

Valencia Erika
EPO
Patentlaan, Rijswijk
Nederland
evalencia@epo.org

ABSTRACT

We propose a model for dialogue between non task oriented agents, based on the dissonance theory. Non task oriented agents are studied as a model for non expert agents, as opposed to task oriented agents, in order to provide models for social science simulations. Dialogue between non task oriented agents can not be modelled like task oriented dialogue because not ask is provided to define the beginning and the termination of a dialogue, with respect to a common goal. The dissonance theory has been proposed by cognitive researchers as a *drive* for acting. Therefore, dissonance is a seducing theory to model the *motivation* of an agent to open a dialogue. Pertinence is also introduced to model the development and termination phases of such a dialogue. An implemented system, called OPDS, in then presented and evaluated.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.1 [**Information Systems Models and Principles**]: Miscellaneous

General Terms

Algorithm, Experimentation

Keywords

Information-based Agents, Dissonance Theory, Pertinence, Trust, Confidence, Coherence, Historic of a Dialogue: Path and Channel.

1. INTRODUCTION

While task-oriented agents interact with other agents in view of the completion of their task [13], what drives non-task-oriented agents to open a dialogue? This question is of primary interest for social informatics [3] because multi-agent based simulation for social issues is an increasing application domain of multi-agents research. Social issues deal with what we call *ordinary people*, that is people that have no special expertise, and cannot be modelled by task oriented agents (TO agents). Therefore, ordinary people are rather modelled by what we call *non-task-oriented* (NTO) agents (contrary to wooldridge, cite). In this paper we are concerned with a model for dialogue between NTO agents. We consider these NTO agents from the informational point of view, and model them as knowledge bases exchanging information in a dialogue.

Dialogue is divided into three main phases that are the opening [13], the development and the termination. The first question when concerned with NTO agents is to give a model for the motivation [11] of a NTO agent to open a dialog. Then, we will explore the processes dictating the pursue of this dialogue as well as its termination.

Concerning the opening phase of a dialogue, we propose a model for the motivation based on cognitive dissonance. Cognitive dissonance is a theory proposed by Festinger [5, 6] stating that dissonance is a painful state for a human agent that leads him to act in a way that will reduce the dissonance in order to reach the maximum possible of consonance. This dissonant state has therefore the status of a *drive*, we will call it the *dissonance drive*. We postulate that if the dissonance state of an agent has informational origins, then the dissonant drive will lead him to act in the field of informational interaction in order to reach again a maximum level of consonance. Acting in the field of informational interaction is precisely realized through the opening of a dialogue. The mechanism of development of the dialogue between NTO agents is also based on an informational concept: the pertinence.

In the following sections, we analyze what would be the dissonance theory applied to NTO agents. A in the next section, we propose the OPDM model for dialogue between NTO agents based on dissonance and pertinence. In the next section, we present the OPDS system that is the implementation of the OPDM model in the Mathematica [15] language, in particular we present the agent language description, the conversation policies and the main algorithms used for simulations. We give simulation outputs examples, and we evaluate our results, before concluding on this first version of the OPDM model.

2. DISSONANCE AND PERTINENCE FOR NTO DIALOGUE

We explore here the hypothesis that a model for dialogue between NTO agents can be based on the dissonance theory and the pertinence, in particular we propose that:

- The opening phase of a dialogue between NTO agents can be based on dissonance dissonance theory;
- The development phase is regulated by pertinence.

We also give a criterion for the termination of a dialogue.

2.1 Phases of a dialogue

Classically, in task oriented agents systems, dialogue is articulated into three main phases [13]: its opening, its development, and its termination. For task oriented agents, the opening of a dialogue is dictated by the need to *complete a task*, and this need exists as soon as a task is defined [1, 2]. The termination of a dialogue is then automatically reached when the given task is achieved. In other words, for task-oriented agents, the beginning and the termination of a dialogue is measured as a function of an external element: the task. For these reasons, the phase of a dialogue that has been most studied and modelled in the task-oriented multi-agents community is the development of the dialogue.

2.2 NTO agents

For NTO agents the question of the opening and of the termination can not be so solved with an external measure element, since no **common** task is provided (biswas aamas 2002). If we consider that NTO agents have a goal to drive their behavior, this goal can only be **private**. In fact, NTO agents do not form a CSCW-like systems but rather **collectives**, in the sense that they mostly co-act rather than cooperate (cite), they can even sometimes have conflicting goals within a perfectly functioning collective. We are therefore interested in a private drive model to open a dialogue.
As already said, NTO agents are considered here from the informational point of view, and modelled as information based agents: they contain a knowledge base and exchange information in a dialogue.

2.3 The Dissonance theory

The dissonance theory is based on the hypothesis that humans have a specific *need* to fulfill, called *cognitive consonance* [10]. We say that there is *consonance* when a fact f_a is a consequence of a fact f_b. The notion of consequence here is not necessarily the first order logics usual "consequence", but rather a "psychological measure". Indeed, dissonant theory comes from cognitive science and has been developed to model human behavior, in particular in learning tasks.
In an attempt to apply the dissonance theory to information-based NTO agents, we will state that there is a consonance if a fact f_a is a *logical* consequence of a fact f_b, precisely in the sense of the first order logics.
Coming back to the original theory, *dissonance* is then defined as a state where the desirable consonance is lost, in other words that a fact f_a is *not* a consequence of a fact f_b anymore. From the definition itself of dissonance, it is implied that if an agent was in a consonant *state* and then reaches a dissonant *state*, that means that a *transition* occurred. This transition is due to the *reception* of a *new* information fact from the agent.
This scheme is close to what happens in a dialogue where agents send and receive pieces of information. That is why we take the dissonance theory as a model to explain, *how*, an agent can be brought into a dissonant state, and how he may have the *need* to go back to a consonant state. This *need* will the *drive* that leads him to *act*.
According to which acts are available to an information-based agent, there are three categories of solutions for the agent to solve the dissonance, and go back to a consonant state:

- The first one of them, S_1, is to ignore the new information, that is the most recently received, and decide that it is not compatible with its own knowledge system : that means that the agent decides not to "believe" the new information.
- A second solution, S_2, is to declare that some piece of information he had before and that participates to the incoherence of his new knowledge system is wrong. This means that the agent decides that the new information has more confidence degree than his old beliefs.
- Finally, the third solution, S_3, to solve the incoherence is to suppose that the new piece of information and the old piece of information that the agent possessed before and that participates to the incoherence are only apparently in contradiction, and that some kind of further explanation would solve the puzzle and explain how those two could become compatible. That is then a good reason for an agent to open a dialog in quest for this hypothetical extra information.

We propose that applying the solution S_3 to get out from a dissonant state, is a motivation for a NTO agent to open a new dialog.

3. THE OPDM MODEL

In multi-agents systems, the agents are defined using an agent's definition formalism [16], the queries are written in an agent's communication language [8], and the dialogue is decomposed into exchange steps, according to conversation policies [14], [7].
In order to model the above hypothesis that a dissonant state drives a NTO agent to open a dialogue, we need a formal definition of an NTO agent, of a dissonant state for such an agent, and of the application of the solutions S_1, S_2 and S_3. We will also model the two other phases of a dialogue and give a formal definition of pertinence and of the termination criterion. Our model is called OPDM for Ordinary People in Dialogue Model.

3.1 Formal definition of a NTO agent

In a first step of modelling, we consider NTO agents as *information based* agents. By information based agents, we mean agents carrying a knowledge base (noted KB_i for an agent A_i) and acting with information oriented goals G_i. Information oriented goals are goals concerning the *obtention* of new information, but also *agreeing* or *exchanging* information, as well as achieving a coherence among the information of the knowledge base.

Each agent's knowledge base KB_i includes facts, noted f and rules, noted r. For the simplicity of the analysis, the formalism used to model the information in the knowledge base is a first order logic with a three values ($true, false, unknown$) valuating function. The agents are defined in a description language implemented over based on the mathematica programming language [15]. Each agent's base KB_i is defined according to the following structure:

- **Facts** are defined by: a *symbol* $f_{i,j}$ where i refers to the agent A_i and j is the numbering of the fact in A_i's list of facts; a *body* ${f_{i,j}}^{\beta}$ containing the fact's predicate in terms of first order logics; a *path* ${f_{i,j}}^{\pi}$ that is a *conversational chain* of agents through which the fact came to A_i; and also the *lists* of the agents that agreed ${f_{i,j}}^{\{agree\}}$, rejected ${f_{i,j}}^{\{reject\}}$ or knew ${f_{i,j}}^{\{know\}}$ already the fact $f_{i,j}$. The path ${f_{i,j}}^{\pi}$ and the lists ${f_{i,j}}^{\{\}}$ are build *sequentially* during conversations.
- **Rules**: they are implications in the sense of first order logics, and are noted $r_{i,j}$ where i refers to the agent A_i and j is the numbering of the rule in A_i's list of rules. Like the facts, they are also defined with a path and agree, reject and know lists.
- **Channels** are an *unidirectional* structure, noted $ch_{i \to b}$, where i stands for the agent A_i and b stands for its locutor A_b. A new channel is opened *each time* a dialog is opened with a new locutor [1] and is never closed. $ch_{i \to b}$ is composed of a *symbol* b, referring to the agent to which it is connected, the *list* of the facts already told to A_b, $ch_{i \to b}^{tolds}$, the *list* of the facts to tell to A_b, $ch_{i \to b}^{totell}$, selected among the facts that participate into a dissonance, and the integer cumulated counts of all the agrees $ch_{i \to b}^{agree}$, rejects $ch_{i \to b}^{reject}$, and knows $ch_{i \to b}^{know}$, obtained during the dialogues on this channel.

Therefore an agent is defined as $A_i = \{KB_i, G_i\}$ with:

$$KB_i = \{\{f_{i,n}\}, \{r_{i,m}\}, \{ch_{i,p}\}\}$$

where n, m, p are parameters to represent respectively the number of the fact, rule and channel, in ordered sets of facts, rules and channels . The pieces of *content* information exchanged by information-based agents can therefore be either facts f, or rules r, a new content information received by an agent is noted i_{new}, it can either be a fact f or a rule r.

The "functioning" of the agents is done with respect to their goals G_i. This functioning, the management of the coherence of the knowledge bases, as well as the communication capabilities of an agent are managed through external functions and are discussed in the next section.

3.2 Formal dissonant state

As already stated, we will say that an agent A_1 is in a dissonant state if there is a logical *incoherence* between its own base KB_1, and a new information received by him and noted i_{new}. The logical incoherence being defined as:

$$\{KB_1 \cup i_{new}\} \Longrightarrow \perp$$

Note that an agent can be brought to this a dissonant state by any previous experience that gives him a new piece of information Ii: for example a previous dialogue with an agent A_2, but also perceptive experiences. In a diachronically opened world, there is no actual beginning but only an interactional stream because the system is constantly evolving, so for all practical purposes, there are always previous experiences that can bring an agent into a dissonant state.
In order to track a dissonant state, each agent must check for the logical coherence of his new base $\{KB_1 \cup i_{new}\}$ each time a new piece of information i_{new} is received. This checking is done by an external function presented in the implementation section.

3.3 Solving dissonance: Confidence and Trust

To get out of its dissonance state, A_1 may apply one of the three solutions cited above: S_1, S_2 or S_3. To select which solution to apply, the agent calculates the degree of confidence he can put in each of the elements implied in the dissonance: The old pieces of information $f_{1,old}$ or $r_{1,old}$ of KB_1 that are not coherent with i_{new} and the new piece of information i_{new}.

and compares these values. For that purpose, we formally define these confidence measures in $[0, 1]$ in the next subsection. Once the measures are calculated, the agent A_1 then selects a solution according to:

- If the the confidence degree in i_{new} is less than the confidence degree in internal facts and/or rules participating to the dissonance, then the solution S_1 is chosen, and we say that i_{new} is rejected.
- If the confidence degree in i_{new} higher than the confidence degree in internal facts and/or rules participating to the dissonance, then the solution S_2 is chosen and we say that i_{new} is *integrated* to KB_1, while the internal facts and/rules participating in the dissonance are erased from KB_1.
- If the confidence degrees in i_{new} and in the internal facts and/or rules participating to the dissonance are "comparable" in value, then the solution S_3 is chosen. What we mean by comparable is of course relative to the norm chosen for the measures. It is an adjustable parameter of the implemented system, and we will give a numerical example in the next section.

Moreover, in the case where i_{new} did not come from a perceptive experience, the agent A_1 may also calculate the trust he has for the source agent A_2 that send i_{new} to him, and compare it with his own "self-confidence" (the trust he puts in himself as an agent).
These values are calculated according to the following definitions, for a given agent A_j

3.3.1 *Measure of confidence in an internal information of KB_i1*

The measure of confidence in an information of KB_1 is the confidence degree C_{1,f_j}, respectively C_{1,r_j}, in any fact $f_{1,j}$, resp. any rule $r_{1,j}$, of his knowledge base KB_1. This measure is a function of the *history* of the acquisition of the

[1] We will see that channels are used to record contextual information about the exchanges between agents. The viewpoint of an agent on such an exchange is called a channel.

information $f_{1,j}$, resp. $r_{1,j}$, by A_1: this history is stored in the path $f_{1,j}{}^{\pi}$, resp. $r_{1,j}{}^{\pi}$.

The path $f_{1,j}{}^{\pi}$ keeps track of the succession of agents through which $f_{1,j}{}^{\pi}$ was transmitted to A_1, each time, together with their own confidence in this information, for example:

$$f_{1,j}{}^{\pi} = \{(A_3, C_{3,f}), (A_5, C_{5,f}), (A_9, C_{9,f}), ..., (A_k, C_{k,f})\}$$

where A_3 is the last agent that send $f_{1,j}$ to A_1, and A_k was the first one of the chain. Then, the measure C_{1,f_j} is given by:

$$C_{1,f_j} = ((0.8 \times C_{3,f}) + (0.1 \times C_{5,f}) + (0.1 \times C_{9,f}))$$

which means that only the three last agents of the chain are taken into account, with a weighted sum giving much more weight to the last agent in the chain, here A_3. Of course, this weighted sum $\sum_{k=1}^{n}(w_k \times c_{k,f})$, could take into account more agents of the chain ($n > 3$) and the weights w_k could be decided according to a different repartition function.

3.3.2 *Measure of confidence in an outside information*

The **confidence** degree $C_{1,f_{new}}$ of A_1 in f_{new} does not really correspond to the actual confidence that the agent A_1 puts in the fact f_{new} *per se*, because $C_{1,f_{new}}$ is merely a preliminary *score* used by the agent to compare it to the confidence of an internal fact $C_{1,f}$ (or/and an internal rule $C_{i,r}$), in order to decide which solution S_1, S_2 or S_3 he should apply to get out from the dissonant state.

The score taken for $C_{i,f_{new}}$ is equal to A_2: $C_{2,f_{2,new}}$ if this fact is send by an agent A_2. Note that if the new fact f_{new} comes from a perceptive experience, we can either give an arbitrary value to $C_{i,f_{new}}$ or give a confidence degree depending on the perceptive *means* through which the fact is perceived by A_i, but this analysis is not within the subject of the current study.

For all practical purposes, the score will then always be the one of the source agent. In the case where the receiver agent A_1 decides to integrate this new information (case of solution S_2), then a confidence degree is given to f_{new}, which now becomes part of KB_1 and is renamed $f_{1,j+1}$. This confidence degree is calculated with the following formula:

$$C_{1,f_{new}} = C_{1,f_{j+1}} = (C_{2,f_{new}} \times t_{1\to 2})$$

where $t_{1\to 2}$ is the trust that A_1 has for A_2, this parameter is defined below.

3.3.3 *Measure of trust in another agent*

The **trust** degree $t_{1\to 2}$ representes the trust that the agent A_1 has for the agent A_2. Each agent A_i measures and updates a trust degree $t_{i\to j}$ associated with each **other** agent A_j. The case where $i = j$ corresponds to the self-trust defined below with a different formula.

The trust degree $t_{i\to j}$ depends on the interaction *history* between the agent A_i and A_j, in a unidirectional $1 \leftrightarrow 1$ correspondence, this information is stored in the channel $ch_{i\to j}$.

Indeed, the channel $ch_{i\to j}$ comprises the cumulated counts of all the *agrees* $ch_{i\to j}^{agree}$, *rejects* $ch_{i\to j}^{reject}$, and *knows* $ch_{i\to j}^{know}$, obtained during the dialogues between A_i and A_j from A_i's point of view. The trust degree $t_{1\to 2}$ is then given by:

$$t_{1\to 2} = \frac{((ch_{i\to j}^{agree} + ch_{i\to j}^{know}) - ch_{i\to j}^{reject})}{3}$$

This formula is the most simple one, but of course we could also consider different weights for each count, called w_{agree}, w_{know} and w_{reject} as long as the *rejects* are subtracted to the other two, and that the overall result is normalized by $N = w_{agree} + w_{know} + w_{reject}$.

3.3.4 *Measure of self-confidence*

The **self-trust** degree $t_{1:1}$ [2] corresponds to self-confidence that the agent A_1 has. In our model, $t_{1:1}$ depends on the interaction *history* between the agent A_1 and all the rest of the agents A_j ($j \neq 1$), in a $1 \leftrightarrow (n-1)$ correspondence.

The history of the interaction of A_1 with all the other agents is stored in all the channels $ch_{1\to j}$ of A_1. The general self-trust degree of an agent A_i is then given by:

$$t_{i:i} = \frac{1}{3 \times (n-1)}[$$

$$(\sum_{j=1, j\neq i}^{n} (ch_{i\to j}^{agree})) + (\sum_{j=1, j\neq i}^{n} (ch_{i\to j}^{know})) - (\sum_{j=1, j\neq i}^{n} (ch_{i\to j}^{know}))]$$

This formula takes into consideration the total amount of agrees and knows and rejects that A_i got from the $(n-1)$ other agents. Note that this value is different from the average of all the trusts $t_{i\to j}$ that A_i has for all the other agents A_j, $j \neq i$.

3.4 Pertinence and Focus

In the OPDM model, the opening of a dialogue between NTO agents is based on a model of dissonance and its development is based on the pertinence of the information exchanged. Indeed, since information based agents are not guided by a common task during the development of the dialogue, the main criterion that decides the pursue of a dialogue is also based on a *measure over the information* exchanged.

For that purpose, we define a measure of the pertinence p_f or p_r of a content information. The pertinence is modelled by the lexical intersection defined in the Worldnet project [4]. This pertinence is calculated with respect to a current focus of the dialogue.

The *focus* of a dialogue is dependent on the content information exchanged, and may be different for each agent because they have their own point of view on a dialogue. In a dialogue between A_i and A_j, these points of view are stored in the channel $ch_{i\to j}$ for A_i and in the channel $ch_{j\to i}$ for A_j, for example. For one agent A_i for example, the information

[2] the : instead of the $\to$ symbol is merely a convention for the expressivity of the symbol.

exchanged during a dialogue is stored in the lists $ch_{i\to j}^{tolds}$ and $ch_{i\to j}^{totell}$ of his channel. But these lists store the content information exchanged from the very first dialogue and that do not necessarily relate to the actual focus of the present dialogue. Only the three last facts or rules of each list are taken into consideration when defining a focus. Again, this number is chosen for practical purposes and could be different when appropriate testing would dictate so.

In the OPDM model, the focus of a dialogue between A_i and A_j, from the point of view of A_i, is given by:

$$Focus(i \leftrightarrow j, A_i) = \cup\{ch_{i\to j}^{tolds'}, ch_{i\to j}^{totell'}\}$$

where $tolds'$ stands for the last three elements of the list $tolds$, and $totell'$ stands for the last three elements of the list $totell$.

This focus is used by A_i to test the lexical pertinence of each new content information f_{new} send to him. The pertinence is then the boolean result from the intersection $\cap\{Focus(i \leftrightarrow j, A_i), f_{new}\}$.

3.5 Two termination criteria

In a model where dialogue is opened in the goal of solving a dissonance, the ending of at his dialogue is achieved when the said dissonance is solved. This state is detected by the same mechanism that tracked the dissonance. In fact the coherence of the knowledge base KB_i is tested each time a new information is added or an old information is erased from it.

But it is possible that a dissonant can never get out from a dissonant state through a single dialogue with another agent. In order to prevent them from cycling in a useless discussion, we have two mechanisms:

- Each agent checks if the same information has already been told to a particular other agent, this is stored in the list ${ch_{i\to j}}^{tolds'}$ of a channel;
- The pertinence of each new content information is also checked via the formula given above.

If, in a channel, there are no more items in the *totell* list, after a check in the *tolds* list, and that the new information content received from the other agent on this channel is not pertinent then the dialogue is terminated.

4. IMPLEMENTATION: OPDS SYSTEM

We defined the OPDM model consisting in formal definitions of NTO agents, dissonance, pertinence, trust, confidence and self-confidence. These notions are linked together in a model for dialogue between NTO agents. The opening of such a dialogue is conditioned by dissonant agents, its development is conditioned by the solving of this dissonance through the selection of an appropriate solution, this selection being performed according to trust and confidence measures, and the development and termination pertinence are also guided by the pertinence of the information exchanged. Before giving the algorithm that manages these different stages of a dialogue, we define the communication language used for the implementation of the OPDM model.

4.1 Communication language

An agent A_s, called the sender, sends a sentence σ_{sr} to an agent A_r, the receiver. The sentence σ_{sr} is called a sentence to differentiate it from a query, because σ_{sr} transports a piece content information. The sentences are written in a communication language, ACL, that is composed of three levels: The primitive dialog acts (defined according to speech acts [12] concerning information exchange, that is give or ask information and all the reactions to that), the conversation policies and the script language.

4.1.1 Primitive dialogue acts

We define the following primitive dialog acts **for the sender** A_s:

- `INFORM`: A_s gives a content information i. An answer can be returned to this act but not necessarily. The same information i is not proposed more than once by A_s to the same receiver A_r.
- `ASK`: A_s asks A_r for a content information i and waits until A_r sends it. During an ask, A_s does not send any content information to A_r.
- `END`: A_s has nothing more to say because there is no content information $i \in KB_s$ that is pertinent with respect to the focus of the dialogue and that has not been send via an INFORM to A_r.

And we define the following primitive dialog acts **for the receiver** A_r:

- `AGREE`: A_r agrees with the content information i received from A_s. Moreover, A_r did not have the information i in his knowledge base KB_r, and the information i is pertinent with respect to the focus of the dialogue. The information i is integrated in the base KB_r.
- `REJECT`: A_r rejects the content information i received from A_s, which means that it is the contrary of an AGREE. The information i was not present in the knowledge base KB_r of A_r, and this information is not integrated to KB_r. The reasons for i not to be integrated in KB_r are either that i is not pertinent with respect to the focus of the dialogue, or that it cannot be integrated because of the application of a type S_1 solution to a dissonant state.
- KNOWS: the content information i received by A_r from A_s is already present in KB_r.
- `FAIL`: A_r received an ASK from A_s concerning a content information i that A_s wants to get from A_r, but A_r has no content information j that verifies either $j = i$ or j pertinent with i. Therefore, A_r has nothing to answer to the ASK.

A last dialog act CHANGETOPIC is provided to change the focus of a dialogue. Indeed, the termination of a dialogue as defined above would be immediately reached with the first non pertinence. Since not all the knowledge in a knowledge base is connected through common symbols, it may happen that two agents have still knowledge to share with pertinent exchanges but do not "know" it.

4.1.2 Conversational Policies

The second level of the communication language consists in the communication policies (CP). Many policies may be defined using the CPSL. For example, we define as primitives of the conversation policies for our system the two following schemes:

- DUO: A_1 opens a dialogue with one agent A_2, the two agents can be in turn either the sender or the receiver with respect to the dialogue acts defined above. It is based on INFORM dialogue acts, it is a model for the social simulation of chats.
- RANDS: A_1 opens a dialogue with a random selection of an arbitrary number n of locutors, $A_2 ... A_n$.

And we also define the two following *modes* for the ASK dialogue act:

- ASKONE: A_s asks A_r for only one content information $i$matching his query q_{sr}.
- ASKALL: A_s asks A_r for all the possible content informations $i_1, ..., i_n$ matching his query q_{sr}.

With the CPSL it is possible to define conversational behaviors according to social models and to simulate them on our implemented multi agents system.

4.2 Scripting language

The OPDS system is an object-oriented program written with a scripting language that we defined over the mathematica programming language [15]. We define the following classes: AGENT, RELATION, FACT, INDIVIDUAL, CHANNEL.

Where INDIVIDUALS are the constants and variables of the first order logic predicates corresponding to the rules and facts. The other classes correspond to the concepts of agents, relations, facts and channels of the OPDM model.

Global lists keep record of the total existing agents, facts and rulers in the world of the simulation. The sender and receiver of a particular exchange are noted SENDER and RECEIVER, while the focus in a dialogue is noted CURRENTFACT.

Basic definition and visualization functions are also defined for agents: DEFAGENT and SEE. A visualization function is also provided for a complete simulation, it can show different detail levels relevant for analyzes, with the function PRINTLEVEL.

Dialog acts defined above are implemented by the corresponding functions, and executed by EXECINFORM, EXECAGREE and so on. The code for EXECINFORM is given below as an example.

EXECINFORM: (A_r, i_{new}):
If $(i_{new} = f \in KB_r)$
Then A_r **sends a** KNOW(f), **and** $ch_{r \to s}^{know} = ch_{r \to s}^{know} + 1$;
Else
  If $(i_{new} \cap KB_r \neq \emptyset)$
  Then
    If (i_{new} is a fact)
    Then A_r **creates new fact in** KB_r:$f_{i,NMAX+1} = i_{new}$,
      and $NMAX = NMAX + 1$,
      and send a AGREE,
      and $ch_{r \to s}^{agree} = ch_{r \to s}^{agree} + 1$;
    Else A_r **creates new rule in** KB_r:$r_{i,MMAX+1} = i_{new}$,
      and $MMAX = MMAX + 1$,
      and send a AGREE,
      and $ch_{r \to s}^{agree} = ch_{r \to s}^{agree} + 1$;
  Else A_r **sends a** REJECT,
  and $ch_{r \to s}^{agree} = ch_{r \to s}^{agree} + 1$;

The selection by say A_r of an information content to send in a transaction is done with NEXTFACTCHOICE. The complete algorithm of this function is not given here because it is rather mong, but the basic idea is the following: The receiver maintains an internal list of the facts that he didn't learn (via a KNOW) from A_s and selects the pertinent ones (via lexical connexity with the focus) among them. Then, from this candidates list, he removes the ones already told (using the tolds of the channel with $a8s$). If no candidates are left, then the dialog is finished, A_r can either change the focus by using the CHANGETOPIC function, or terminate. This choice is done according to two parameters: a maximum value of exchanged transactions set by the user (via a global time counter), and if there are facts left in KB_r that have not been told (not regarding the pertinence this time).

Note that the symbols used for the facts $f_{i,n}$ and rules $r_{j,m}$ are universal in each simulation, in the sense that a same symbol designates a same content information for all the agents. That means that we are not concerned with semantical heterogeneity in a first step of our model, and that the pertinence, coherence, and presence of an information in a base are only *lexically* checked.

The management of one dialogue transaction between two agents, called DUO and defined as a communication policy, is done by the function DUO. In this function, first it is tested whether the two channels $ch_{s \to r}$ and $ch_{r \to s}$ already exist (previous dialogues), otherwise they are created and all their variables are initialized (counters). A fact is selected in KB_s with the function selection function of the type of NEXTFACTCHOICE. At the begining the only facts or rules that are in the totell list are those coming from dissonances. This is the actual *drive* of A_s to start the dialogue. Here again, the dialogue may continue as long as there is something pertinent in the totell lists of one of the agents, or as long as a maximum time counter, set by the user, is not reached.

4.3 Examples of simulation

To start a simulation we create a new world with the command CREATEWORLD. This initializes all the variables and lists. Agents are arbitrarily created with DEFAGENT depending on which simulation is whished. For example, the silmplest interaction can be run by using the ASKONE policy:

```
RUN[ASKONE[_r]];\\
 *******  ASKONE[A,B]  *******\\
A -> B\\
ACTLIST= {ASK[_r,ASKONE]\\
ASK _r\\
B knows nothing about it and makes it a goal.\\
B -> A\\
ACTLIST= {FAIL[_r]}\\
FAIL\\
_r\\
```

Where ACTLIST is an internal list that keeps record of all the facts or rules that are candidates to be told, according to the criteria od NEXTFACTCHOICE described above. In this example, A_B could not find anything pertinent with respect to r and produced a FAIL. However, in the process, he kept the information r as a *goal*, it means it had been added to his own goals list G_B with a track of the agent that asks for this information A_A. As soon as A_B will learn anything about r (pertinent) he will put this new information in his totell list for A_A. This way of augmenting one's goals is also a possible drive to enter in dialogue. It corresponds to the will of finding something pertinent.

Here is a more complex example running az whole conversation policy: a DUO.

```
 RUN[DUO[C,B]];
 *******  DUO[C,B]  *******
C -> B
 ACTLIST={INFORM[fC71]}
 INFORM  fC71 :  r[4]
 B  gets info required by  {A}.
 B rejects this fact.
B -> C ACTLIST={REJECT[fC71,?],\\
CHANGETOPIC[fC71],INFORM[fB98]}\\
 REJECT  fC71  REJECTS=  {B[?]}\\
REJECTCOUNT= 1 CHANGETOPIC.  \\
fC71 INFORM  fB98 :  q[3] \\
C  learns the new fact  fC72 = q[3]. \\
C -> B ACTLIST={AGREE[fB98],INFORM[fC70]} \\
AGREE  fB98  AGREES= {C} AGREECOUNT= 1\\
INFORM  fC70 :  r[3]  \\
B gets info required by {A}. \\
B learns fB99 = r[3].

------- LOOP stopped.
B  has nothing NEW to say.\\
B -> C \\
ACTLIST={AGREE[fC70],END[fC70]}\\
AGREE  fC70  AGREES=  {B} \\
AGREECOUNT=  1\\
END. \\
fC70
```

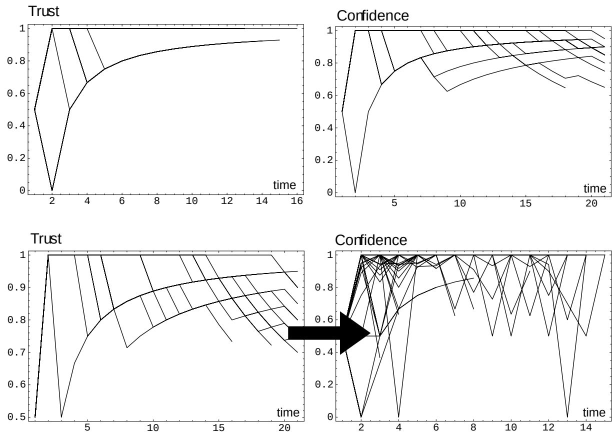

Figure 1: Trust and confidence: All the degrees get higher through the runtime because the world contains few connexity between the facts and rules and therefore, few incoherence.

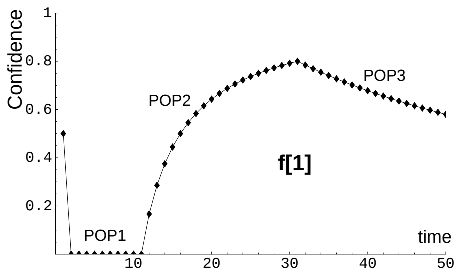

Figure 2: The confidence in a fact f[1] for three consecutive runs. In a first dialogue, its confidence was 0 because it was rejected. Then in the second dialogue, the confidence for this fact was growing, and in the third one, it decreased again as a consequence of new rejections from third interlocutor.

In this example, the print level shows each action of the agents: the INFORM, REJECT, AGREE, we can also follow the updates of the counters. After agent B had nothing new to say a last step shows its reaction to the fact $fC70$ received: an agree.

4.4 Evaluation

The evaluation of the OPDS system can not be an evaluation of the complexity of the algorithm because it was no designed in that sense and basically all the function have combinatorial complexity. In the case of small numbers of agents per simulation, this is not,; however, a constraint. The evaluation that is more interesting is to measure the confidence and trust degrees evolving through time when several dialogues are run. An agent having only new facts with respect to the others will have a high self-confidence after a few runs because nobody will contradict him (by dissonances detected). An agent having conflicting facts with others might get a lower self-confidence also. A plot of the evolution of trust and confidence degree is shown in figure 1 and figure 2 shows the evolution of the confidence degree in a fact f[1] during three consecutive runs.

Worlds with more or less intrinsic incoherence[3] are tested and the main consequence is to bring down the self-confidence of agents when the incoherence grows up. Semantical heterogeneity, if reintroduced, could change this linear dependence.

5. CONCLUSIONS

We proposed a model of dialogue between non task oriented agents based on dissonance theory and pertinence measure. The main hypothesis of this model is that a dialogue can be opened by an agent in the goal of solving an internal dissonant state, and that it may be developed by the agents as long as the exchange of information is pertinent with respect to the ficus of the conversation.

Pasquier et al's [9] already proposed that the dissonance theory could bring kinteresting highlights to the problem of structuring dialogue between agents. The idea that is could be a model for the opening of such a diagloue between non task oriented agents that no not interact as a function of an external task, is however not further modelled or implemented. It could be interesting to introduce the difference between an internal dissonances (within one KB) and external dissonances (between agents KBs), to follow one other interesting idea of their paper. This difference is however not clear from our model in its form because all the dissonance are coming from dialogue with other agents and non from perpections for the moment.

To model the notion of dissonant state, and the application of solutions to it, we introduced the confidence and trust degrees, that are meta-knowledge of the agents over the information they posses. This *meta-knowledge* is calculated using the history of the dialogues that is recorded in the paths and channels.

In the definitions we gave above, we see that the agents, the facts and the dialogue acts are defined as a function of the *information* they carry. The OPDM model may be considered as an information *market*, where facts are valuated by the agents with their value (true, false, unknown) and their confidence degree; where agents are valuated by others with the facts they know, agree or reject and their mutual and trust degree; and where *transactions* are evaluated by their pertinence degree.

Concerning the management of dissonant agents in a simulation, we can provide a stack M_i for each agent A_i to keep track of the dissonances D_{ij} that the agent A_i could not solve so far and that he will try to solve whenever possible. That means that if a new agent, A_3, comes into the system, if A_1 is aware of his presence, he may use this new source of information to solve the next dissonance on his stack M_i. In such a case, A_i will open a dialogue with A_3 in order to solve the last dissonance on the stack M_i. The stack M_i would be limited to a number n of dissonant events because, in practice, agents cannot keep a track of an arbitrary number of dissonances to solve. The agent A_i is said to "forget" the old dissonances that are not kept in M_i, that is why the dissonances stack is also called the memory. By this mechanism we would allow agents to terminate a dialogue without having solved all their dissonances, and in our simulations agents will be able to coexist in a system with unsolved dissonances.

[3]In the sense that the facts and rules in the knowledge bases of all the agents carry much incoherence.

6. REFERENCES

[1] A. Bond. *Proceedings of the 1990 Conference on Office Information Systems*, chapter Comitment: A computational Model for Organizations of Cooperating Intelligent Agents. Cambridge, 1990.

[2] S. Cammarata, D. M. Arthur, and R. Steeb. *Proceedings of the 8th International JOint Conference on Artificial Intelligence chapter = Strategies of Cooperation in Distributed Problem Solving, pages = 767-770, year = 1983, volume = 2, address = Germany.*

[3] P. Davidsson. *Journal of Artificial Societies and Social Simulation*, volume 5, chapter Agent Based Social Simulation: A Computer Science View. 2002.

[4] C. Fellbaum. *Breadth and Depth of the Lexicon.* 1999.

[5] L. Festinger. *A theory of cognitive dissonance.* Evantson, Peterson, iii edition, 1957.

[6] L. Festinger. *Conflict, decision, and dissonance.* Standford University Press, 1964.

[7] F. Flores, M. Graves, B. Hartfield, and T. Winograd. *ACM TOOIS*, volume 6, chapter Computer Systems and the Design of Organization in Interaction, pages 153–172. 1988.

[8] R. Fritzson, T. Finin, D. McKay, and R. McEntire. Kqml: A language and protocol for knowledge and information exchange. *13th International Distributed Artificial Intelligence Workshop*, 1994. July 28-30, Seattle, WA.

[9] P. Pasquier and B. Chaib-draa. Cohrence et conversations entre agents: vers un modle bas sur la consonance cognitive. *JFIADSMA'2002*, 28-30 Octobre, Lille, France 2002.

[10] J.-P. Poitou. *La dissonance cognitive.* Armand-Collin, collection u edition, 1974.

[11] J. Searle. *Intentionality: An essay in the philosophy of mind.* Cambridge University Press, New-York, 1983.

[12] J. R. Searle. *Speech Acts.* Cambridge University Press, herman edition, 1972.

[13] P. Torroni. A study on the termination of negociation dialogues. *Proceediongs of the first International Joint Conference on Autonomous Agents and Multi-Agents Systems*, pages 1223–1230, July 2002 AAMAS'2002. Bologna, Italy.

[14] T. Winograd. *Computer Supported Cooperative Work*, volume 1, chapter A language Action Perspective on the Design of Cooperative Work. Greif, 1988.

[15] S. Wolfram. *The Mathematica book.* Wolfram Media. Cambridge University Press, fourth edition, 1999.

[16] M. Wooldridge. *Reasoning about Rational Agents.* MIT Press, 2000.