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 then as knowledge bases exchanging information in a dialogue.

Dialogue is divided into three main phases that are the opening [12], 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 anmas 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 $f_k$ is a consequence of a fact $f_s$. The notion of consequence here is not necessarily the first order logic 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 $f_k$ is a logical consequence of a fact $f_s$, 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 $f_k$ is not a consequence of a fact $f_s$ 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 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 obtaining 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) 
valeuing 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 $f_{i,j}^b$ containing the fact's predi-
  cate in terms of first order logics; a path $f_{i,j}^p$ that is a 
  conversational chain of agents through which the fact came 
  to $A_i$; and also the sets of the agents that agreed 
  $f_{i,j}^=$, rejected $f_{i,j}^=$ or knew $f_{i,j}^=$ already 
  the fact $f_{i,j}$. The path $f_{i,j}^p$ 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 $ch_{i\rightarrow j}$ 
  , where $i$ stands for the agent $A_i$ and $b$ stands for its 
  locutor $A_b$. A new channel is opened each time a dia-
  log is opened with a new locutor 1 and is never closed. 
  $ch_{i\rightarrow j}$ is composed of a symbol $b$, referring to the agent 
  to which it is connected, the list of the facts already 
  told to $A_b$, $ch_{i\rightarrow b}^{old}$, the list of the facts to tell to $A_b$, 
  $ch_{b\rightarrow i}^{new}$, selected among the facts that participate into 
  a dissonance, and the integer cumulated counts of all the 
  agree $ch_{i\rightarrow j}^{new}$, reject $ch_{i\rightarrow j}^{old}$, and knows $ch_{i\rightarrow j}^{new}$, 
  obtained during the dialogues on this channel.

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

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

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 ex-
changed 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 $A_i$ is in a disso-
nant state if there is a logical incoherence between its own 
base $KB_i$, and a new information received by him and noted 
in$_{new}$. The logical incoherence being defined as:

$$\{KB_i \cup i_{new}\} \models \perp$$

Note that an agent can be brought to this a dissonant state

1We will see that channels are used to record contextual in-
formation about the exchanges between agents. The view-
point of an agent on such an exchange is called a channel.

by any previous experience that gives him a new piece of 
information $i$: for example a previous dialogue with an 
agent $A_2$, but also perceptive experiences. In a discrini-

cially opened world, there is no actual beginning but only an 
interactional stream because the system is constantly evolv-
ing, 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 $\{KB_i \cup i_{new}\}$ each time a new piece of information $i_{new}$ is received. This checking is 
done by an external function presented in the implementa-
tion section.

### 3.3 Solving dissonance: Confidence and Trust

To get out of its dissonance state, $A_i$ 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 $f_{i,old}$ or $r_{i,old}$ of 
$KB_i$ 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 subsec-
tion. Once the measures are calculated, the agent $A_i$ 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 par-
  ticipating to the dissonance, then the solution $S_1$ is 
  chosen, and we say that $i_{new}$ is rejected.

- If the confidence degree in $i_{new}$ is higher than the confi-
  dence degree in internal facts and/or rules participat-
  ing to the dissonance, then the solution $S_2$ is chosen, 
  while the internal facts and rules participating in the 
  dissonance are erased from $KB_i$.

- 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 per-
ceptive experience, the agent $A_i$ 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 defi-
nitions, for a given agent $A_i$.

#### 3.3.1 Measure of confidence in an internal information 
of KB$_i$

The measure of confidence in an information of $KB_i$ is the 
confidence degree $C_{i,j}$, respectively $C_{i,j}$, in any fact $f_{i,j}$, 
resp. any rule $r_{i,j}$, of his knowledge base $KB_i$. This mea-
sure is a function of the history of the acquisition of the
information $f_{1,j}$, resp. $r_{1,j}$, by $A_1$; this history is stored in the path $f_{1,j}^*$, resp. $r_{1,j}^*$.

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

$$f_{1,j}^* = \{(A_3, C_{a,j}), (A_5, C_{a,j}), (A_6, C_{a,j}), \ldots, (A_k, C_{a,j})\}$$

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

$$C_{1,j} = ((0.8 \times C_{a,j}) + (0.1 \times C_{a,j}) + (0.1 \times C_{a,j}))$$

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 $A_k$. Of course, this weighted sum $\sum_{k=1}^{n}(wk \times C_{a,j})$, 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 $C_{i, j_{new}}$ of $A_i$ 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 $C_{i, j_{new}}$ is merely a preliminary score used by the agent to compare it to the confidence of an internal fact $C_{1,j}$ (or/and an internal rule $C_{r_1}$), in order to decide which solution $S_1$, $S_2$ or $S_3$ he should apply to get out from the dissontant state.

The score taken for $C_{i, j_{new}}$ is equal to $A_2$: $C_{2, j_{new}}$, if this fact is send by an agent $A_2$. Note that if the new fact $f_{new}$ comes from a perception experiment, we can either give an arbitrary value to $C_{i, j_{new}}$ 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:

$$C_{i,j_{new}} = C_{i,j+1} = (C_{2,j_{new}} \times t_{i\rightarrow j})$$

where $t_{i\rightarrow j}$ 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 $t_{i\rightarrow j}$ represents the trust that the agent $A_1$ has for the agent $A_2$. Each agent $A_i$ measures and updates a trust degree $t_{i\rightarrow j}$ 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 $t_{i\rightarrow j}$ 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 $\chi_{i\rightarrow j}$.

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

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

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 weights 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 $t_{1\rightarrow 1}^2$ corresponds to self-confidence that the agent $A_1$ has. In our model, $t_{1\rightarrow 1}$ 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 $\chi_{i\rightarrow j}$ of $A_1$. The general self-trust degree of an agent $A_i$ is then given by:

$$t_{i,i} = \frac{1}{3 \times (n-1)} \left[ \sum_{j=1, j \neq i}^{n} (\chi_{i,j}^{know}) + \sum_{j=1, j \neq i}^{n} (\chi_{j,i}^{know}) - \sum_{j=1, j \neq i}^{n} (\chi_{i,j}^{know}) \right]$$

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 $t_{i\rightarrow j}$ 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 pursuit of a dialogue is also based on a measure over the information exchanged.

For that purpose, we define a measure of the pertinence $p_j$ or $p_i$ of a content information. The pertinence is modelled by the lexical intersection defined in the Wordnet 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 $\chi_{i\rightarrow j}$ for $A_i$ and in the channel $\chi_{j\rightarrow i}$ for $A_j$, for example. For one agent $A_i$, for example, the information $\chi_{i\rightarrow j}$ for $A_i$ and in the channel $\chi_{j\rightarrow i}$ for $A_j$. For example. For one agent $A_i$, for example, the information

2the symbol $\rightarrow$ symbol is merely a convention for the expressivity of the symbol.
exchanged during a dialogue is stored in the lists $\mathcal{h}_{i\rightarrow j}^{\text{told}}$ and $\mathcal{h}_{i\rightarrow j}^{\text{told}'}$ 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

$$\text{Focus}(i \leftrightarrow j, A_i) = \bigcup \{ \mathcal{h}_{i\rightarrow j}^{\text{told}}, \mathcal{h}_{i\rightarrow j}^{\text{told}'} \}$$

where $\text{told}$ stands for the last three elements of the list $\text{told}$, and $\text{told}''$ stands for the last three elements of the list $\text{told}''$.

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

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 $KB$ 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 $\mathcal{h}_{i\rightarrow j}^{\text{told}'}$ of a channel:

\[ \text{The pertinence of each new content information is also checked via the formula given above.} \]

If, in a channel, there are no more items in the $\text{told}''$ list, after a check in the $\text{told}$ 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 $\sigma_r$ to an agent $A_r$, the receiver. The sentence $\sigma_r$ is called a sentence to differentiate it from a query, because $\sigma_r$ 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 $i \in KB$ 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 $KB_r$, 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 $S_i$ 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 CHANGE TOPIC 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 pertinent. 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 conversations policies for our system the two following schemes:

- **DUE**: \( A_1 \) 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, \( A_2 \ldots A_n \).

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

- **ASK ONE**: \( A_1 \) asks \( A_r \) for only one content information matching his query \( q_{ir} \).

- **ASK ALL**: \( A_1 \) asks \( A_r \) for all the possible content informations \( i_1, \ldots, i_n \) matching his query \( q_{ir} \).

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 rules 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 **CURRENT FACT**.

Basic definition and visualization functions are also defined for agents **DEFA GENT** 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**:

\[
(A_r, i_{new}) : \\
\text{if } (i_{new} = f \in KB_r) \text{ then } A_r \text{ sends a } KNOW(f), \text{ and } ch_{r \rightarrow f}^{know} = ch_{r \rightarrow f}^{know} + 1; \\
\text{else if } (i_{new} \cap KB_r \neq \emptyset) \text{ then } \text{ if } (i_{new} \text{ is a fact}) \text{ then } A_r \text{ creates new fact in } KB_r; f, NMAX AX + 1 = i_{new}, \\
\text{and } \text{NMAX } = \text{NMAX } + 1, \text{ and send a } AGREE, \text{ and } ch_{r \rightarrow f}^{agree} = ch_{r \rightarrow f}^{agree} + 1; \\
\text{else } A_r \text{ creates new rule in } KB_r; f, \text{ MMAX AX } + 1 = i_{new}, \text{ and } MMAX = \text{MMAX } + 1, \text{ and send a } AGREE, \text{ and } ch_{r \rightarrow f}^{agree} = ch_{r \rightarrow f}^{agree} + 1; \\
\text{else } A_r \text{ sends a } REJECT, \text{ and } ch_{r \rightarrow f}^{agree} = ch_{r \rightarrow f}^{agree} + 1; \\
\]

The selection by say \( A_r \) of an information content to send in a transaction is done with **NEXTFACTCHOOSE**. The complete algorithm of this function is not given here because it is rather monotonous, 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 \( A_r \) and selects the pertinent ones (via lexical connectivity with the focus) among them. Then, from this candidates list, he removes the ones already told (using the **told** of the channel with **CH**). If no candidates are left, then the dialogue is finished, \( A_r \) can either change the focus by using the **CHANGE TOPIC** 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 \( KB \), 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 semantic 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 **DUE** and defined as a communication policy, is done by the function **DUE**. In this function, first it is tested whether the two channels \( ch_{A \rightarrow B} \) and \( ch_{B \rightarrow A} \) already exist (previous dialogues), otherwise they are created and all their variables are initialized (counters). A fact is selected in \( KB_r \) with the function selection function of the type of **NEXTFACTCHOOSE**. At the beginning the only facts or rules that are in the **told** list are those coming from dissonances. This is the actual drive of \( A_r \) to start the dialogue. Here again, the dialogue may continue as long as there is something pertinent in the **told** 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 being run. For example, the simplest interaction can be run by using the ASKONE policy:

```
RUN[ASKONE [r1];
    ******** 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[r1]}
    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 of REJECTFACTCHOICE described above. In this example, A 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_B. As soon as A_B will learn anything about r (pertinent) he will put this new information in his total list for 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 as a whole conversation policy: a DUG.

```
RUN[DUG[c,B];
    ******** DUG[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],?1},
    CHANGEFACTIC[fC71,INFORM[fB98]]
    REJECT fC71 REJECTS= {B,?1}
    REJECTCOUNT= 1 CHANGEFACTIC.
    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 fB98 = r[3],
    ------ LOOP stopped.
    B has nothing NEW to say.
    B -> C
    ACTLIST={AGREE[fC70],INFORM[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 connectivity 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 not 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.
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 fucus of the conversation.

Pasquier et al.'s [3] already proposed that the dissonance theory could bring interesting highlights to the problem of structuring dialogue between agents. The idea is that is could be a model for the opening of such a dialogue 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 perforations 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 valued by the agents with their value (true, false, unknown) and their confidence degree where agents are valued 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, \( A_1 \), 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 \( M_1 \). In such a case, \( A_i \) will open a dialogue with \( A_1 \) 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.

6. REFERENCES


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