T-Compound Interaction
and
Listening Agents

Eric Platon¹, Nicolas Sabouret², and Shinichi Honiden³

¹ National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda, 101-8430 Tokyo
² Laboratoire d’Informatique de Paris 6, 8, Rue du Capitaine Scott, 75015 Paris
³ {platon, honiden}@nii.ac.jp, nicolas.sabouret@lip6.fr

Abstract. This paper proposes a formal model of interaction featuring the recent concept of overhearing in Multi-Agent Systems. This enacts indirect interactions whereby listening agents hear the conversation between two or more other agents that subscribe to this mechanism. We represent overhearing with an interaction composite named the T-compound, so that it is distinguished from traditional direct interactions. The formal model is based on the π-calculus owing to its intrinsic focus on interactions. We exploit its syntax to build the T-compound, and describe a method for basic design of the interaction dimension in MAS. From this work, we intend to increasingly embody features of interactions such as mobility, scalability, and security to avoid eavesdropping.

1 Introduction

Multi-agent systems (MAS) rely on the dynamic of their constituting nodes and links, namely the interactions among their distributed agents, either human, hardware or software. Our work highlights the design of interactions in MAS, and we think relevant to exploit the recent concept of overhearing [2,3] in addition to traditional point to point interactions. This notion refers to indirect interactions that occur frequently in natural systems [4], typically when two agents are discussing and a third one is just a passive audience that could however intervene if required. Exploitation of both direct and indirect interactions leads to new perspectives on systems, and we will show some situations whereby this double usage is even necessary. Research about overhearing is still in early stages so that few applicative demonstrations and models were proposed so far. In this context, we intend to build a formal model named the T-compound, based on the π-calculus [5].

This paper begins in Section 2 with a presentation of the concept of overhearing, its interest for MAS, and the relevance of formal modelling with the π-calculus. Then, we propose in Section 3 our model for cooperative agents and a methodology to describe the interaction dimension in MAS. In Section 4, we develop an example with the proposed method. Finally, we discuss our approach in Section 5, before concluding and opening this work.
2 MAS, Overhearing, and $\pi$-calculus

2.1 MAS and Overhearing

The notion of overhearing was recently introduced in the MAS community as reported by Gutnik et al. [6]. It was originally proposed to endow agents with monitoring abilities to detect failures in the apparent behaviour of other agents, i.e. their interactions. This concept means one agent has the capacity to capture information from the interaction of two or more other agents. Fig.1 depicts such a situation in its simplest case.

![Fig. 1. Overhearing Situation](image)

MAS presently exploit direct interactions, such as the discussion link between agents A and B on Fig.1. Overhearing is an instance of indirect interaction that might be relevant in the MAS paradigm. In fact, research in the field of natural sciences show that numerous MAS based on complex societies exploit meaningful forms of communication without explicit receiver. For instance, termites build their hills by working together, but they do not exchange any information directly. They rely on the notion of stigmergy whereby they determine their behaviours according to the current state of the environment. One termite puts a piece of material for the hill, and other termites (including the first one) will then pile on top, toward their common purpose [7].

Indirect interactions already leveraged relevant results in MAS with stigmergy and other techniques [2]. Overhearing has now an increasing number of applications —probably due to its high occurrence in our daily-life. Original work refer to monitoring [8], group communication [2], and some forms of coordinations [9–11]. We add to this list our ongoing work on intelligent assistants: users may interact with heterogeneous distant agents (web services, another user, etc.) and an overhearing assistant could provide advanced support initiatives according to the interaction context [12]. In addition to these explicit references to overhearing, we found the phenomenon often appears in agent systems that focus on different concerns. Hence, the Helper Agent reacts to silences in an instant messaging discussion between humans to suggest common topics and increase the interest of the participants [13]. The M system surveys the actions of people in a virtual meeting room to optimise at run-time their workspaces [14]. COLLAGEN observes the flight ticket reservation process of the user to propose alternatives when no solution can be found for a given request [15].
the remainder of this paper, we attempt to make more systematic the usage of overhearing in such systems, and possibly in novel applications.

2.2 Interaction and Formal Model

In order to exploit different kinds of interactions, MAS developers will need a representation scheme. Furthermore, overhearing leads to increasing the complexity of the network weaving among all agents. For \( n \) agents, one count \( O(n^2) \) direct interactions and \( O(n^3) \) indirect ones\(^1\). Formal models consist in proper representations of phenomena and remain mostly neutral regarding implementation details. Also, they can clarify the view of the system and reduce the effects of complexity by using mathematical compact formulae, based for instance on sets or recursivity.

In consequence, we propose in this paper a formal model of interaction including overhearing. Such a model was already proposed once as far as we know in [6], so that a comparison of the two approaches is in Section 5. One original feature of our framework is to rely on the \( \pi \)-calculus from Milner [5] to leverage its interaction- and dynamism-oriented syntax, mechanisms, and expressive power that ‘can in principle model (...) any computational aspect of agents’ [16].

This heritage provides a robust model of traditional interactions and our extension enacts an instance of overhearing. In addition to the grounds provided with the calculus, Milner developed a set of techniques to study concurrent systems, including structural congruence and bi-simulation relations. In the former, two agents are equivalent if they have the same interaction patterns. In the latter, the equivalence is defined in terms of apparent behaviours. We expect the comparison between interaction structures and between external behaviours will enable advanced reasoning capabilities in agents exploiting the T-compound (e.g. conversation recognition [6]). In particular, one agent may hear a conversation and try to match the stream to a known protocol. As it is unlikely to perfectly match an existing models (well-known problem in case-based reasoning) [6], the notion of equivalence allows more flexibility. In the remainder of this paper, we focus first on the syntax of the formal model, whereas the exploitation of equivalence belongs to our future work.

2.3 The \( \pi \)-calculus in this paper

The model presented in this paper exploits a subset of the \( \pi \)-calculus, originally from R. Milner [5]. This section aims at explaining the elements we retained and their meaning. The \( \pi \)-calculus is a modern process algebra for concurrent systems. Its main features include the representation of interactions among concurrent agents and the mobility of links that stand for both agent mobility and changes in the interaction network (reorganisation, life-cycle of agents).

\(^1\) These two complexities refer to the complete set of possible interactions among \( n \) agents. However, implementations such as blackboards and peer-to-peer reduce significantly these values. They appear here as illustrations of the general case.
\[ \mathcal{P}_\pi \] is the set of agent names denoted by capitalised words. The set of Greek letters \( \mathbb{N} = (\alpha, \beta, \ldots) \) represents the interaction channels that can link two agents. Other strings and small characters in the set \( \text{Str} \) label the messages that are sent in the different channels. Finally, \( I \) is an interval of integers \([0,1,2,\ldots]\).

**Syntax.** We now define the well-formed formulae (wff) of our restricted \( \pi \)-calculus. The following gathers operators and notations that the formal model exploits; the next part then defines their semantics.

- `'.' (dot) is the successor operator
  \[ \rightarrow \text{It accepts two signatures as follows. Given the channels } \alpha, \beta \text{ and the agent } P, \text{ the well-formed formulae are } \alpha.P \text{ and } \alpha.\beta \]
- '+ represents the choice operator
  \[ \rightarrow \text{It allows writing } P+Q \text{ for any agent } P \text{ and } Q. \text{ The generalisation allows composing an arbitrary number } n \text{ of agents: } \sum_{i=1}^{n} A_i = (A_1 + \ldots + A_n) \]
- '| is the composition operator
  \[ \rightarrow \text{It accepts the formula } P \mid Q \text{ for any agent } P \text{ and } Q, \text{ and the generalisation: } \prod_{i=1}^{n} A_i = (A_1 \mid \ldots \mid A_n) \]
- '⟨.⟩ and (.) represent the sending and reception operators
  \[ \rightarrow \text{They accept the same syntax for any channel } \alpha \text{ and message } x: \alpha⟨x⟩ \text{ and } \alpha(\text{x}). \text{ The operator omission denotes any of them can be used} \]
- '0 is the null agent, in addition to \( \mathcal{P}_\pi \)
  \[ \rightarrow \text{It is usually omitted at the end of definitions, i.e. } P = \alpha.Q \text{ is written instead of } P = \alpha.Q.0 \]

Finally, the well-formed agents verify the following equation:
\[ P ::= 0 \lor \alpha.P_0 \lor \sum_{i \in I} P_i \lor \prod_{i \in I} P_i \quad (1) \]

An agent \( P \) is either of the items in this formula, and also their composition due to the recursive definitions.

**Semantics.** First, the null agent \( 0 \) is constant and has no activity (neither internal nor interactive). It is a final state that represents the termination (end of life) of the agent.

The successor operator `'.' describes the usage sequence of channels. The case \( \alpha.P \) means \( \alpha \) is used and then the agent behaves as \( P \). Also, \( \alpha.\beta.\gamma \) denotes a sequence of three channels leading to the null agent.

The sum of agents relies on the usual choice operator. \( P \) can behave as any member of the sum. For instance, \( P = \alpha.P_0 + \beta.P_1 \) will evolve as \( P_0 \) if \( \alpha \) is used, and as \( P_1 \) if \( \beta \) is triggered.

Similarly, the parallel operator 'I' represents the composed execution of \( P_0 \) and \( P_1 \). The focus on interactions of the calculus composes agents by communication channels as presented hereafter in the system evolution.

Along the interaction channels, two complementary actions can occur, namely the sending and reception of messages. The formulae \( \alpha(\text{x}) \) and \( \beta(\text{y}) \) respectively mean that the message \( x \) is sent through \( \alpha \) and \( y \) is received through \( \beta \).
System Evolution. The π-calculus defines how systems evolve. The basic mechanism is the reaction between two composed agents (parallel execution) along a common channel; one sending a message and the other one receiving. Let’s illustrate how this is run.

\[
P \overset{\text{def}}{=} (\alpha(x).P_0 + \beta(y).P_1) \mid \alpha(x).0
\]  

According to the reaction rule, \(\alpha(x).0\) reacts with the first element \(\alpha(x).P_0\) of the sum, so that \(x\) is passed through \(\alpha\). The second element of the sum is discarded (choice) and the system becomes (we use the intuitive property that \(A\) in parallel with 0 is equivalent to \(A\)) [5]:

\[
(P_0) \mid 0 = P_0
\]

The π-calculus features much more syntactic elements and advanced notions, but the present notations and mechanisms are sufficient in this paper.

3 T-compound Model

3.1 T-compound Formula

Informally, the T-compound is depicted on Fig.1, page 2; the shape of the interaction justifying the name of the composite. The T-compound is formally a 6-tuple that verifies structural properties.

Let us consider 3 different agents \((A, B, P) \in P^3\) and 3 distinct channels \((\alpha, \beta, h) \in \mathbb{N}^3\) for the communications \((A, B), (B, A),\) and \((A, P)\) respectively (see Fig.2). Let us also define 3 agents \(M_X\) for \(X \in \{A, B, P\}\) with the following notation and formulae:

\[
\begin{align*}
\text{Notation} : \quad & \mathbb{N}^* \overset{\text{def}}{=} \{ \mu \mid \exists n \in I, \exists (\mu_i)_{i \leq n} \in \mathbb{N}^n \text{ so that } \mu = \mu_1 \ldots \mu_n \} \\
& \mathbb{N}^* \text{ is the set of arbitrary sequences of channels composed with } \cdot
\end{align*}
\]

\[
\begin{cases}
\exists (\epsilon_i)_{i \in I}, \forall i \in I \epsilon_i \in \mathbb{N}^*, \epsilon_i \neq \alpha & M_A = \sum_{i \in I} \epsilon_i A \\
\exists (\delta_i)_{i \in I}, \forall i \in I \delta_i \in \mathbb{N}^* & M_B = \sum_{i \in I} \delta_i B \\
\exists (\gamma_i)_{i \in I}, \forall i \in I \gamma_i \in \mathbb{N}^* & M_P = \sum_{i \in I} \gamma_i P
\end{cases}
\]  

The T-compound is defined in its basic form by the next formula (5). This is simply referred to as ‘T’ to model that \(P\) can only hear \(A\). The case whereby \(P\) can also hear \(B\) is the ‘full T’ and is composed from the single one, as explained in the next section 3.2.

\[
\begin{align*}
A & \overset{\text{def}}{=} \alpha(x).h_\alpha(x).A + \beta(y).A + M_A \\
B & \overset{\text{def}}{=} \beta(y).B + \alpha(x).B + M_B \\
P & \overset{\text{def}}{=} h_\alpha(x).P + M_P
\end{align*}
\]  

\[\overset{\text{def}}{=} (A|B|P)\] where
Note that a channel between two agents does not mean they are actually interacting, since this depends on their intentions or imposed protocols. Instead, channels represent which interactions are possible at this level of modelling. For example, a hierarchy of agents usually allows bilateral communications along the social ladder (channels do exist), but sometimes there is no practical exchange.

The definition of the compound means a T is the parallel execution of three agents, each of them playing specific interactions. Agent definitions are recursive including a term $M_X$. The recursion represents the interaction cycle of agents as expected along their lives, that is agent $A$ chooses one action, performs it, and then recovers the capability to choose from the initial action set. The term $M_X$ features the behaviours of agents out of the T to keep the formula general enough, and it may be equal to 0. Among the three $M_X$, the case $A$ has one special constraint $\forall i \in I \epsilon_i \neq \alpha$. This is required to express that $A$ cannot use the channel $\alpha$ out of the T, so that the commitment for overhearing remains coherent.

In the agent formulae, $A$ can first send $x$ through $\alpha$ and has to send it through $h_\alpha$ to return to its initial state (otherwise the $\pi$-calculus syntax means the agent is blocked, waiting for triggering $h_\alpha$). Thus, $B$ receives the message as a direct interaction (the primary intention of $A$) and $P$ receives a copy representing the listening. Moreover in the definition, $A$ can receive $y$ through $\beta$ from $B$, or behave as $M_A$. Symmetrically, $B$ can send $y$ via $\beta$, receive $x$ via $\alpha$, or behave as $M_B$. Finally, $P$ can receive $x$ as overheard message through $h_\alpha$, or behave as $M_P$. Fig.2 shows the correspondence with the formula. In the $\pi$-calculus, we represent overhearing as the copy of a direct channel, i.e. the dashed arrow is actually $h_\alpha$ as copy of $\alpha$.

This model assumes agents agree to be overheard, eavesdropping being a special security issue that could be modeled the same way as overhearing. In effect, the T-compound is only the infrastructure that support agent intentions to exploit the model.

In some conditions, one may need to authorise overhearing by default. Some applications would be for simulation of realistic phenomena or in some kinds of forum rooms. However, this implies intricate security concerns in real-world open MAS, so that we first focused on a tacit hypothesis of cooperative agents. Despite this apparent limitation, many MAS applications contain groups of cooperating
agents that trust each other. In this frequent context, the T-compound of this paper can be applied.

### 3.2 Interaction Design Elements

From the foundation formula of the T-compound, we derive in this section a formal framework to seamlessly exploit overhearing aside direct interactions in MAS.

Given $S \subseteq \mathcal{P}_T$ an agent set (the system to be modeled), we define three properties over the subsets of $S$, namely $TALK$, $CONVERSE$ and $OVERHEAR$. In the following equations, $X \models \phi$ means that the set of agent $X$ satisfies $\phi$.

\[ \forall S_0 \subseteq S : \quad S_0 \models TALK(A, x, B) \iff \begin{cases} |S_0| \geq 2, (A, B) \in S_0^2, A \neq B, \\ x \in \text{Str} \\ \exists \alpha \in \mathbb{N} \text{ so that } \\ A = \alpha(x).A \text{ and } \\ B = \alpha(x).B \end{cases} \]  

(6)

\[ \forall S_0 \subseteq S : \quad S_0 \models CONVERSE(A, x, B, y) \iff \begin{cases} TALK(A, x, B) \\ TALK(B, y, A) \end{cases} \]  

(7)

In $TALK$, one agent sends $x$ to another agent directly. $CONVERSE$ then requires two symmetric instances of $TALK$, and this property is actually a writing facility for the next formulae. When considering any message between $A$ and $B$, we will note $TALK(A, B)$ and $CONVERSE(A, B)$.

\[ \forall S_0 \subseteq S : \quad S_0 \models OVERHEAR(A, x, B, P) \iff \begin{cases} |S_0| \geq 3, \\ P \in S_0 \land P \neq A, P \neq B \\ TALK(A, x, B) \\ TALK(A, x, P) \end{cases} \]  

(8)

This property denotes that when $A$ sends $x$ to $B$ and $P$ is listening, then $A$ has to send the same information to $B$ and $P$. When considering any message between $A$ and $B$, we will note $OVERHEAR(A, B, P)$. This formula differs from the definition (5) since $A$ and $B$ are not engaged yet in a conversation (only $A$ talks), and this is the purpose of the next properties. We need to distinguish these cases for the model granularity. Indeed, the concept of overhearing appears as soon as an agent can hear messages for others. However, this is insufficient for an interaction, since it requires a complete discussion, i.e. a conversation. Hence, the T-compound $A^T B$ is finally the combinations:

\[ \forall S_0 \subseteq S : \quad S_0 \models A^T B \iff \begin{cases} OVERHEAR(A, B, P) \\ TALK(B, A) \end{cases} \]  

(9)

\[ \forall S_0 \subseteq S : \quad S_0 \models A^T_{full} B \iff \begin{cases} OVERHEAR(A, B, P) \\ OVERHEAR(B, A, P) \end{cases} \]  

(10)

$A^T B$ corresponds to (i) $P$ overhearing talks from $A$ to $B$ and (ii) talk from $B$ to $A$. This last talk is required to express that $A$ and $B$ can discuss, to match
(5), \( \overline{A}T^B_{full} \) completes the previous case so that \( P \) also overhears talks from \( B \). Table 1 summarises the basic elements we use in this formal model to represent MAS interactions including overhearing.

### Table 1. Interaction Design Elements

<table>
<thead>
<tr>
<th>Formal Element</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TALK(A,B)</td>
<td>A sends messages to B.</td>
</tr>
<tr>
<td>CONVERSE(A,B)</td>
<td>A and B converse.</td>
</tr>
<tr>
<td>OVERHEAR(A,B,P)</td>
<td>A sends messages to B and P hears them.</td>
</tr>
<tr>
<td>( \overline{A}T^B_{full} )</td>
<td>A and B converse and P hears A's talks.</td>
</tr>
<tr>
<td>( \overline{A}T^B_{full} )</td>
<td>A and B converse and P hears both talks.</td>
</tr>
</tbody>
</table>

### 3.3 MAS Interactions with our Model

From this set of formulae and a set-based representation of MAS, we propose to exploit our formal model in the description of two views of a given system, namely the system-level and agent-centred interaction sets. These two tools may be of use in the design of the interaction dimension of MAS, as they provide points of view orthogonal to the traditional interaction protocol concerns. The next section is devoted to an example to show the usage of the abstract formula introduced in this part.

Given an agent set \( S \subseteq \mathcal{P} \), we define \( \mathcal{I}_S \) as the system-level set of permitted interactions in \( S \), and \( \mathcal{A}_S \) as the agent-centred view of \( S \). In order to exploit our model to build these two views, we follow a methodology depending on system specifications.

Specifications should constrain possible interactions in the system and end with communication rules or interaction protocols. Consequently, they provide the basic information to be compiled in \( \mathcal{I}_S \). The next sequence shall be performed from the initial state \( \mathcal{I}_S = \emptyset \). Interaction patterns are identified and the corresponding agent tuples are added to \( \mathcal{I}_S \), until complete coverage of the specification requirements.

1. Select all 3-agent tuples that verify \( \overline{A}T^B_{full} \) and input them in \( \mathcal{I}_S \)
2. Then same process for 3-agent tuples that verify \( \overline{A}T^B \)
3. Then same process for 3-agent tuples that verify \( OVERHEAR(A,B,P) \)
4. Then same process for 2-agent tuples that verify \( CONVERSE(A,B) \)
5. Then same process for 2-agent tuples that verify \( TALK(A,B) \)

The sequence ends with an interaction set in which interactions are not redundant. That is the most complex interactions are first captured so that the next steps only add simpler compounds that could not fit the complex one. The
formula (11) shows the general form of $I_S$. The character ‘∗’ stands for an arbitrary number of occurrences of the compound. The absence of property signature means compounds are applied to any potential agents.

$$I_S = (TALK^∗, CONVERSE^∗, OVERHEAR^∗, T^∗, T^*_\text{full})$$  \hspace{1cm} (11)

In addition to the focus on the interaction dimension of MAS, $I_S$ can be rewritten following the rules of the $\pi$-calculus, shifting to the agent-centred view of $A_S$. As all elements from Table 1 are defined in the calculus syntax, we can develop all compounds in $I_S$, group the results by agent, and compile $A_S$ as the parallel execution of all these agents (the complete system). This rewriting process ends with the general formula:

$$A_S = \prod_{i \leq |S|} \sum_{j \leq I_i} \alpha_j A_i$$  \hspace{1cm} (12)

where for integers $i$ and $j$, $I_i$ is the number of interactions for agent $A_i$, $\alpha_j$ are elements of $\mathbb{N}^*$ as defined in formula (4). This view can be relevant when studying roles in interaction protocols, since a single formula contains all the interactions of one agent. Also, this can be useful at the MAS programming level, for instance in the case of the ‘Interaction-Oriented Model by Textual representation IOM/T’ in [18] where the approach of interaction protocols is orthogonal to the design of other parts of the MAS, similarly to our model.

4 Example: The Board of Directors

This example models a meeting among the head of a company and its division directors. With another vocabulary, our system targets a user and its software advisor agents. In the following, we suppose all agents can listen to all discussions, and the user is put aside to receive the final advice from the completed debate.

Given $n \geq 3$ agents $(A_i)_{i \leq n}$ and the integers $i$ and $j$, $\alpha_{ij}$ is the communication channel from $A_i$ to $A_j$. The agent $U$ represents the user interface that compiles the final report from the board and $c_i$ the corresponding channel from agent $i$. Consequently, the complete system is $S = \{(A_i)_{i \leq n}, U\}$. Hereafter is the general $I_S$ formula for $n$ agents, and we then detail the case $n = 3$ to illustrate the methodology.

$$I_S = \{\langle A_k T^* A_j \rangle_{i < j < k}, \langle TALK(A_i, c_i, U) \rangle_{i \leq n}\}$$  \hspace{1cm} (13)

The first term of $I_S$ represents the discussions among the advisors and their ability to overhear conversations in the meeting room, even if they are not active participants. The second term is the final report from each agent to the user interface $U$.

Let us now study in more details the case $n=3$. Fig.3 shows the interactions that must appear according to the scenario specifications.

The application of the methodology based on the specifications yields the following $I_S$. 

1. ‘all agents can listen to all discussions’ leads to $A_1.A_2.T_{full}', A_2.A_3.T_{full}, and A_1.A_3.T_{full}$
2. $P.T.B \rightarrow \text{None remaining}$
3. $OVERHEAR(A, B, P) \rightarrow \text{None remaining}$
4. $CONVERSE(A, B) \rightarrow \text{None remaining}$
5. ‘user receives the final advice’ gives $(TALK(A_i, U))_{i \leq 3}$

$$\mathcal{I}_S = \{(A_i.T_{full})_{i<j<k<3}, (TALK(A_i, c_i, U))_{i \leq 3}\}$$ (14)

The final formula corresponds to the general case as expected and represents all interactions occurring in $S$. If we develop the terms using the property definitions (6) to (10) and we group elements according to agents, we obtain $S$ for $n=3$:

$$A_S = (\prod_{i=1}^{3} A_i)|cU$$ (15)

where $c = c_1(r_1).c_2(r_2).c_3(r_3)$. For all $i \leq 3$, the formula of $A_i$ is similar to the one for $A_1$:

$$A_1 = (\alpha_{12}(x_{12}).\alpha_{13}(x_{13}).A_1 + \alpha_{13}(y_{13}).\alpha_{12}(y_{12}).A_1 + //A_1 \text{ talks,}$$

$\quad \alpha_{21}(x_{21}).A_1 + \alpha_{31}(x_{31}).A_1 + //\text{others overhear}$

$\quad \alpha_{21}(x_{23}).A_1 + \alpha_{31}(x_{32}).A_1 + //\text{One talk to A}_1$

$\quad c_1(r_1).A_1) //A_1 \text{ reports}$ (16)

From the formulae (14) and (16), system designers can respectively obtain a model of all interactions including overhearing and the list of interactions per agent. The former can be useful to constrain the system (e.g. prevent some overhearing), and the latter leads to the definition of roles in interaction protocols and offers an alternative view of the lifeline in AUML [19].

5 Related Work

Gutnik and Kaminka proposed recently the first formal model of overhearing in the MAS community [6]. Their purpose was to identify conversations among
agents and they model this mechanism with a conversation system and recognition algorithms. Their representation embodies conversation notions (roles, states, transitions, speech acts, etc.) and the practical exploitation for their issue of identification. Although they propose in this article a ‘comprehensive formal model of the general problem’ of overhearing, this first attempt is specialised to a peculiar usage of the concept, i.e. conversation type identification. Our model is based on the ‘general’ representation of interactions provided by the π-calculus, and we attempted to extend it for an application independent model of overhearing that might be of use in MAS interaction design.

Busetta et al. proposed an implementation architecture of overhearing [2]. Albeit this work is not a formal model, it stands close to our proposal. The article describes a multicast communication among agents in a cooperative group. When taking on a channel defined by a discussion topic, all authorised listener agents receive the information. Formally, this architecture implements a generalised version of our T-compound. The term $\alpha(x).h_\alpha(x).A$ from the definition (5) is extended to send copies for overhearing at an arbitrary number of listeners. However, this would produce heavy formulae with our current syntax, so that we ought to consider another approach in future work as stated hereafter.

6 Conclusion and Opening

In this paper, we proposed in π-calculus a formal model of interaction that embodies the recent concept of overhearing, represented here as an interaction composite named the T-compound. The aim of this model is to provide a general description of interactions in MAS, orthogonally to other design issues (agents, environment or organisation). In addition, a methodology was proposed to build two views for the study of MAS interactions. The first representation shows all interactions that can occur in a given system. Rewriting this set with the underlying calculus, we can also represent an agent-centred description of all interactions. These two views of the same system can provide MAS designers with relevant information for the analysis and design stage.

Our current model covers partially the interaction requirements of MAS. For instance, the notion of dynamism and mobility are not included yet. This will entice considering agents that have new acquaintances, join or quit dynamically the system, or become mobile. This was indeed a motivation to use the π-calculus, in addition to advanced features we intend to introduce as stated in section 2.2. We also pointed out that Busetta et al. proposed an implementation more general than the model presented in these pages. Although we can actually write it, the formulae become cumbersome to express and exploit. The scalability of such a syntax is rather low. Hence, we are working on the agent environment so that overhearing would be relayed through it. Work exist for similar ideas [9–11], but partially centralised so that we are developing an original approach. Finally, a counterpart of overhearing is eavesdropping. To avoid this unwanted leaks, we study how to integrate structuring elements in the formal model, one solution
being to introduce a network domain-like metaphor to manage interaction by
groups and enforce security.

References

1. Demazeau, Y.: ‘From Interactions to Collective Behaviour in Agent-Based Sys-