

# Une interface conversationnelle pour une aide intelligente

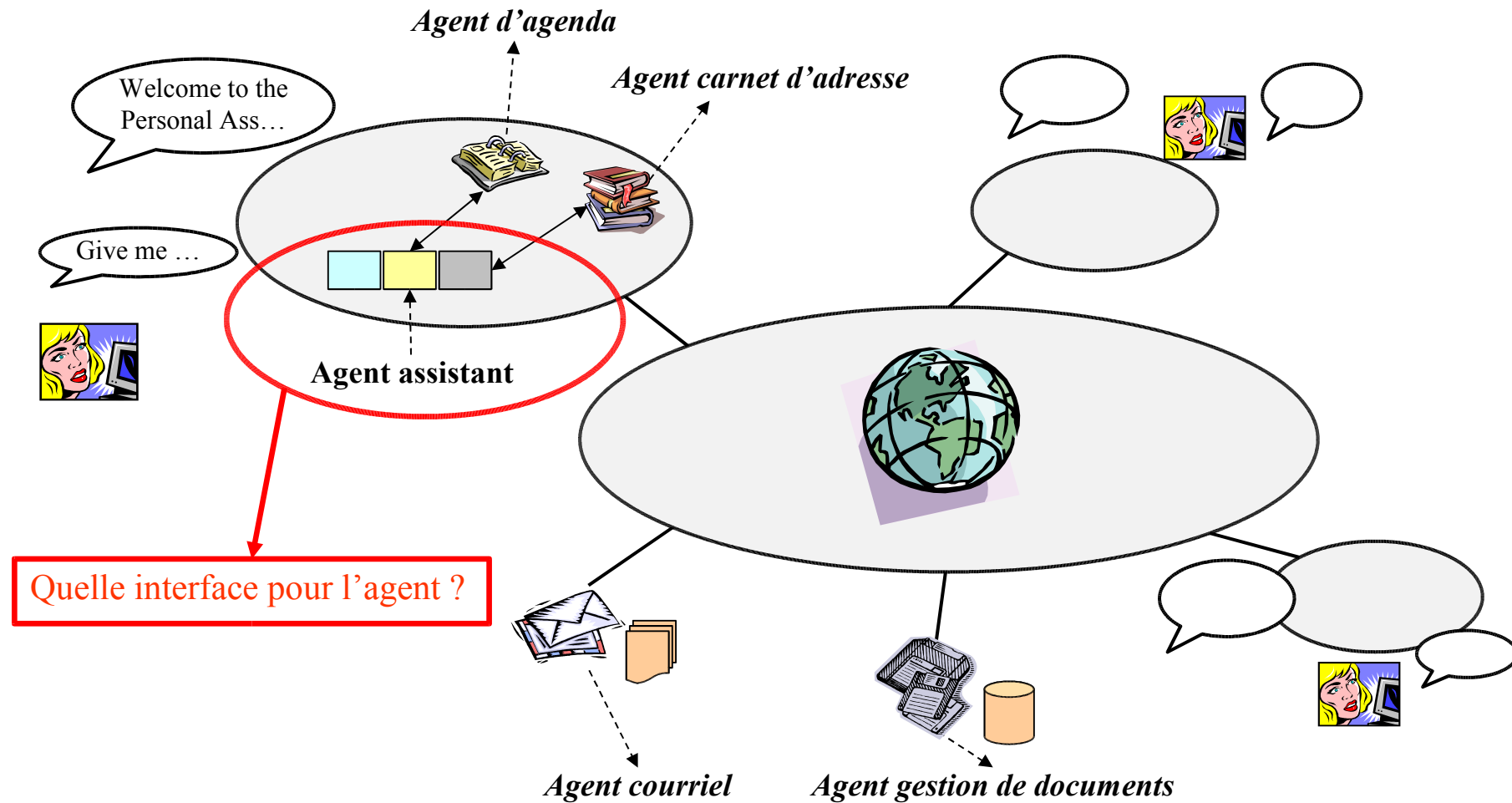
**Emerson Cabrera PARAIISO**  
**Jean-Paul A. BARTHES**

## Sommaire

- Contexte du travail
- L'agent assistant et son interface
- Problématique
- L'ICAI
- L'architecture de l'interface
- Le rôle des ontologies
- L'interaction entre les agents
- Conclusions

# Contexte du travail

- Plate-forme OMAS
- Plusieurs projets développés (surtout dans la gestion des connaissances)



## Qu'est ce qu'un agent assistant personnel?

- **Définition :**
  - Du point de vue informatique, un agent assistant personnel est un agent qui *joue le rôle d'assistant* et dont l'objectif est de *diminuer la charge de travail* de son maître pendant la réalisation des activités professionnelles qui demandent une *interaction avec l'ordinateur* ou un autre moyen informatique (F. ENEMBRECK, 2003).

*jouer le rôle d'assistant*

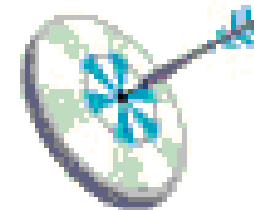
+

*diminuer la charge de travail*

+

*faciliter l'interaction*

=



# L'approche conversationnelle

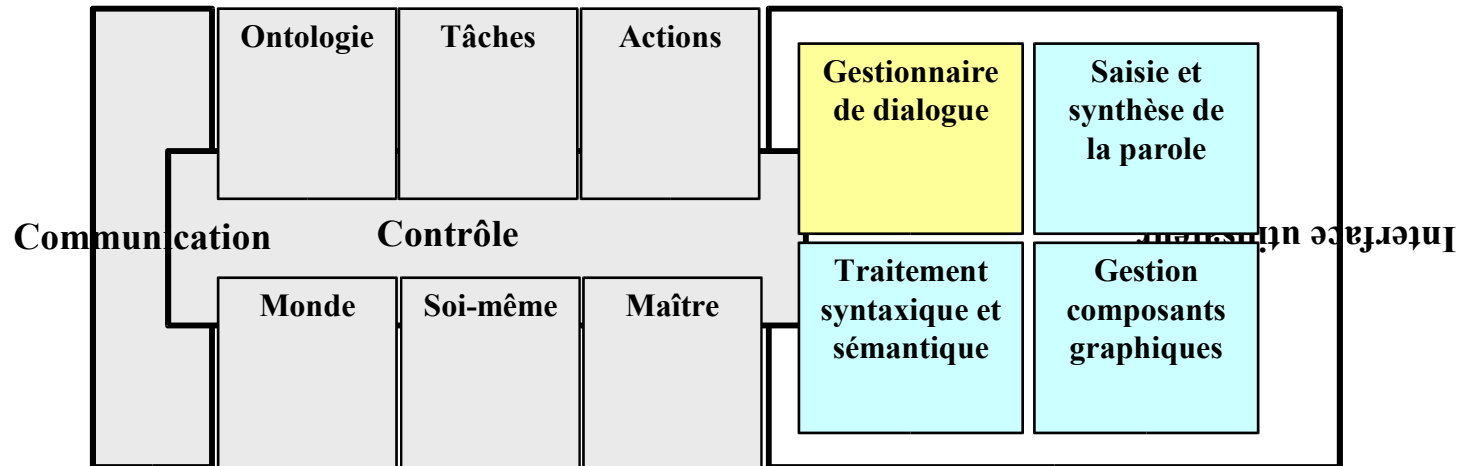
The screenshot displays the 'Personal Assistant Interface: SMAPI Version - user: Emerson'. The interface includes a 'Commands for your PA' list, a 'Notifications for FLORE' window, and a main dialogue area. The dialogue shows a sequence of interactions: the user asks 'What is the subject?', the system responds 'our meeting today', the user asks 'What is the content', the system provides details about a meeting and asks for confirmation to execute a task. The user confirms, and the system proceeds with the task, asking for an email address and confirming the execution of the message.

Below the dialogue, there are control elements: 'Micro Volume is: [input]', 'Word Correction is:  ON', 'Personal Agent is: **Connect**', 'Microphone is:  ON', and 'Debug is:  OFF'.

At the bottom, a file explorer window shows a list of files and folders, with a blue arrow pointing to a specific file.

**Amélioration de la qualité de l'aide par rapport à l'approche traditionnelle – WIMP (*w*indow, *i*con, *m*enu, *p*ointer)**

# L'agent assistant personnel et son interface



acteur	énoncé	action
Utilisateur	"Please, open my email account."	Ouvrir le client d'email
Utilisateur	"Locate and open the latest version of BUCKS.doc"	Envoyer la demande de document à l'agent staff responsable
Utilisateur	"List all meetings at my office for the next 3 days"	Envoyer la demande à l'agent staff responsable
Agent	"A new electronic message just came in"	Attendre la réponse de l'utilisateur
Utilisateur	"Look for documents on references to Assistant Agents"	Suggérer des meilleurs critères ou envoyer la demande à l'agent staff de recherche

# Problématique

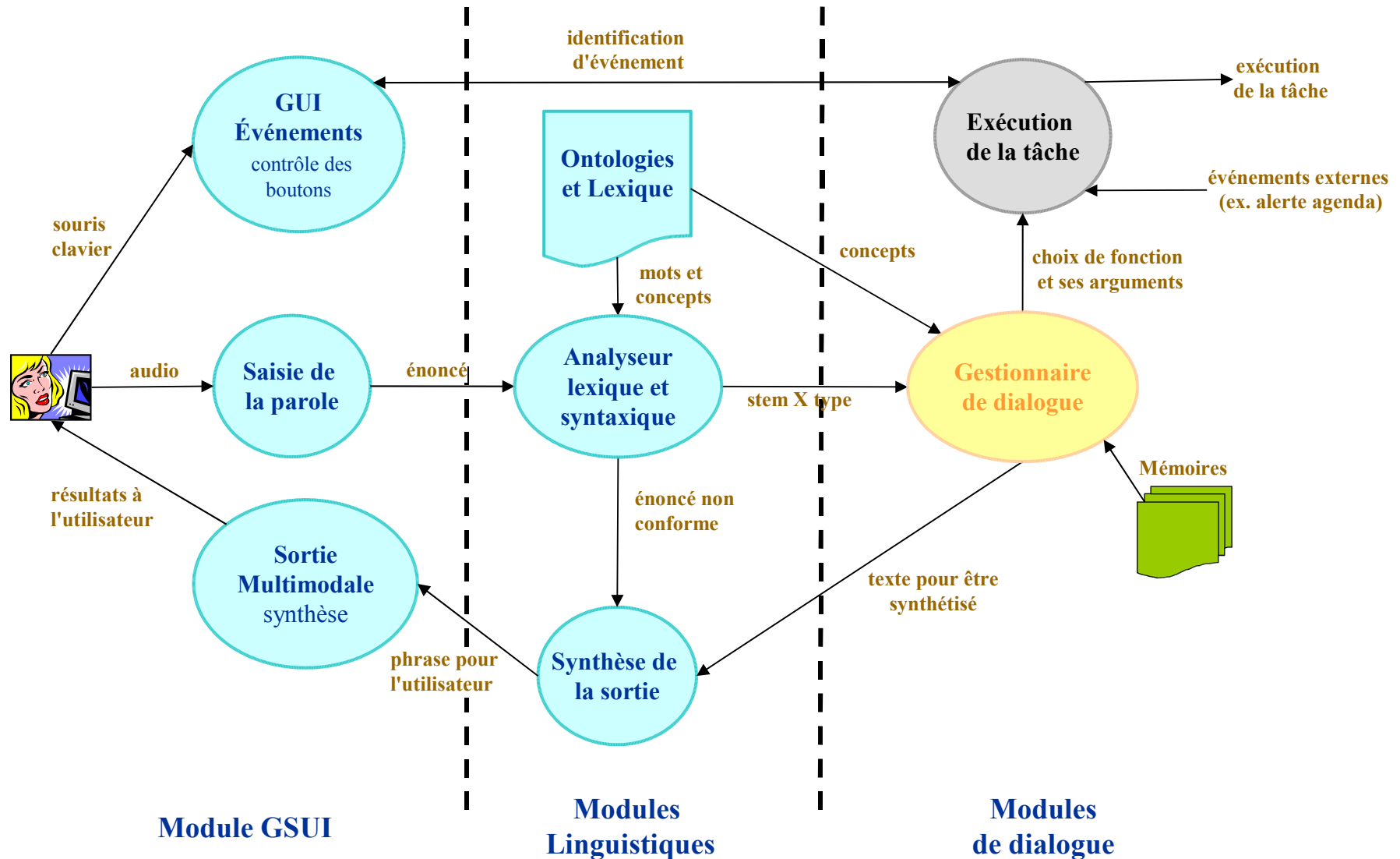
- Principaux défis :
  - conception d'un mécanisme d'analyse syntaxique, capable de traiter des énoncés parlés, pas forcément bien reconnus ;
  - conception d'un analyseur sémantique, capable d'interpréter les énoncés, pas toujours liés au domaine de l'application ;
  - conception d'un gestionnaire de dialogue capable de gérer des conversations sur des domaines spécifiques, mais aussi de déclencher plusieurs tâches simultanément et de bien gérer l'arrivée de messages provenant de plusieurs sources différentes (d'autres agents) ;
  - mise en place d'une structure de mémoires, capable de fournir des informations à la gestion du contexte de la conversation ;
  - conception d'une politique d'affichage d'informations, gérée par l'agent assistant, pour organiser la façon d'interrompre l'utilisateur : pour lui poser des questions et pour lui présenter des informations diverses, comme des résultats d'exécution des tâches ;
  - capacité d'ancrer le raisonnement de l'agent sur les ontologies dont l'agent dispose.

# Interface conversationnelle pour une aide intelligente

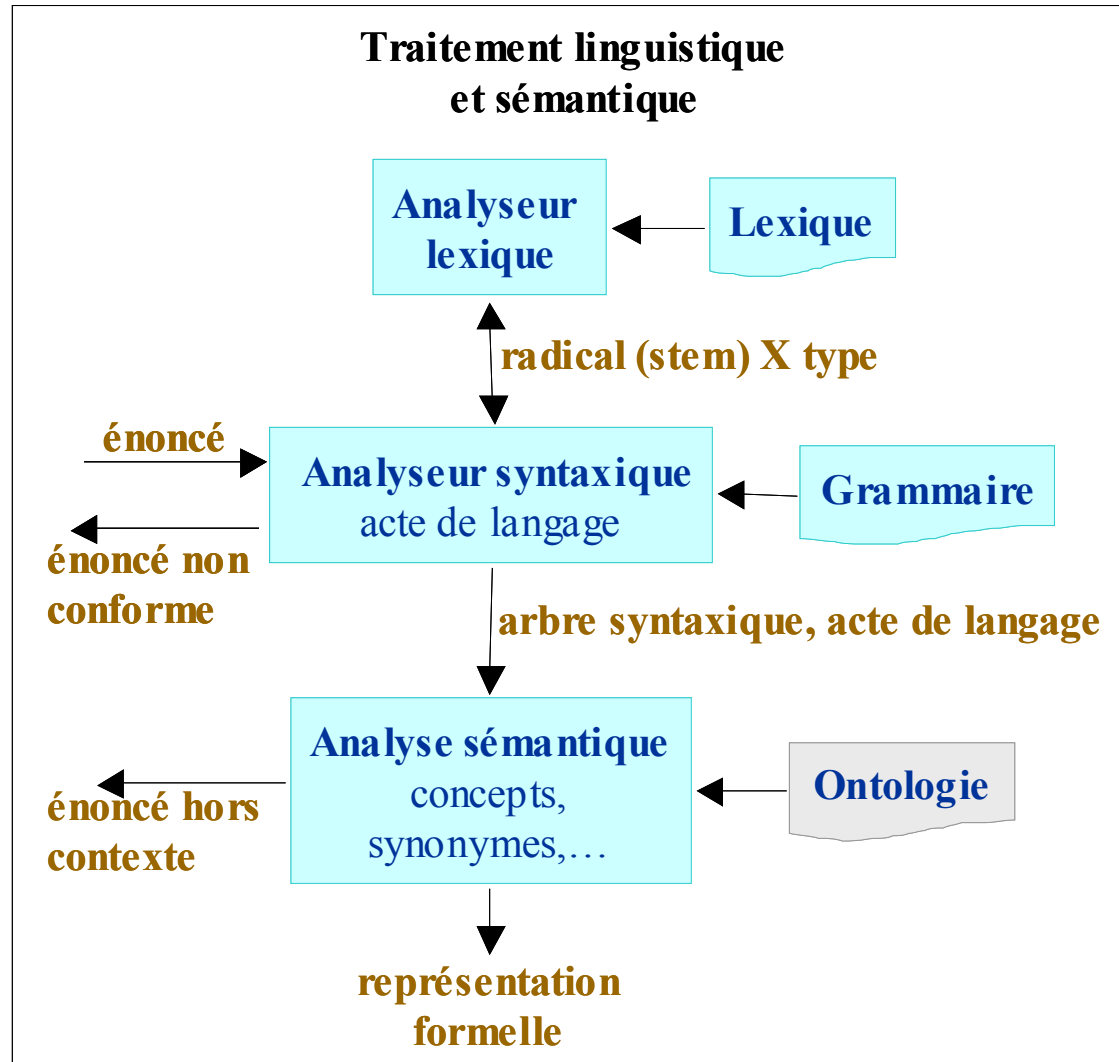
- **Définition :**
  - Une ICAI est le résultat de l'union d'un **mécanisme conversationnel en langage naturel parlé** et de la **gestion intelligente** de ce mécanisme, permettant le déroulement d'un **dialogue coopératif** et capable de gérer le déclenchement de plusieurs tâches à la demande de l'utilisateur, avec un minimum d'effort de la part de ce dernier.
- **Comment :**
  - Interface conversationnelle, basée sur :
    - la définition d'un système de dialogue fondé sur les actes de langage directifs (ordre, question et réponse) ;
    - l'adoption d'une stratégie coopérative pour le système de dialogue ;
    - l'ancrage sémantique à travers les ontologies ;
    - la séparation physique des connaissances de domaine et des tâches.
  - la mise en place d'une politique d'affichage d'informations, un premier pas vers une politique de présentation ;
  - une ontologie du domaine conçue pour la gestion de dialogues avec l'agent assistant.



# L'interface conversationnelle pour l'agent assistant

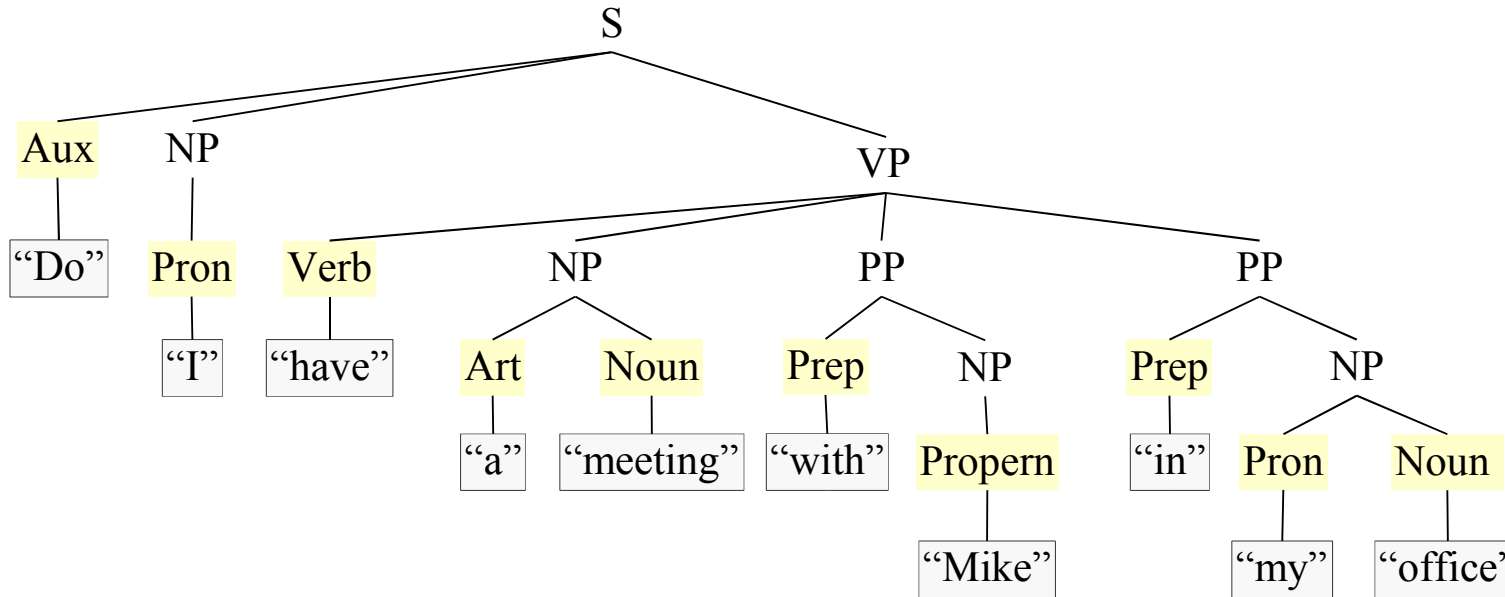


# Le traitement des entrées



# L'analyse syntaxique

USR: Do I have a meeting with Mike in my office?



VO

DN

Do I have a meeting [with Mike] [in my office]

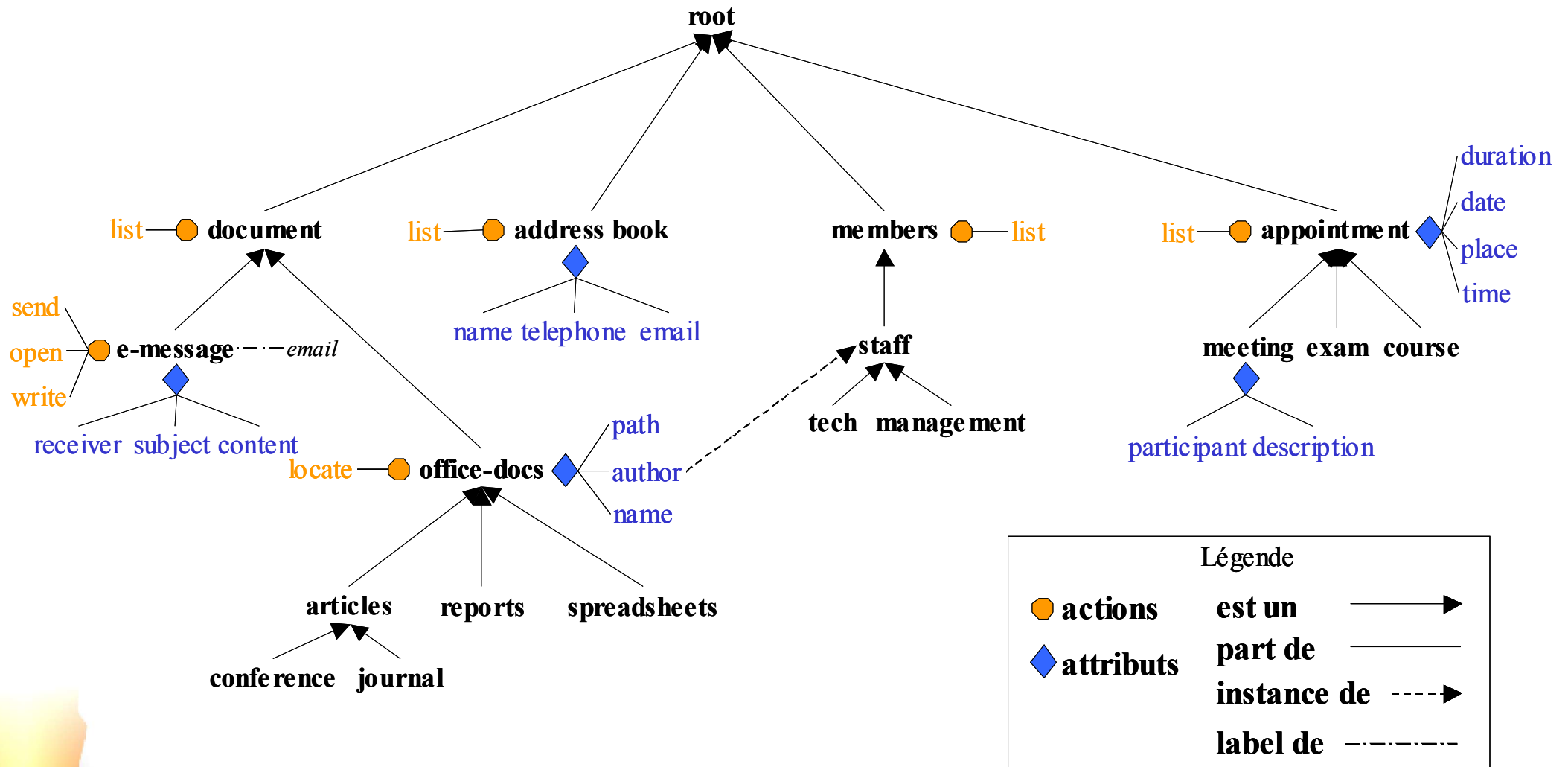
DN



## L'interaction avec l'utilisateur : Actes de langage

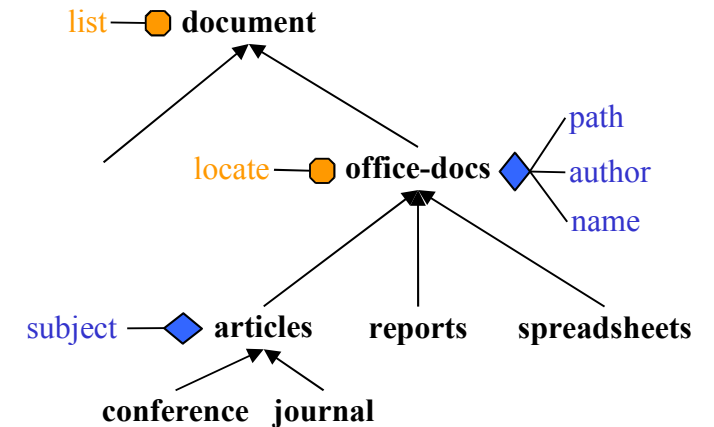
- Une fois interprétées les entrées, « quoi faire » ?
- Chaque énoncé est compris comme une action voulue par son émetteur
- Les actes illocutoires :
  - Directif (requête, suggestion, invitation, avertissement)
- En classant les énoncés de l'utilisateur on peut :
  - Guider l'interaction (déclencher des tâches)
  - Comprendre son état émotionnel
  - Utiliser le même type de « discours » (parler la « langue » de l'utilisateur)

# Ontologie du domaine (extrait)



# Construction de l'ontologie du domaine

- **Concepts :**
  - l'utilisation des relations d'hyponymie/hyperonymie (*is-a*) et de méronymie (*has-a*)
- **Les attributs :**
  - pour chaque attribut : type, liste de mots synonymes, cardinalité et la restriction de domaine appropriée (*time, space, people* et *general*)
- **Les actions :**
  - actions applicables à chaque concept doivent être explicitées



- **Les tokens :**
  - liste de mots synonymes pour chaque concept
- **Les instance :**
  - instantiation multiple

# Construction de la représentation formelle

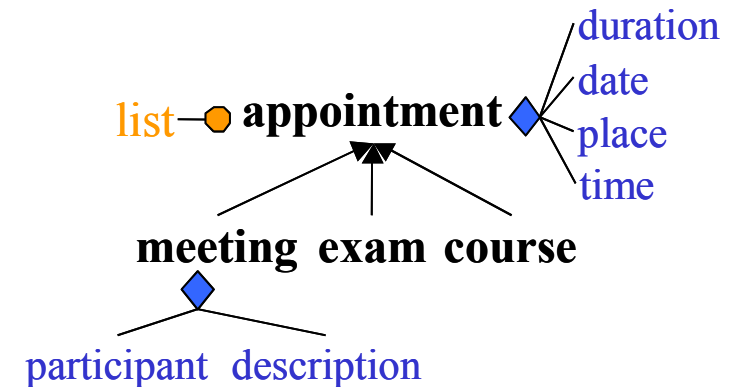
```

<formula> ::= '(' <action> 1*<ontology-components> 1*')'
<action> ::= <token>
<ontology-components> ::= '(' 0*<attribute-name> 0* '(' <concept> ') '
<attribute-name> ::= <token>
<concept> ::= <concept-name> 0*<attribute-list>
<concept-name> ::= <token>
<attribute-list> ::= '(' ':' <attribute-name> <attribute-value> ') '
<attribute-value> ::= <token> | 'nil'
<token> ::= string
    
```

VO → DN

Do I have a meeting [with Mike] [in my office]

DN



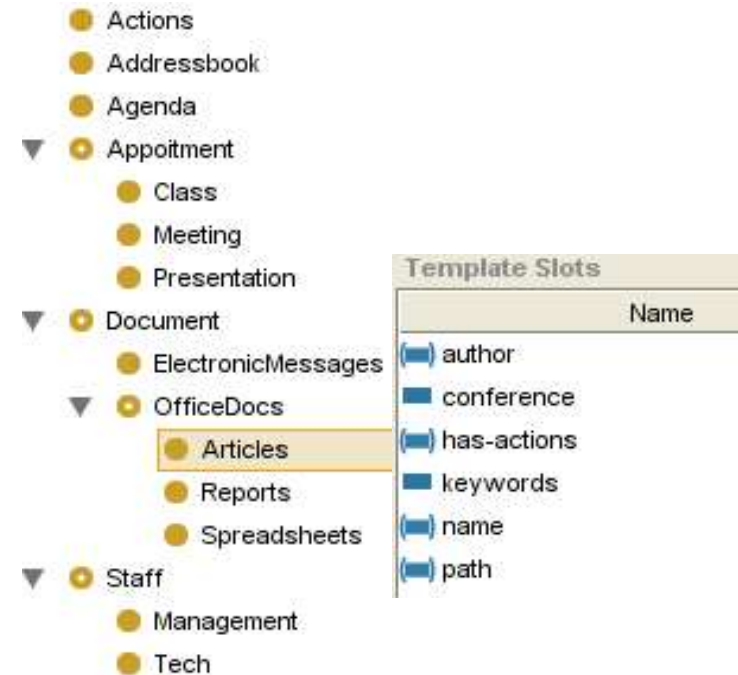
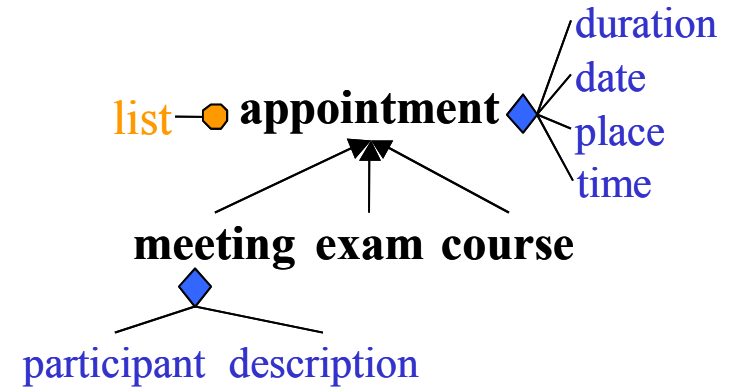
```
(list (Meeting (:participant "Mike") (:place "office")))
```

# Ancrage sémantique à l'aide des ontologies

USER: Do I have a meeting with Mike in my office tomorrow?

```
(Meeting (:date "27-03-2005")
(:time nil)
(:place "office")
(:duration nil)
(:participant "Mike")
(:description nil))
```

```
(list (Meeting (:place "office")
(:participant "Mike")
(:date "27-03-2005"))) )
```





## Contexte et référence

USR (1): What is the email address of Mary?

```
(list (address (AddressBook (:name "Mary"))))
```

AP (2) : The email is: mary@company.com.

USR (3): Send an email to her.

```
(E-Message (:receiver "Mary")
(:subject "nil")
(:content "nil"))
```

USR (1): What is the starting time of the meeting with Mike?

```
(Meeting (:date "17-05-2005")
```

AP (2) : Starting time is: 14:00.

```
(:time "14-00")
```

USR (3): Where is it planned to be?

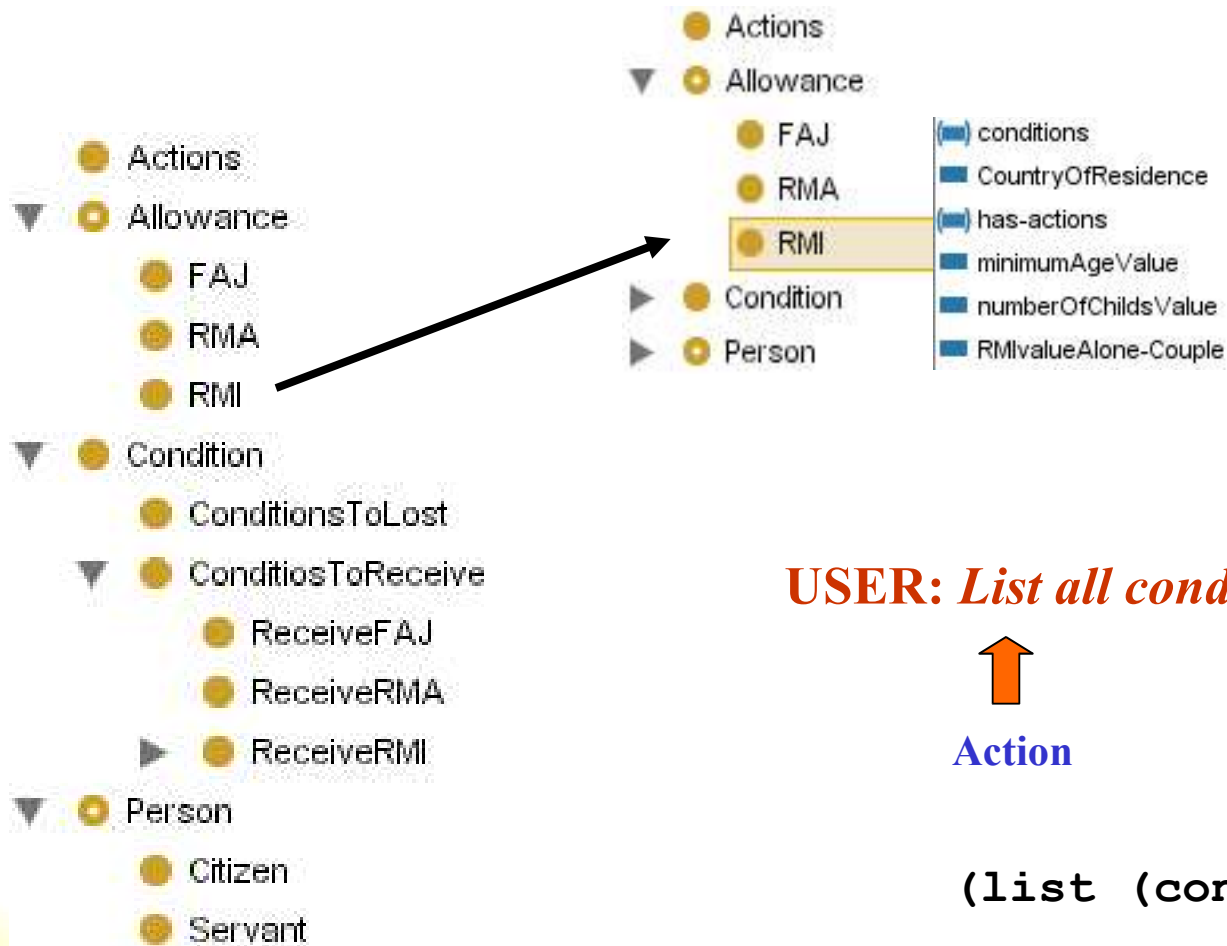
```
(:place "office")
```

```
(:duration "1-00")
```

```
(:participant "Mike")
```

```
(:description "nil"))
```

# Le modèle des tâches



```

- <rdf:RDF xml:base="http://www.owl-ontologies.com/unnamed.owl">
- <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
</owl:Ontology>
- <owl:Class rdf:ID="Person">
  <protege:abstract>true</protege:abstract>
</owl:Class>
- <owl:Class rdf:ID="Citizen">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
- <owl:Class rdf:ID="Servant">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
- <owl:Class rdf:ID="RMI">
- <rdfs:subClassOf>
  <owl:Class rdf:ID="Allowance"/>
</rdfs:subClassOf>
</owl:Class>
  
```

**USER: List all conditions to receive an RMI**

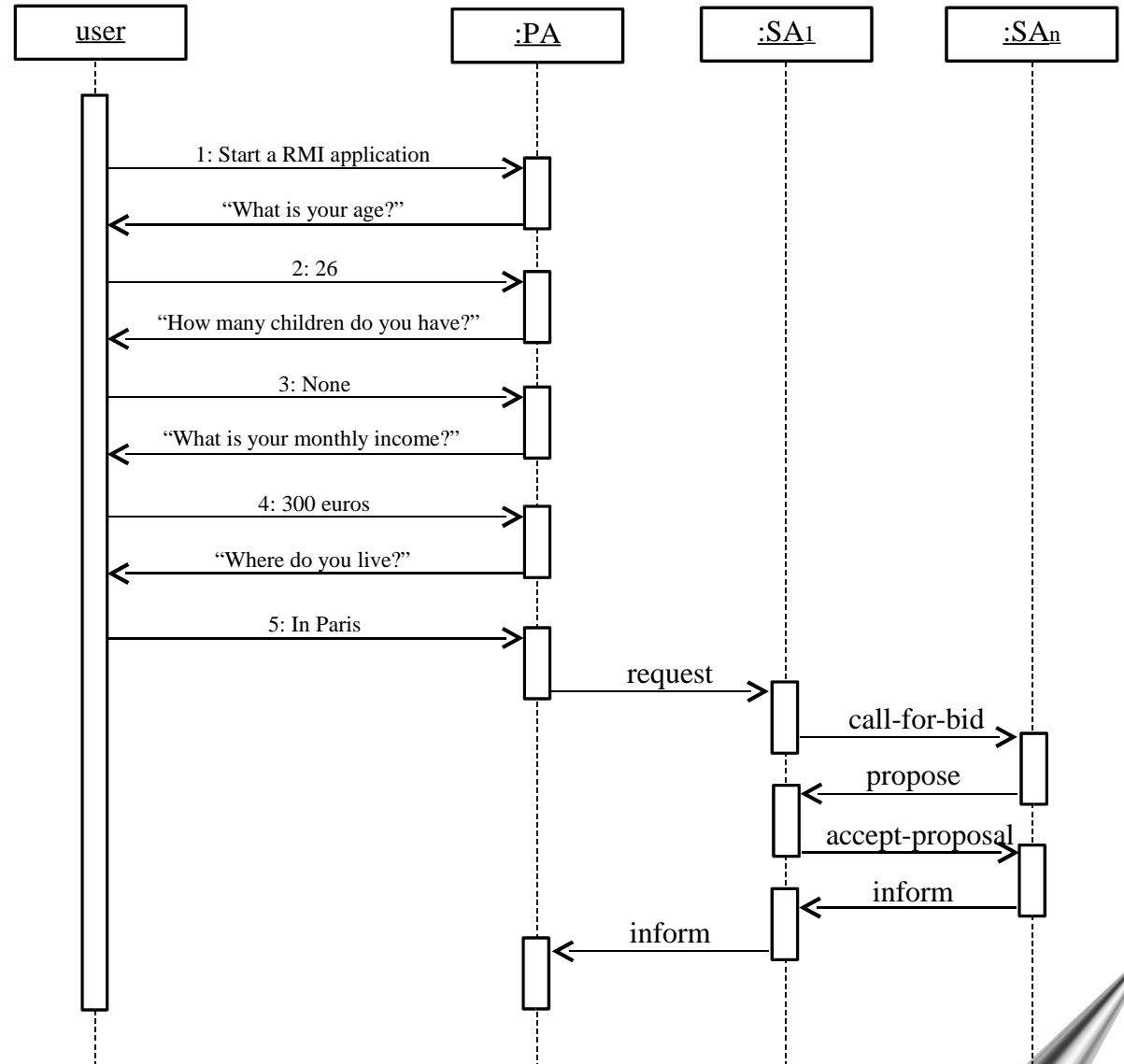
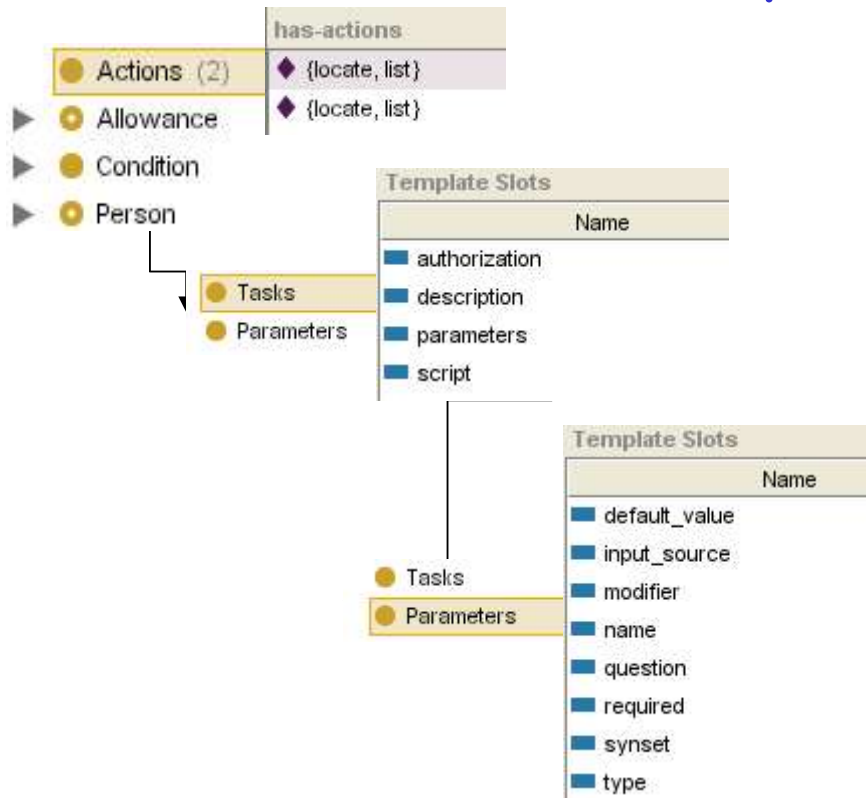


**(list (conditions (Allowance)))**

**(Allowance (:conditions nil) ... )**



# Remplissage des paramètres



**USER: Start an RMI application**

(start (RMI))

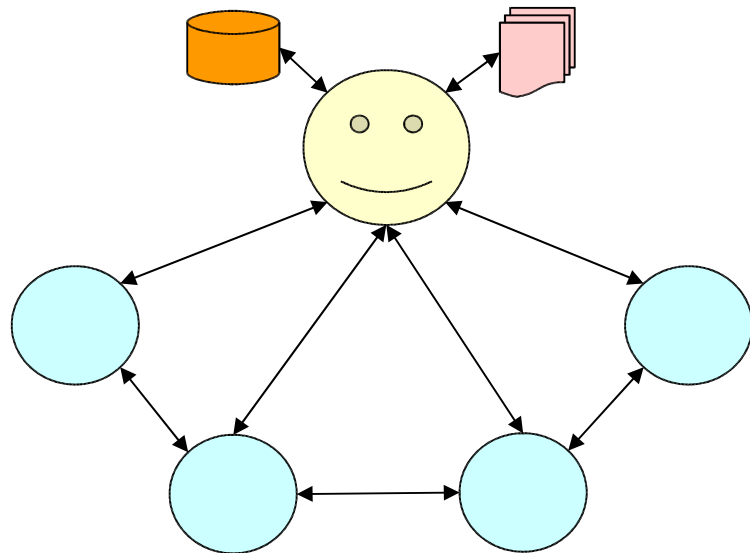
```

(RMI (:countryOfResidence nil)
 (:minimumAgeValue nil)
 (:numberOfChildrenValue nil)
 (:RMIValue nil)
 (:conditions nil))
    
```

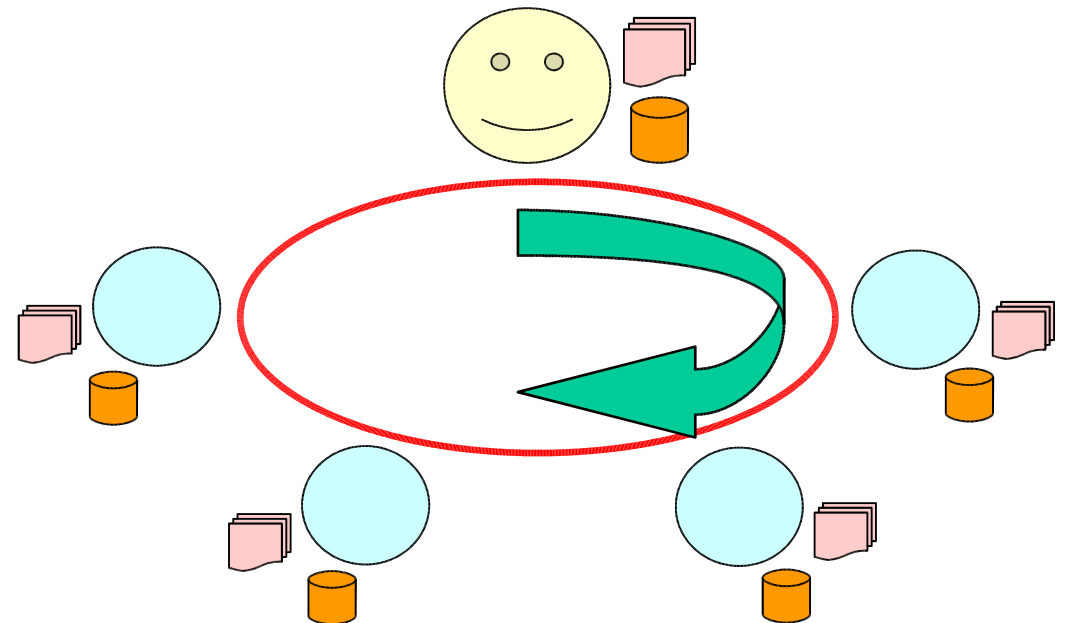



# Distribution des tâches

« Assistant Facilitateur »




« Centre de Services »



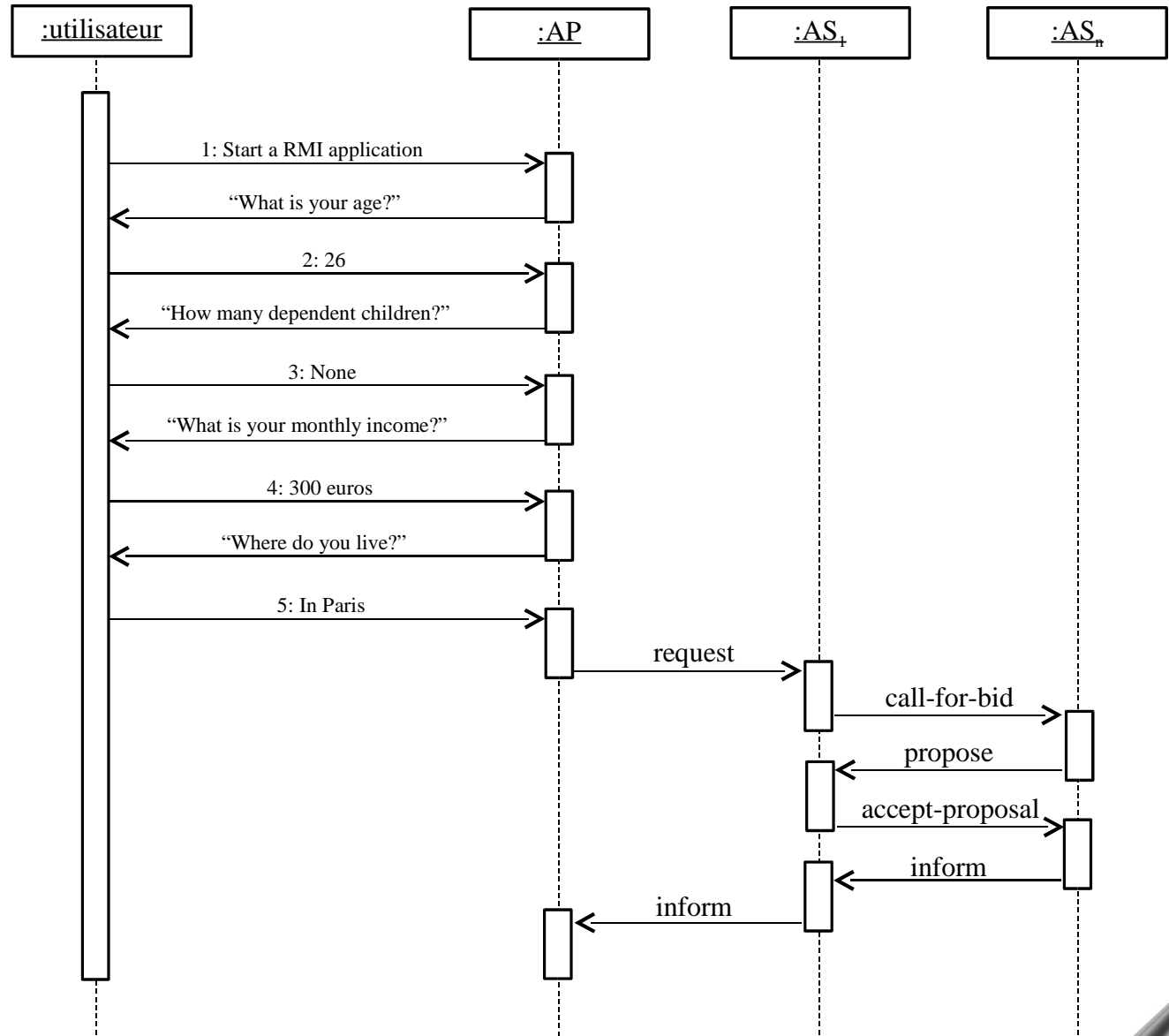
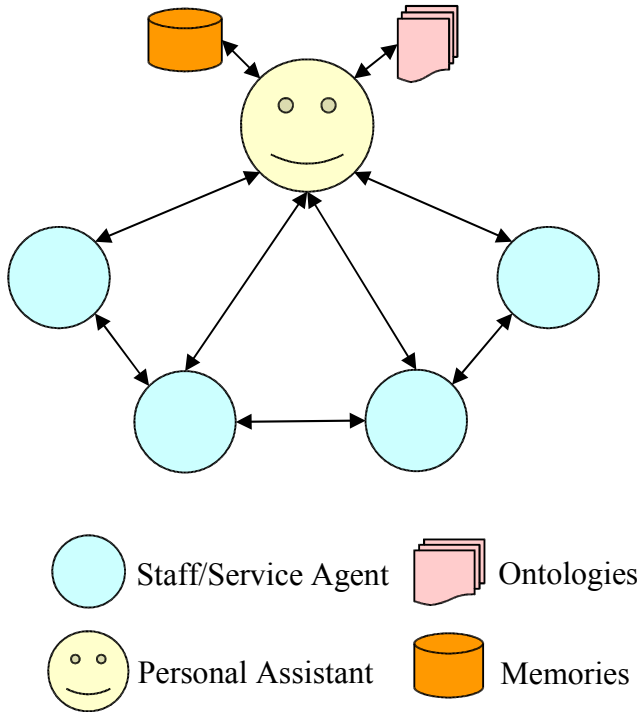
 Agent de service

 Ontologies

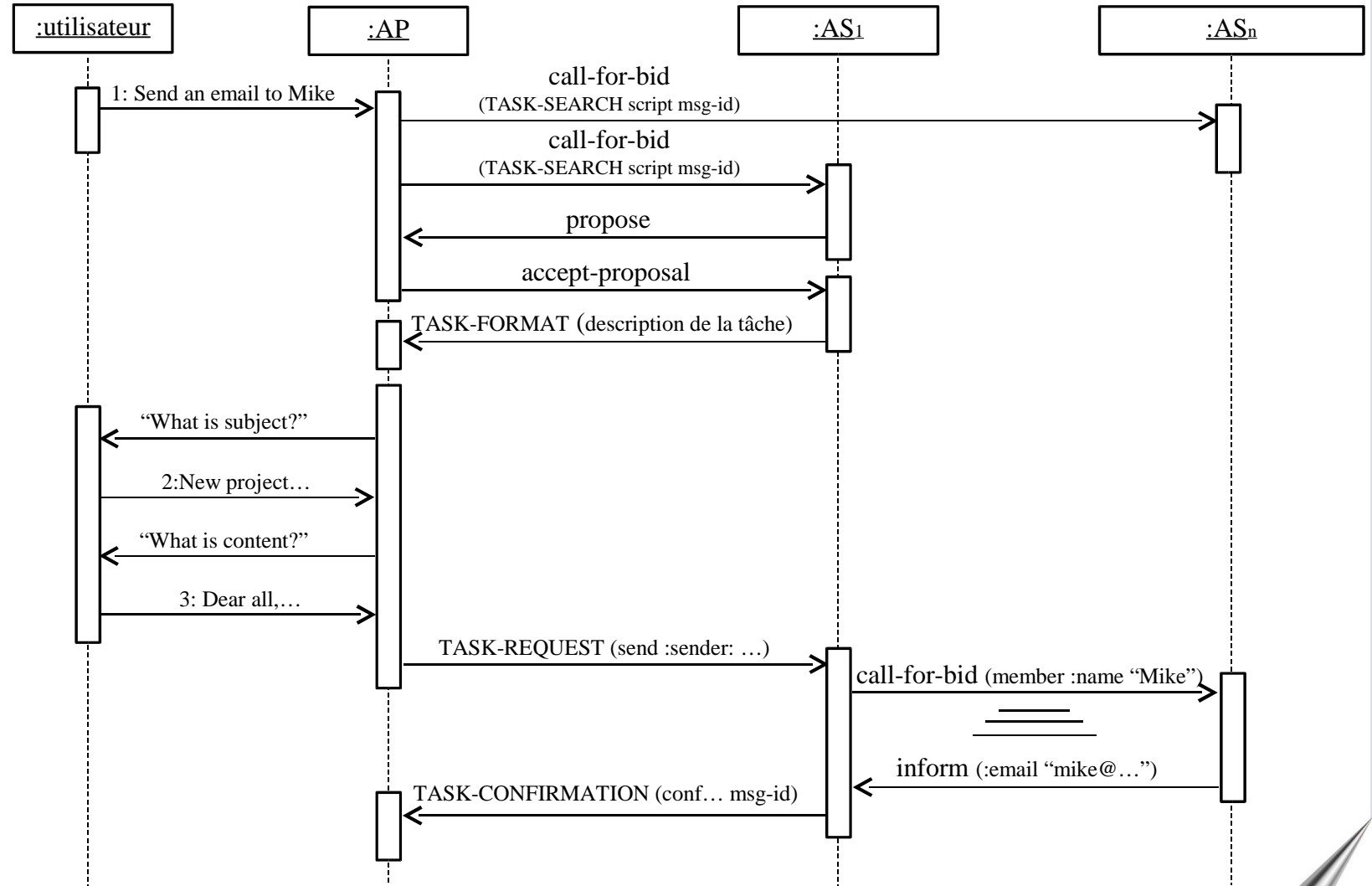
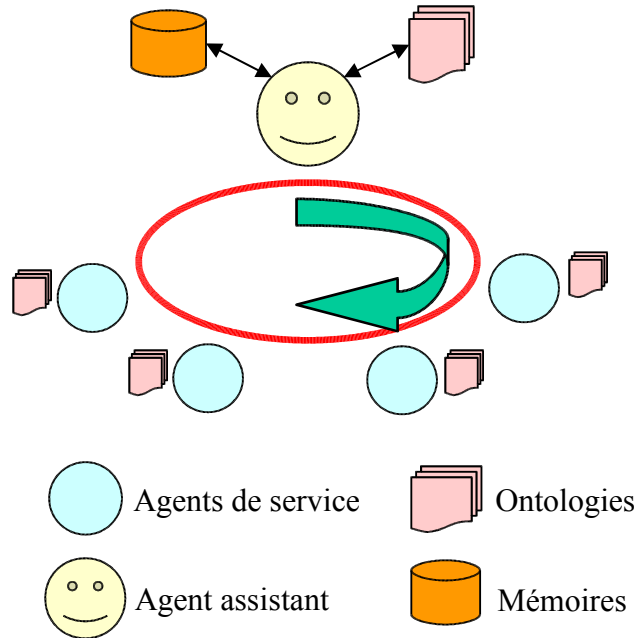
 Agent assistant

 Mémoires

# Interactions - Assistant Facilitateur



# Interactions - Centre de services



# Langage de contenu - gestion des agents

**INFO content**

**REQUEST-INFO question-description msg-id**

**INFO-REQUESTED msg-id content**

**TASK-SEARCH script msg-id**

**TASK-FORMAT task-description msg-id**

**TASK-REQUEST script msg-id**

**TASK-RESULT msg-id response content**

**TASK-CONFIRMATION msg-id content**



# L'évaluation de l'interface conversationnelle

Données relevées	Testeur 1	Testeur 2	Testeur 3	Testeur 4
Durée de la session en minutes	9	24	13	16
Nombre d'échanges (turn-takings)	46	78	64	82
Nombre d'entrées hors sujets acceptées	0	0	0	1
Nombre de tâches déclenchées	7	8	8	9
Temps moyen pour le déclenchement d'une tâche (en secondes)	13.86	19.86	15.14	20.67
Nombre d'offres d'aide spontanée	0	1	1	2
Nombre de mots non reconnues	4	5	4	7
Nombre moyen de mots dits par entrée	4.63	3.83	4.05	3.45
Nombre d'entrées refusées	4	9	5	11



## Conclusions et travaux futures

- Deux prototypes réalisés
  - l'architecture proposée se montre adéquate
  - l'interface est prévisible
  - l'évolution du modèle générique de l'agent assistant
- Les travaux futures et les voies de recherche
  - amélioration des prototypes
  - application réelle
  - conception et mise en place d'une politique de présentation
  - étude de l'impact de la mobilité de l'utilisateur
  - approfondissement de l'étude du rôle des ontologies dans l'interprétation et dans la personnalisation de l'interaction entre l'utilisateur et son agent assistant

## Les travaux en cours - Groupe Agents

- **Gilson SATO**
  - Modélisation et implémentation d'un agent numérique pour le support des coordinateurs de Communautés de pratique distribuées qui participent dans le développement de produit
- **Adriana VIVACQUA**
  - Modélisation et implémentation numériques pour informer les personnes d'opportunités de collaboration
- **Fathia BETTAHAR**
  - Expressivité des formalismes de représentation des ontologies : application à l'orchestration dynamique de services dans le cadre d'e-Gouvernement
- **Kejia CHEN**
  - Structuration de la mémoire d'un agent assistant : applications aux EIAH
- **Emerson PARAISO**
  - Interface conversationnelle pour une aide intelligente

Fin

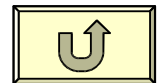
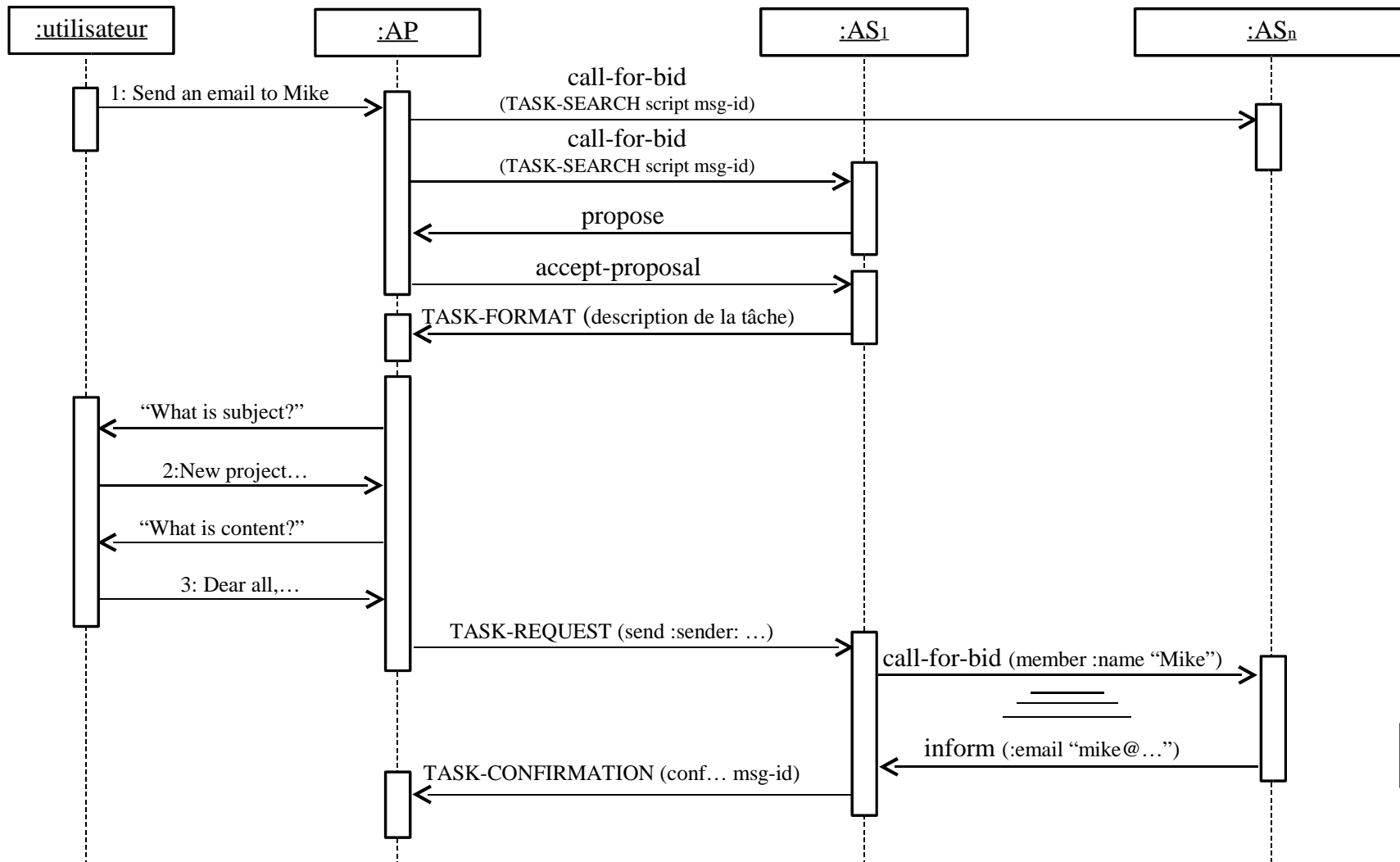
Merci.

[emerson.paraíso@utc.fr](mailto:emerson.paraíso@utc.fr)

# Task Search

- call-for-bid:

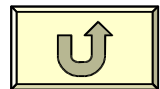
(TASK-SEARCH (list (address (AddressBook (:name "Mike"))))) 8)



# Task Format

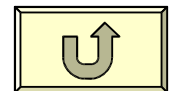
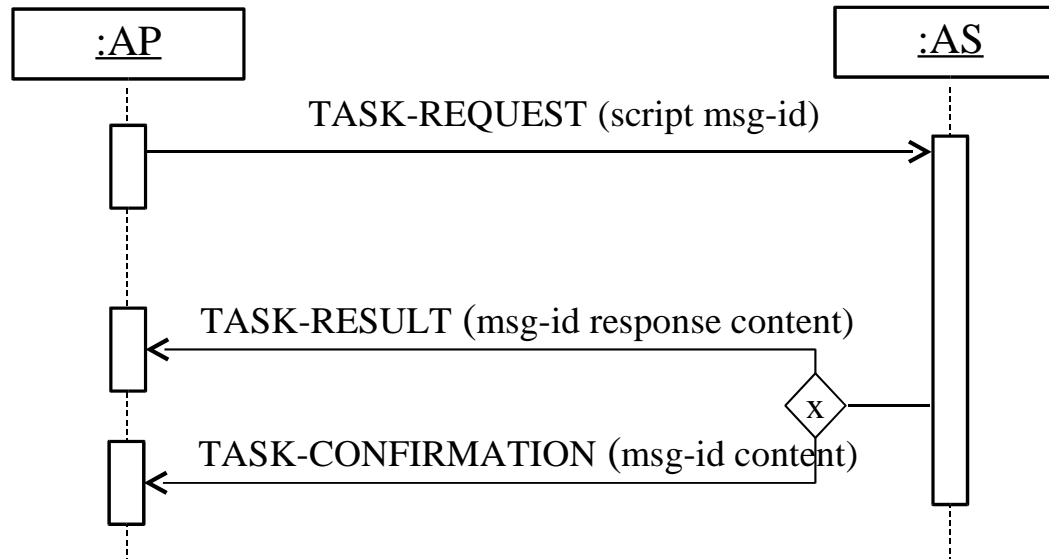
```
(TASK-FORMAT (<Task>
  <prm prm_id="1">
    <name>RECEIVER</name>
    <synset>receiver to</synset>
    <question>Who is the receiver</question>
    <type>propern</type>
    <required>>true</required>
    <input_source>vocal</input_source>
    <default_value>none</default_value>
  </prm>
  <prm prm_id="2">
    <name>SUBJECT</name>
    <synset>subject about</synset>
    <question>What is the subject</question>
    <type>text</type>
    <required>>true</required>
    <input_source>vocal</input_source>
    <default_value>none</default_value>
  </prm>
  <prm prm_id="3">
    <name>CONTENT</name>
    <synset>content text</synset>
    <question>What is the content</question>
    <type>text</type>
    <required>>true</required>
    <input_source>keyboard</input_source>
    <default_value>none</default_value>
  </prm>
  <script>@SENDEMAIL</script>
  <authorization>>true</authorization>
</Task>)
```

105)



# Task Request

```
(TASK-REQUEST (@SENDEMAIL
  (:receiver "Marc")
  (:subject "Setting a meeting...")
  (:content "Dear Marc, Mrs. Mary Smith is working ...")) 2)
```



# Mémoires de l'agent

- Historique des instances des concepts
- Mémoire à court terme
  - liste des tâches en exécution
- Mémoire à moyen terme
  - liste des tâches déjà exécutées
- Mémoire à long terme
  - *log* (stockage physique) des tâches exécutées au cours d'une session de dialogue
- Mémoire de conversation
  - liste des énoncés parlés par l'utilisateur et par le système, pour chaque tâche.