

**TALISMAN**  
**UN SYSTEME MULTI-AGENTS GOUVERNE PAR DES LOIS LINGUISTIQUES**  
**POUR LE TRAITEMENT DE LA LANGUE NATURELLE**

MARIE-HELENE STEFANINI, ALAIN BERRENDONNER\*, GENEVIEVE LALLICH, FLAVIO OQUENDO  
e-mails : {stefanini, lallich, oquendo} @ criss.fr  
Fax : (+33) 76-82-56-75. Telex : UNISOG 980910. Tel : (+33) 76-82-54-18  
CRISS- Université Pierre Mendès France  
BP 47- 38040 GRENOBLE CEDEX 9, FRANCE  
\* Séminaire de Linguistique Française  
1700 Université de FRIBOURG, SUISSE

**ABSTRACT**

Natural language processing raises the problem of ambiguities and multiple solutions which follow from them. The knowledge gained when using the morpho-syntactic analyser CRISTAL showed how necessary it was to overcome this issue. The architecture with sequential levels, in which each module corresponds to a linguistic level (pretreatments, morphology, syntax, semantics) has shown its limits. A sequential architecture does not allow a real exchange between different modules. This leads to the non availability of the linguistic information for the reduction of ambiguities, at the moment they are needed.

The necessity for cooperation between different modules has lead us to envisage a new architecture which stems from the techniques of distributed artificial intelligence.

The paper presents this new architecture which based on distributed artificial intelligence techniques treats the inherent problems of natural language processing. One of the originalities resides in the distributed treatment of sentence analysis (as apposed to a classic sequential treatment) and in the introduction of linguistic laws which allow the management of the communication between agents, without appealing to a central control. The Talisman system is an environment which integrates linguistic tools where different agents can bring into use different methods such as symbolic and/or statistic ones.

The Talisman system contributes to the following points in the linguistic domain :

- the restriction of ambiguities by agent cooperation,
- rendering structures less complex by using local grammar rules,
- the treatment of uncertain information.

It can :

- function with partial analyses at different classical levels of analysis,
- change strategies according to the applications or the corpus used,
- use linguistic laws which are easily modifiable.

At the implementation level, the system brings openness to the modification of dictionaries, grammars and strategies of analysis, and the necessary mechanisms for the integration of new modules.

Talisman is a linguistic environment based on the most recent techniques used in software engineering

environments. It provides mechanisms for data and control integration of linguistic tools.

This paper is organized as follows. After a short overview in section 2 about the problems of sequential architectures, we establish the main objectives of our work in section 3. We present, in section 4, the contribution of multi-agents systems "governed by laws". In section 5, we define the structure of an agent and of its society. The implementation is presented in section 6.

**Keywords**

Environment framework for linguistics tools, natural language, written french, distributed artificial intelligence, multi-agent systems, law governed systems, communication protocols.

**RESUME**

Nous présentons une architecture nouvelle issue des techniques de l'intelligence artificielle distribuée pour traiter des problèmes inhérents à la langue naturelle tels que les ambiguïtés. Une des originalités réside dans la conduite distribuée de l'analyse de la phrase (à l'opposé d'un traitement séquentiel classique) et dans l'apport des "lois" qui permettent de gérer la communication entre agents, sans faire appel à un contrôle centralisé. Le système Talisman, fondé sur le concept de système multi-agents gouverné par des lois, est un environnement d'intégration d'outils linguistiques où les différents agents du système peuvent mettre en œuvre des méthodes telles que des méthodes symboliques et/ou statistiques.

**Mots-clés**

Environnement d'intégration d'outils linguistiques, langue naturelle, français écrit, intelligence artificielle distribuée, systèmes multi-agents, systèmes gouvernés par des lois, protocole de communication.

# 1. INTRODUCTION

L'analyse automatique de la langue naturelle soulève le problème des ambiguïtés [Rady 1983] et des solutions multiples qui en découlent. L'expérience acquise avec l'utilisation de l'analyseur morpho-syntaxique CRISSTAL<sup>1</sup> [Lallich & al 1990] nous a montré combien il était nécessaire de maîtriser ce problème afin d'arriver à une solution juste. Face à cet impératif, l'architecture par niveaux dite "séquentielle", dans laquelle chaque composant correspond à un niveau linguistique (prétraitements, morphologie, syntaxe, sémantique) a montré ses limites [Sabah 1991]. En effet, la démarche linéaire privilégiée à chaque étape de l'analyse un module linguistique spécifique, or, une architecture séquentielle ne permet pas un réel échange entre les différents modules, ce qui entraîne la non-disponibilité, aux moments opportuns, d'informations linguistiques nécessaires à la réduction des ambiguïtés.

La nécessité d'une coopération entre les différents modules nous a amenés à envisager une nouvelle architecture issue des techniques de l'Intelligence Artificielle Distribuée.

Après un bref exposé, en section 2, des problèmes inhérents à l'architecture séquentielle, nous établissons les objectifs de notre travail dans la section 3. Nous présentons, en section 4, l'apport des systèmes multi-agents "gouvernés par des lois". En section 5, nous définissons la structure d'un agent et celle d'une société d'agents et nous exposons des exemples de coopération et de conflit entre agents. L'implantation de Talisman sera décrite en section 6.

## 2. LES LIMITES DE L'ARCHITECTURE SEQUENTIELLE

CRISSTAL, comme beaucoup de systèmes d'analyse de la langue naturelle, est construit selon une architecture séquentielle et utilise une description de la langue basée sur trois niveaux : morphologique, syntaxique et sémantique.

Le niveau morphologique comprend :

- les prétraitements : ils normalisent les caractères et découpent le texte en formes ;
- l'analyseur morphologique flexionnel : il attribue aux formes une ou plusieurs catégorisations ;
- la levée d'ambiguïtés basée sur des filtres de Markov : elle réduit les catégorisations possibles ;

Le niveau syntaxique opère sur les résultats morphologiques et comprend :

- un regroupement des morphèmes discontinus pour se ramener à un modèle hors contexte ;
- la décomposition des phrases en propositions ; unités syntaxiques complètes et plus simples à analyser ;
- l'analyse syntaxique de la proposition exécutée par un système expert ;

Le niveau sémantique construit la structure logique à partir de la structure syntaxique. Il comprend :

- une transduction de l'arborescence : passage du formalisme de constituants à celui de grammaire fonctionnelle ;

- une résolution d'anaphores intra-phrastiques, en créant une relation de liage entre un terme anaphorique et son référent ;
- un marquage des opérations d'énonciation afin de fournir des informations sur la nature des syntagmes nominaux.

L'architecture séquentielle présente des inconvénients. Elle génère des solutions parasites. La multiplicité de ces solutions croît aussi avec l'utilisation d'une grammaire générale.

### 2.1. Les solutions parasites et leur combinatoire

Chaque niveau sécrète des ambiguïtés qu'il ne peut résoudre, faute d'accès à des informations globales. De plus, une description par niveaux correspond à une série d'interprétations successives indépendantes les unes des autres. Or, l'information dont on dispose à un moment donné peut concerner plusieurs niveaux d'analyse. Par exemple :

*le (α) lui (β) poste (γ) - il ce matin ?*  
les ambiguïtés : α préverbal / déterminant  
β nom / préverbal  
γ verbe / nom

L'information morphographique du tiret suivi d'un pronom sujet permet de faire des inférences indirectes qui lèvent les ambiguïtés :

- le tiret implique (*poste* ∈ verbe)
- (*poste* ∈ verbe) implique (*lui* ∈ préverbal)
- (*lui* ∈ préverbal) implique (*le* ∈ préverbal).

Ces informations sont utiles à divers niveaux de l'analyse, en particulier en syntaxe, pour construire le sous-arbre correspondant à un syntagme verbal. De plus, la propagation des ambiguïtés d'un module vers un autre provoque une explosion combinatoire. Par exemple, on a 288 interprétations hors contexte possibles pour la phrase :

*La (φ) machine (γ) garde (γ) en (ν)  
mémoire la (φ) trace (γ) des  
programmes (γ).*

les ambiguïtés : φ : préverbal / nom / déterminant  
γ : verbe / nom  
ν : préposition / préverbal

Une meilleure stratégie serait d'obtenir, en cours de traitement morphologique, des indications de la syntaxe sur la structure des verbes *machiner*, *garder* et *tracer*.

### 2.2. La complexité d'une grammaire générale

La complexité résulte de l'utilisation d'une grammaire générale qui essaie de prévoir toutes les possibilités de la langue en les mettant sur le même plan. En effet, la langue naturelle comprend des habitudes de parole qui n'ont pas toutes le statut de règles strictes, mais souvent de simples préférences et ne concernent pas globalement toutes les constructions, mais souvent, localement un domaine limité de constructions spécifiques.

Par exemple, pour la grammaire du syntagme nominal, on a besoin de reconnaître quelques cas de construction telle que : N" -> N" N"

Cette règle permet de construire des syntagmes nominaux résultats de la juxtaposition de deux syntagmes nominaux.

ex : *Alger la blanche, le docteur X...*

<sup>1</sup> utilisé dans le projet ESPRIT MMI2 [MMI2 89].

Cette règle interfère constamment avec d'autres règles et produit des solutions parasites que l'on pourrait supprimer en divisant la grammaire en partie standard et partie spécifique. (cf 3.2)

### 3. LES OBJECTIFS DE TALISMAN

Notre but est de définir une architecture qui compose différentes phases d'analyse et qui, en fonction des problèmes de reconnaissance que posent les structures linguistiques rencontrées, soit capable d'ordonner convenablement les divers sous-ensembles de règles et de les activer au bon moment. Les améliorations attendues sont la restriction des ambiguïtés par coopération de modules, l'activation de grammaires locales pour décomplexifier les structures et le traitement des informations incertaines.

#### 3.1. La restriction des ambiguïtés par coopération de modules

Pour résoudre une ambiguïté morphologique, il est souvent nécessaire d'avoir préalablement effectué une analyse syntaxique partielle du secteur du texte qui la contient [Berrendonner 1990]. Par exemple :

*Cet article est le fruit d'une réflexion  
menée au CRISS sur l'analyse (γ)  
automatique (ζ) des langues.*

les ambiguïtés    γ :    verbe / nom  
                          ζ :    nom / adjectif

Au niveau morphologique, le choix entre nom ou adjectif pour *automatique* dépend d'une désambiguïté de *analyse* au niveau syntaxique.

De même, pour résoudre un problème syntaxique, il est souvent nécessaire d'en avoir résolu un autre. De plus, ces dépendances syntaxiques sont très irrégulières et varient beaucoup d'une phrase à l'autre.

En syntaxe, le rattachement des syntagmes prépositionnels dans l'arbre d'analyse est souvent problématique. Pour éviter de contruire *le CRISS sur l'analyse*, des stratégies sur la longueur des compléments (*au CRISS, sur l'analyse automatique...*), ou des stratégies probabilistes sur la dépendance régulière à droite peuvent être mises en œuvre.

#### 3.2. L'activation de grammaires locales

Les grammaires locales [Berrendonner 92] se caractérisent par leur moment d'intervention dans l'analyse, le contexte sur lequel elles opèrent, les informations linguistiques qu'elles produisent ainsi que par la propagation de ces informations aux niveaux ultérieurs. Contrairement aux grammaires générales, elles s'appliquent sur une configuration précise et fournissent une information "sûre" ou très probable. Elles sont utilisées à différentes fins qui peuvent se conjuguer :

- soit ramener les cas particuliers au cas général qui sera alors traité par une grammaire générale. Ce sont alors des grammaires de transformations.

exemple1 : la normalisation de structure d'une phrase interrogative pour la ramener à une forme assertive,  
exemple2 : la normalisation d'une forme verbale "composée" pour la ramener à une forme verbale unisegmentale ordinaire (regroupement de constituants discontinus).

- soit prédire de l'information "sûre" à l'usage des niveaux ultérieurs. Par exemple : la réécriture de la séquence *du en de le* doit s'accompagner de l'interprétation "*le, article*" car elle introduit toujours un syntagme nominal. Elle se distingue ainsi de la séquence *de le* où le doit être interprété comme un pronom introduisant un syntagme verbal à l'infinitif.

#### 3.3. Le traitement des informations incertaines

Les séquences textuelles contiennent tout à la fois des indices "sûrs" sur lesquels on peut fonder des prédictions et des indices porteurs d'informations diverses (*que, ne*) sur lesquels on ne peut avoir que des présomptions. Contrairement à l'information sûre qui peut être donnée ou calculée, l'information incertaine relève d'heuristiques fondées sur la langue ou sur le corpus : *ne* est le plus souvent annonceur de négation, *que* d'une subordonnée, mais *ne...que* est une négation restrictive. c'est-à-dire un opérateur sur un syntagme verbal qui ne doit être traité ni par le modèle de la négation, ni par celui de la subordination.

Les informations plausibles ne sont pas exploitées par une grammaire générale. Par exemple, les préférences structurales des sujets parlants :

- rien n'empêche d'employer une relative restrictive dans un syntagme nominal concernant un possessif :

ex: *ma cravate, qui est réservée pour les cérémonies...*

- néanmoins, dans certains textes, statistiquement, la quasi totalité des relatives après un nom possessif [Poss N] ne sont pas restrictives, mais appositives.

- il y a donc une préférence, (non une règle contraignante), pour éviter des structures telles que :

{Poss N P restrictif}N (nom possessif suivi d'une proposition restrictive) et l'analyse peut en tenir compte, au moins pour pondérer les solutions possibles.

### 4. UN SYSTEME MULTI-AGENTS GOUVERNE PAR DES LOIS

A l'opposé des architectures séquentielles, l'approche de l'Intelligence Artificielle Distribuée [Bond & Gasser 1988] répartit l'intelligence dans des *agents*. Ceux-ci sont des entités autonomes capables d'agir rationnellement sur elles-mêmes et sur leur environnement en fonction de leurs observations, de l'état de leurs connaissances et de leurs interactions. Cette approche conduit à la réalisation de systèmes dits *multi-agents* [Gasser & Ferber 1991]. Ils sont définis comme étant une société d'agents autonomes en interaction. Ces systèmes comprennent deux principaux modes de communication :

- les architectures par partage d'informations appelées "tableaux noirs" [Newell & Simon 1972], [Nii 1986].
- les architectures fondées sur la communication entre agents [Gasser & al. 1987].

L'inconvénient des "tableaux noirs" vient du fait que leur dispositif de contrôle est complètement centralisé, ce qui pénalise la distribution du raisonnement. Cette architecture a été appliquée en linguistique à la reconnaissance de la parole [Erman & al. 1980], et plus récemment, au traitement du français écrit : système

HELENE [Zweigenbaum 1989] et système CAMEL [Sabah 1990].

Par contre, les architectures basées sur la communication par messages entre agents permettent la distribution totale des connaissances, du contrôle et des résultats partiels. La distribution doit garantir l'autonomie partielle ou totale des agents. Chaque agent a une tâche spécifique et un protocole d'interaction avec les autres. Il dispose en outre de capacités de raisonnement et de décision qui le rendent autonome. Cependant, des problèmes de structuration de la communication tels que la synchronisation, la concurrence et le partage de ressources entre agents apparaissent lorsque les agents connaissent trop d'autonomie. Une société d'agents complètement autonomes n'est pas souhaitable pour le traitement de la langue naturelle, où le comportement des agents doit être guidé par des règles linguistiques.

Pour y remédier, nous proposons d'introduire le concept de "loi" tel qu'il a été défini par Minsky dans l'approche des systèmes dits *gouvernés par des lois* [Minsky 1988], [Minsky 1991]. Cette approche implique l'établissement d'un ensemble de règles communes à la société d'agents. Ce nouveau système informatique que nous avons appelé "système multi-agents gouverné par des lois" présente les caractéristiques suivantes :

#### *La modularité*

L'éclatement d'une tâche complexe en sous-tâches moins complexes entraîne une réduction de la complexité. Le système résultant est plus facile à développer, à tester et à maintenir.

Pour le traitement de la langue naturelle, nous pouvons ainsi conserver nos différents niveaux en les intégrant dans des agents experts en morphologie, en syntaxe, négation, etc.

#### *La robustesse*

Le système peut continuer à travailler même si la résolution d'une tâche échoue. Par exemple, si les agents en morphologie ou en syntaxe sont bloqués lors de la désambiguïsation d'une séquence de catégories, l'agent en statistique peut travailler indépendamment.

#### *La distribution du contrôle et des connaissances*

Elle permet de raisonner et d'acquérir ou de modifier des connaissances, au niveau local.

Par la distribution du contrôle, ces systèmes se prêtent plus naturellement à la mise en œuvre des grammaires locales que les systèmes linéaires. La distribution de la connaissance facilite la représentation et la modification de ces grammaires.

#### *Les expertises en concurrence*

Contrairement aux systèmes experts classiques, les agents peuvent avoir des expertises différentes, complémentaires ou contradictoires, sur un problème donné. Les conflits engendrés peuvent être résolus par des négociations entre agents.

Par exemple, pour l'analyse morphologique, deux agents pourront être mis en concurrence : l'un expert en morphologie dérivationnelle, l'autre en morphologie flexionnelle.

#### *La législation*

Les lois du système représentent un certain "code de conduite" pour tous les agents. Elles réglementent aussi certains conflits, ce qui permet, par exemple, de traiter les informations incertaines.

## 5. LE SYSTEME TALISMAN

Le système Talisman que nous avons développé est une implémentation des systèmes multi-agents gouvernés par des lois.

Nous présentons successivement les concepts d'agent et de société d'agents, définis dans le cadre du groupe PLEIAD<sup>2</sup>, et leur mode de communication particulier fondé sur des lois.

### 5.1. Description d'un agent

Un agent comprend deux parties, l'une interne représentant sa structure, l'autre externe qui traduit son comportement vis-à-vis de l'extérieur.

#### 5.1.1. Structure d'un agent

Sa structure interne comprend une partie statique et une partie dynamique.

##### **La partie statique**

Elle comporte trois composantes :

- *les connaissances*, c'est-à-dire l'expertise du domaine. Ces connaissances sont de deux types : les différentes ressources et la mémoire locale de l'agent qui contient ses résultats partiels et ses connaissances sur les autres agents,
- *les compétences* qui mettent en œuvre des algorithmes et des capacités de raisonnement,
- *les stratégies de contrôle* utilisées par l'agent pour prendre des décisions et mener des résolutions.

Par exemple, pour l'agent en syntaxe nommé SYNT :

- ses connaissances sont les dictionnaires et les grammaires qui jouent le rôle de serveurs d'information. Sa mémoire locale comporte des connaissances sur l'agent MORPH expert en morphologie,
- ses compétences sont représentées par un système expert qui supervise l'algorithme d'Earley et par différents automates dont un détecteur de circonstantsiels,
- ses stratégies font appel à des prédictions linguistiques fondées sur le comportement réactionnel de certains mots (les verbes, les déverbaux, les adjectifs, etc.).

##### **La partie dynamique**

Elle contient les mécanismes de raisonnement, de décision et de détection des incohérences, et les actions qu'il peut entreprendre sur l'environnement ou sur lui-même, par exemple, la mise à jour de sa mémoire locale.

#### 5.1.2. Comportement d'un agent

Les agents sont des entités dynamiques. Leurs actions sont perçues dans leur environnement par une production d'événements. Ceux-ci modifient l'environnement et les actes de communication. Les agents sont qualifiés de flexibles, adaptatifs et

<sup>2</sup>PLEIAD (Pôle et Lieu d'Echanges en Intelligence Artificielle Distribuée) regroupe les laboratoires grenoblois suivants : ARTEMIS, ICP, LIFIA, TIM3 (Université Joseph Fourier), CRISS (Université Pierre Mendès France) et LETI (CEA).

autonomes dans leurs actions et donc dans leurs prises de décision.

## 5.2. Les différents agents de Talisman

Dans Talisman, la société comprend actuellement huit agents linguistiques experts en :

- prétraitements (PRET) : délivre des informations morphographiques et morpho-syntaxiques utiles aux autres agents,
- morphologie (MORPH) : met en œuvre une expertise pour lever les ambiguïtés contextuelles, son but étant de donner la bonne catégorie à chaque mot,
- segmentation (SEGM) : renseigne sur le nombre de verbes attendus dans la phrase et décompose celle-ci en propositions relatives, complétives, etc.
- syntaxe (SYNT) : comprend des grammaires locales en interrogatives, infinitives, par exemple un agent local nommé NR traitant les restrictions négatives etc ; son but étant de donner la meilleure représentation syntaxique de la phrase,
- négation (NEGA) : repère les unités dans lesquelles s'effectue la négation,
- ellipses (ELLIP) : analyse les propositions elliptiques et les complète,
- coordination (COORD) : traite les cas simples de coordination,
- statistique (STAT) : utilisé sur des séquences qu'il est impossible de désambigüiser à l'aide d'heuristiques linguistiques.

Ces agents sont étudiés en détails dans [Stefanini & Oquendo 1992].

## 5.3. Description de la société d'agents

La société d'agents comprend elle aussi une structure interne et un comportement défini par les lois.

### 5.3.1. Structure d'une société

La structure interne de la société peut être hiérarchique ou hiérarchique. Les interactions de base entre agents comportent des messages d'informations, de requêtes, de confirmation, etc. La communication entre agents peut être directe ou indirecte. Dans notre application, la société ne comprend pas de contrôle centralisé et la communication entre agents est indirecte.

### La partie statique

Elle se compose aussi de trois entités :

- les connaissances de la société c'est-à-dire les agents et leurs liens,
- les compétences de la société, c'est-à-dire les résultats de la coopération entre agents,
- les stratégies de la société, c'est le résultat de l'interaction entre agents selon leurs connaissances et leurs compétences.

Les connaissances de Talisman sont représentées par les agents linguistiques. Ses compétences concernent la reconnaissance de la phrase. La société peut agir selon différentes stratégies. Exemple de stratégie : les résultats de l'agent en prétraitements forment un environnement à partir duquel différents agents travaillent en "parallèle" sans attendre que toutes les phases de l'analyse classique

soient terminées. Leur but global est la reconnaissance de la phrase.

### La partie dynamique

Par l'intermédiaire de leur protocole de communication, les agents deviennent coopératifs ou conflictuels, (cf. 5.4.2, 5.4.3) [Demazeau & Mueller 1990].

### 5.3.2. Comportement d'une société

Les lois permettent de définir le comportement de la société.

#### Définition des lois

Les lois régulent et modifient les échanges de messages entre les agents. Elle sont explicites<sup>3</sup> et modifiables.

On distingue ainsi deux étapes dans le passage d'un message :

- l'envoi du message. L'agent expéditeur E envoie un message M à l'agent destinataire D, en utilisant l'opération **send (E, M, D)** ;
- la livraison du message. Le message peut être transformé avant d'être envoyé aux agents destinataires D1 par l'opération **deliver (E, M1, D1)**. L'envoi de messages est transformé par la loi, de la manière suivante :
- le message reste inchangé par rapport à sa destination initiale :  
**send (E, M, D) -> deliver (E, M, D)**
- le message et/ou sa destination sont modifiés :  
**send (E, M, D) -> modify<sup>4</sup>(M, M1, D, D1) & deliver (E, M1, D1)**
- le message est bloqué :  
**send (E, M, D) -> fail**

Ces lois sont mises en œuvre par le serveur de messages.

## 5.4. Quelques exemples

Nous présentons quelques lois qui interviennent dans la société d'agents et nous les appliquons à un exemple de coopération puis de conflits entre agents.

### 5.4.1. Exemples de lois

#### Les lois linguistiques

Nous avons établi des lois pour les phrases assertives, interrogatives, impératives. Ces lois activent un agent spécialisé, en fonction d'indices prélevés dans le texte et de caractéristiques d'un problème d'analyse particulier. Par exemple, en présence d'un "?", rencontré par l'agent en prétraitements, une loi déclenche l'agent en syntaxe qui activera sa sous-grammaire propre aux interrogatives. Ainsi, lors de l'analyse de :

*est-ce que la machine garde en mémoire la trace des programmes ?*

Une loi d'attente intervient :

**send (SYNT, [interrogative,"est-ce...programmes"],...)** -> **deliver (SYNT, attente, ...)**

La loi bloque le travail des autres agents momentanément car il est inutile que l'agent expert en morphologie traite un début de phrase (*est-ce que*) qui peut générer inutilement certaines ambiguïtés alors

<sup>3</sup> Nous utiliserons la syntaxe de Prolog.

<sup>4</sup> La sémantique de ce prédicat est définie par l'utilisateur.

qu'un module de transformation s'attachera à rétablir une proposition assertive. Le traitement de la phrase assertive s'effectuera comme dans 5.4.2.

*Les lois sur la communication entre agents*

Elles évitent certains conflits inutiles.

-exemple1 : si aucun agent ne peut désambiguïser une séquence, il faut consulter l'agent expert en statistique STAT) :

send (\_X, [ambiguë, "chaîne"], ...) → deliver (\_X, "chaîne", STAT)

-exemple2 : lorsque l'agent MORPH peut résoudre lui-même une ambiguïté dans une séquence, il doit demander confirmation aux autres agents :

send (MORPH, [ ambiguë, "chaîne"], ...) → deliver (MORPH, [ambiguë, "chaîne"], [SYNT, SEGM...])

*Les lois de gestion des conflits*

On a un conflit lorsqu'un agent envoie le même message à deux autres agents et en reçoit deux réponses différentes. Pour gérer les conflits on peut donner une loi de priorité ou instaurer des négociations. La gestion des conflits est d'une complexité réelle pour la langue naturelle et requiert un approfondissement de nos travaux actuels.

#### 5.4.2. Exemples de coopération

La coopération s'établit lorsqu'un agent a besoin des services d'autres agents pour accomplir sa tâche. Ainsi, on peut considérer que l'agent en morphologie et l'agent en syntaxe coopèrent. En effet, pour accomplir sa tâche (levée des ambiguïtés morphologiques) l'agent en morphologie a besoin, face à une ambiguïté, de prédiction ou de confirmation de l'agent en syntaxe pour continuer l'analyse de la phrase. Reprenons l'exemple :

*La machine garde en mémoire la trace des programmes.*

La loi indique aux agents que la phrase est assertive.

Les agents s'activent en "parallèle" dans la phrase dès qu'ils reconnaissent un indicateur :

- L'agent en prétraitements reconnaît *des (de* : préposition + *les* : article) et envoie ses résultats à l'ensemble des agents,

- L'agent en syntaxe s'active sur les informations précédentes et construit un début de syntagme prépositionnel,

- L'agent expert en morphologie parcourt la phrase et élimine certaines ambiguïtés avec des règles contextuelles, par exemple pour *la machine* (déterminant + nom) et demande vérification à la syntaxe. *Mémoire* qui n'est pas ambigu (nom) permet de lever l'ambiguïté précédente (*en*) et d'obtenir une préposition. M demande aussi à la syntaxe des prédictions sur les verbes possibles (*garder* ou *tracer*),

- L'agent en segmentation ne repère pas d'indices pour décomposer la phrase (introduceurs de propositions, ponctuations) et en déduit qu'il n'existe qu'une seule proposition et donc qu'un seul verbe. Cette information est envoyée à la morphologie et à la syntaxe.

- La syntaxe consulte son dictionnaire des structures des verbes et choisit le verbe le plus adapté.

#### 5.4.3. Exemples de conflits

Des conflits peuvent s'instaurer entre agents en segmentation (SEGM) et en négation (NEGA) (*Ne...que* structure restrictive, cf 3.3).

*La machine ne garde en mémoire que la trace des programmes.*

L'objectif est d'inhiber les deux agents experts en négation (NEGA) et en segmentation (SEGM).

L'agent en morphologie envoie à tous les agents un message indiquant qu'il a trouvé une catégorie en négation :

send (MORPH, [négation, "ne"],...)

loi :

send (MORPH, [négation, "ne"],...) → deliver (MORPH, ["ne", négation], NEGA)

L'agent NEGA, après avoir reçu ce message, est en attente d'un *pas, plus, jamais...*, il n'obtient pas de réponse de l'agent MORPH et envoie le message suivant :

send (NEGA, [echec, "ne"],...)

La loi livre alors la négation à un agent NR, local à la syntaxe comportant une grammaire locale des restrictions négatives :

send (NEGA, [echec, "ne"],...) → deliver (NEGA, [deb-restrict, "ne"], NR)

L'agent NR reconnaît dans la phrase un "*que*" qu'il définit comme étant la partie finale de la restriction négative et en informe les autres :

( $\alpha$ ) agent NR : send (NR, [fin\_restrict, "que"],...)

L'agent en segmentation repère aussi l'indicateur *que* qui représente pour lui un introduceur de proposition et envoie un message d'information aux autres agents, ainsi qu'un message de requête aux agents SYNT et MORPH puisqu'il attend un verbe entre l'introduceur de proposition et le point final :

( $\beta$ ) agent SEGM :

send (SEGM, [introduceur, "que"],...) → deliver (SEGM, [cherche-verbe, "la trace...programmes"], [SYNT, MORPH])

Or, les deux messages  $\alpha$  et  $\beta$  entraînent un conflit entre les agents SG et NR, qui sera géré par la loi :

send (\_, [ambiguë, "que"],...) → deliver (NEGA, [cherche-verbe, "la trace...programmes"], [SYNT, MORPH])

La syntaxe ne considère pas que *trace* est un verbe. La loi donne donc priorité à la restriction négative :

send (\_, [ambiguë, "que"],...) → deliver (\_, [fin\_restrict, "que"],...)

D'autres conflits peuvent apparaître entre l'agent en segmentation et l'agent en coordination car leurs indicateurs peuvent être communs.

Le problème est de savoir si la coordination se trouve à l'intérieur de la subordonnée ou à l'extérieur car la coordination opère sur n'importe quelles unités

syntactiques donc sur les propositions et à l'intérieur des propositions.

Exemple : *Elle ne met que du rouge le matin, et le soir, que du noir.*

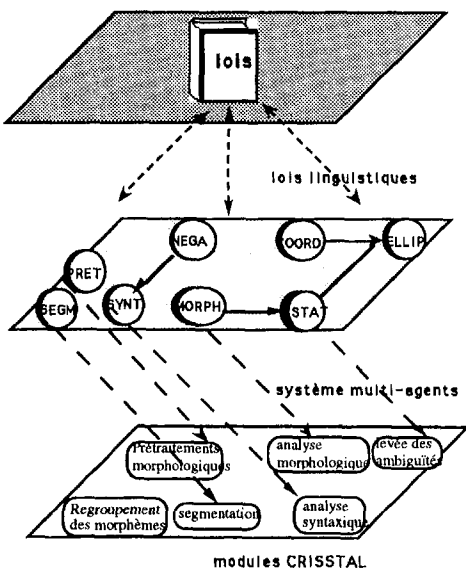
## 6. IMPLEMENTATION

L'architecture de Talisman peut être décomposée en :

- une partie constante où "structure d'accueil" représentée par un serveur de messages (S) qui gère la communication entre agents. Celle-ci est donc indirecte. Le serveur met en œuvre les mécanismes d'imposition des lois.

- une partie variable constituée des différents agents, de l'ensemble des lois et des modules du logiciel CRISSTAL qui implémentent certaines parties de l'expertise de l'agent.

Notre implémentation est à l'état de prototype. La structure d'accueil et les agents sont implémentés en Prolog Bim (version 3.0) sur station SUN4. Les modules du logiciel CRISSTAL sont écrits en C, et en Prolog. Ces modules sont encapsulés dans nos agents et peuvent être comparés à des agents serveurs qui leur rendent des services, sans pour autant être autonomes.



## 7. CONCLUSION

Nous avons présenté une nouvelle architecture informatique basée sur le concept de système multi-agents gouverné par des lois. L'implémentation qui en est faite, Talisman, apporte au domaine linguistique une avancée sur trois points : la restriction des ambiguïtés par coopération de modules, la limitation des solutions parasites par l'activation de grammaires locales, le traitement des informations incertaines.

Notre système peut fonctionner avec des analyses partielles à différents niveaux classiques d'analyse,

changer de stratégies selon les applications ou les corpus employés et utiliser des lois linguistiques facilement modifiables.

Sur le plan de l'implémentation informatique, il apporte la flexibilité indispensable à la modification des dictionnaires, des grammaires, des stratégies d'analyse et l'ouverture nécessaire à l'intégration de nouveaux modules.

Talisman est un environnement linguistique, issu des techniques les plus récentes pour la construction d'ateliers de génie logiciel [Oquendo et al. 1990]. Il fournit des mécanismes pour l'intégration des outils linguistiques par les données et par le contrôle.

Les travaux ultérieurs de Talisman vont concerner la robustesse de l'implémentation informatique, l'ajout de nouveaux agents et l'amélioration des lois de gestion des conflits.

## Remerciements

Nous tenons à remercier Y. Demazeau pour ses commentaires sur la partie multi-agents, sans oublier les conseils de J. Rouault et G. Tassart.

## BIBLIOGRAPHIE :

[Berrendonner 1990] Berrendonner A., "Grammaire pour un analyseur, aspects morphologiques", *Les Cahiers du CRISS*, N° 15, Novembre 1990.

[Berrendonner 92] Berrendonner A., "Précis de grammaires polycatérales", (à paraître).

[Bond & Gasser 1988] Bond A.H., Gasser, L., (eds.) "Reading in Distributed Artificial Intelligence", Morgan Kaufmann, 1988.

[Demazeau & Mueller 1990] *Proceedings of the First European Workshop on Modeling an Autonomous Agent in a Multi-Agent World*, "Decentralized Artificial Intelligence", Demazeau Y., and Mueller, J-P, (eds.) Elsevier Science Publishers (North Holland), July 1990.

[Erman & al. 1980] Erman L. D., Hayes-Roth F., Lesser V. R., Reddy D. R., "The Hearsay-II speech understanding system : integrating knowledge to resolve uncertainty", *ACM Computing Surveys* 12, 1980.

[Gasser & al. 1987], Gasser L., Braganza C., Herman N., "Implementing Distributed Artificial Intelligence Systems using MACE", *Proceedings of Third IEEE Conference on Artificial Intelligence Applications*, 1987.

[Gasser & Ferber 1991] Gasser L., Ferber J., "Introduction à l'intelligence artificielle distribuée et aux systèmes multi-agents", Cours n° 9 Avignon (Expert systems & their applications), Eleven international workshop on expert systems, Avignon, Mai 1991.

[Huhns 1987]. Huhns M.N., (ed.), "Distributed artificial intelligence", Pitman Publishing-Morgan Kaufman, 1987.

[Lallich & al. 1990] Lallich-Boidin G., Henneron G. et Palermi R. : *French System : Achievement and implementation of morpho-syntactic parser* ; in

ESPRIT P2474 MM12 Deliverable d18. Grenoble, CRISS, Janvier 1990.

[Newell & Simon 1972]. Newell A., Simon H.E., "*Human problem solving*", Englewood Cliffs, 1972.

[Nii 1986] Nii P. H., "Blackboard Systems", *AI Magazine, Volume 7*.

[Minsky 1988] Minsky N. H., "Law-Governed Systems", *Rapport de Recherche de l'Université de Rutgers*, February 1988.

[Minsky 1991] Minsky N. H., "The imposition of protocols over open distributed systems", *IEEE Transactions on Software Engineering* February 1991.

[MM12 1989] Esprit Project 2474. MM12. Common meaning representation by D. Sedlock. Report Bim/13, October, 1989.

[Oquendo et al. 1990] Oquendo F., Tassart G., Zucker J.D., "Support for Software Tool Integration and Process-centered Software Engineering Environments", *Proceedings of the Third International Conference on Software Engineering and its Applications*, Toulouse, France, décembre 1990.

[Rady 1983] Rady M., "L'ambiguïté du langage naturel est-elle la source du non-déterminisme des procédures de traitement ?", *Thèse de Doctorat d'état, Paris VI*, juin 1983.

[Sabah 1990] Sabah G., "Caramel : un système multi-experts pour le traitement automatique des langues", *Modèles linguistiques Tome XII Fascicule 1*.

[Sabah 1991] Sabah G., "Parallélisme et traitement automatique des langues", *Actes du colloque ILN'91*.

[Stefanini & Oquendo 1992] Stefanini M. H., Oquendo F., *Talisman : Un environnement en ingénierie linguistique fondée sur une approche multi-agents, Rapport de Recherche du CRISS (à paraître)*.

[Zweigenbaum 1989] Zweigenbaum P., "Hélène : Compréhension de compte-rendus d'hospitalisation". *Deuxième Ecole d'été sur le Traitement des Langues Naturelles, L'ENSSAT, Lannion, Juillet 1989*.