HAL
open science

Modèles d'activités pédagogiques et support de l'interaction pour l'apprentissage d'une langue – le système Sampras

Johan Michel

▶ **To cite this version:**

Johan Michel. Modèles d'activités pédagogiques et support de l'interaction pour l'apprentissage d'une langue – le système Sampras. domain_stic.othe. Université du Maine, 2006. Français. NNT : . tel-00090250

HAL Id: tel-00090250

https://theses.hal.science/tel-00090250v1

Submitted on 29 Aug 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse

présentée
en vue de l'obtention du titre de Docteur
de L'Université du Maine

spécialité : Sciences
mention : Informatique

par

Johan Michel

Modèles d'activités pédagogiques et de support à l'interaction pour l'apprentissage d'une langue

Le système Sampras

SOUTENUE LE 26 JUIN 2006 DEVANT LE JURY COMPOSÉ DE :

Président :
M. François Mangenot, Professeur à l'Université Stendhal - Grenoble 3

Rapporteurs :
Mme Anne Nicolle, Professeur à l'Université de Caen
M. Thierry Chanier, Professeur à l'Université de Franche-Comté

Examinateurs :
M. Georges Antoniadis, Maître de Conférences à l'Université Stendhal Grenoble 3
M. Jérôme Lehuen, Maître de Conférences à l'Université du Maine, responsable scientifique
M. Pierre Tchounikine, Professeur à l'Université du Maine, directeur

Thèse préparée au sein du Laboratoire d'Informatique de l'Université du Maine au Mans
Faculté des Sciences - Université du Maine - avenue Olivier Messiaen, 72085 - Le Mans cedex 9

Remerciements

Je remercie en tout premier lieu Jérôme Lehuen, Maître de Conférences à l'Université du Maine, de m'avoir accompagné au cours de ces recherches. Sa confiance en moi et son enthousiasme communicatif m'ont permis de mener à bien ces travaux, m'aidant dans les difficultés et me laissant libre d'avancer à ma façon.

Je remercie Pierre Tchounikine, Professeur à l'Université du Maine et directeur du LIUM. J'ai pu apprécier sa sagacité scientifique au cours des quelques années passées au laboratoire.

Je remercie Anne Nicolle, Professeur à l'Université de Caen, de m'avoir fait l'honneur d'accepter d'être rapportrice de cette thèse. Nos interactions m'ont permis de voir sous un autre jour certaines parties de mon travail.

Je remercie Thierry Chanier, Professeur à l'Université de Franche-Comté, de m'avoir fait l'honneur d'accepter d'être rapporteur de cette thèse. Son exigence m'a permis d'approfondir et clarifier le propos de mon document.

Je remercie Georges Antoniadis, Maître de Conférences à l'Université Stenhal-Grenoble 3, de m'avoir fait l'honneur de faire partie de mon jury de thèse.

Je remercie François Mangenot, Professeur à l'Université Stenhal-Grenoble 3, de m'avoir fait l'honneur de présider le jury de ma soutenance de thèse.

Je remercie Evelyne Maignan de m'avoir permis de pouvoir tester le prototype Sampras auprès des apprenants dont elle s'occupe à l'Université du Maine. Je remercie aussi l'ensemble des apprenants pour leur précieuse participation.

Je remercie l'ensemble des membres du LIUM avec qui j'ai partagé d'agréables moments et qui m'ont aidé soit au cours de la thèse, soit lors de la préparation de la soutenance et cela même jusqu'au tout dernier moment (n'est-ce pas Thierry ?). Je remercie les doctorants avec qui j'ai collaboré et sympathisé au cours de ces années : Christelle,

Bérangère, Julie, Mathilde, Ludovic, Mikaël, Sofiane, sans oublier Gilles et Moustachman avec qui j'ai formé un des plus beaux trios musicals qui soit. Dorothée, Naïma, et Omar resteront quant à eux mes fidèles compagnons de lutte. Je ne peux oublier Yuchen et Agnès qui ont pris part, à leur façon, à cette aventure. Je remercie aussi « Géwawh », Bruno Richard, Etienne Micoulaud qui se sont toujours montrés attentifs à mes divers problèmes techniques. Je remercie aussi les secrétaires en particulier Marie-Claude qui a tricoté un très beau fourreau pour ma clé USB.

Je remercie aussi divers thésards de l'Université du Maine : Loïc, Mérouane, Ours, le Panda, Thomas, W.O.II., Aroune, les membres de la MLF. Nos footballs, nos soirées à la cantine et ailleurs ont fait l'accompagnement de cette thèse. Que leurs compagnes soient ici dédommagées de mes exactions passées, présentes et à venir.

Je remercie l'ensemble de mes proches et de mes amis qui m'ont soutenu jusqu'au jour de la soutenance. Merci à Alice qui m'a toujours encouragé au cours de ce long travail, partagé mes doutes comme mes avancées, et même « coaché » assidûment dans les derniers instants de la thèse. Merci aussi à Daniel de m'avoir donné l'asile les tous derniers mois de ce travail. Merci enfin à ma famille, mes parents et frères de m'avoir accompagné dans mes choix, assisté quand j'en avais besoin et de m'avoir toujours donné toute latitude.

Sommaire

INTRODUCTION 9

CHAPITRE 1 PROBLÉMATIQUE 15

1.1 ORIGINES ET CADRE DE NOS TRAVAUX DE RECHERCHE 17
1.1.1 TRAVAUX DE RECHERCHE SUR LE DIALOGUE HOMME/MACHINE AU LIUM 17
1.1.2 TRAVAUX DE RECHERCHE SUR LA CONCEPTION D'ENVIRONNEMENTS INFORMATIQUES POUR L'APPRENTISSAGE HUMAIN AU LIUM 18
1.2 OBJECTIF DES RECHERCHES 19
1.3 ELEMENTS THEORIQUES : APPRENTISSAGE DES LANGUES ET CONCEPTION D'ENVIRONNEMENTS INFORMATIQUE D'APPRENTISSAGE DES LANGUES 20
1.3.1 APPRENTISSAGE DES LANGUES 20
1.3.1.1 Approche communicative 20
1.3.1.2 Acquisition d'une seconde langue 21
1.3.2 CONCEPTION DES EIAH LANGUES 25
1.3.2.1 Apprentissage des Langues Assisté par Ordinateur ? 25
1.3.2.2 Migration des théories des sciences humaines aux EIAH langues 26
1.4 ENVIRONNEMENTS INFORMATIQUES D'APPRENTISSAGE DES LANGUES 29
1.5 NOS PROPOSITIONS 34
1.6 METHODOLOGIE 36
1.6.1 DEMARCHE ET HISTORIQUE 36
1.6.2 MISES A L'ESSAI 37
1.7 SYNTHESE 38

CHAPITRE 2 MODÉLISATIONS 39

2.1 TACHE 43
2.1.1 TACHES PEDAGOGIQUES 43
2.1.1.1 Intention didactique 44
2.1.1.2 Tâches pédagogiques dans notre environnement 46
2.1.2 TACHE APPLICATIVE 47
2.1.2.1 Choix de la tâche applicative 48
2.1.2.2 Définition des éléments de l'interface du monde et constitution du lexique 48
2.1.2.3 Description de la tâche applicative 49
2.1.3 TACHE INTERACTIVE 52
2.1.3.1 Possibilités interactives dans le micromonde 53
2.1.3.2 Activité de compréhension 55
2.1.3.3 Activité de production 57
2.2 INTERACTION 60
2.2.1 DU CONCEPT D'INTERACTION DANS LES EIAH 60
2.2.1.1 Dialogue homme-machine et tuteurs intelligents 61
2.2.1.2 Dialogue homme-machine et apprentissage des langues 62
2.2.2 TYPES D'INTERACTION 64
2.2.2.1 Nature des échanges 64
2.2.2.2 Description des interactions 64
2.2.2.3 Types d'interaction au cours de l'activité de compréhension 65
2.2.2.4 Types d'interaction au cours de l'activité de production 68
2.2.3 MODELE D'INTERACTION 70
2.2.3.1 Représentation des échanges 70

2.2.3.2 Déroulement des échanges ... 79
2.3 LANGUE ... 86
2.3.1 TRAITEMENT AUTOMATIQUE DES LANGUES ET APPRENTISSAGE DES LANGUES ASSISTE PAR ORDINATEUR ... 86
2.3.2 ANALYSE ET GENERATION EN CONTEXTE ... 89
2.3.2.1 Rétroaction ... 89
2.3.2.2 Analyse ... 90
2.3.2.3 Génération de la réponse ... 92
2.3.3 LEXIQUE ... 93
2.3.4 UNE ANALYSE, SUPPORT A L'INTERACTION : COMMENT ? ... 95
2.3.4.1 Observation de l'énoncé ... 96
2.3.4.2 Comparaison des énoncés à la description de la tâche ... 106
2.3.4.3 Génération ... 109
2.4 SYNTHESE ... 112

CHAPITRE 3 L'ENVIRONNEMENT SAMPRAS ... 115

3.1 ARCHITECTURE ... 117
3.1.1 ARCHITECTURE DES SYSTEMES DE DIALOGUE ... 117
3.1.2 ARCHITECTURE DE SAMPRAS ... 118
3.2 ASPECTS LIES A L'INTERFACE ... 120
3.2.1 PARTIE GRAPHIQUE ... 120
3.2.1.1 Présentation ... 120
3.2.1.2 Fonctionnement ... 121
3.2.2 MODULE INTERFACE ... 124
3.2.2.1 Représenter le monde ... 124
3.2.2.2 Gérer les interventions du partenaire et de l'apprenant ... 125
3.3 ANALYSE ... 129
3.3.1 LES CONNAISSANCES LEXICALES ... 129
3.3.2 LE MODULE ANALYSE ... 132
3.3.2.1 Exploration de l'énoncé ... 133
3.3.2.2 Vérifications et manipulations ... 136
3.4 LA TACHE APPLICATIVE ... 137
3.4.1 REPRESENTER LA TACHE ... 137
3.4.2 GERER LA TACHE ... 140
3.5 MODULE GENERATION ... 142
3.5.1 RECHERCHE DE LA TACHE SANS INFLUENCE DES INTERACTIONS PRECEDENTES ... 142
3.5.2 RECHERCHE DE LA TACHE AVEC INFLUENCE D'INTERACTIONS PRECEDENTES ... 148
3.6 MODULE INTERACTION ... 151
3.6.1 DESCRIPTION ... 151
3.6.2 FONCTIONNEMENT ... 154
3.7 MISES A L'ESSAI ... 157
3.7.1 PREMIERES MISES A L'ESSAI, HIVER 2004-2005 ... 158
3.7.1.1 Déroulement des mises à l'essai ... 158
3.7.1.2 Observations ... 159
3.7.2 DEUXIEMES MISES A L'ESSAI, HIVER 2005-2006 ... 161
3.7.2.1 Déroulement des mises à l'essai ... 161
3.7.2.2 Observations ... 162

CONCLUSION ET PERSPECTIVES ... 165

ANNEXES ... 171

ANNEXE A – TRACES D'UTILISATION DE SAMPRAS..... 173

ANNEXE B – IMPLÉMENTATION DE SAMPRAS 192

ANNEXE C – LEXIQUE DE SAMPRAS..... 215

ANNEXE D – DOCUMENT POUR LES APPRENANTS..... 221

TABLE DES ILLUSTRATIONS 227

RÉFÉRENCES BIBLIOGRAPHIQUES 230

INTRODUCTION

Contexte général

L'étude et les travaux présentés dans ce mémoire s'inscrivent dans le cadre des recherches sur la conception des Environnements Informatiques pour l'Apprentissage Humain (EIAH). Nos travaux relèvent plus spécifiquement du domaine des environnements pour l'apprentissage des langues étrangères. Les recherches menées dans ce domaine sont multiples et recoupent de nombreux autres domaines allant de la didactique des langues à l'informatique en passant par les traitements automatiques de la langue (TAL) et le dialogue homme-machine (DHM). Ces recherches adoptent des approches pédagogiques différentes et se matérialisent au travers de logiciels hétéroclites quant aux contenus et aux choix en matière d'interaction. Notre approche concerne la conception d'environnements où une langue, le français langue étrangère (FLE) en l'occurrence, est mise en jeu au travers d'une interaction entre un apprenant et un agent conversationnel (que nous nommons partenaire) dans un micromonde. Nous nous basons sur des travaux en didactique des langues, en conception d'EIAH et en dialogue homme-machine.

Problématique

Le projet repose sur l'hypothèse que la conversation avec un agent virtuel dans une langue étrangère est susceptible de créer des situations dans lesquelles l'apprenant puisse, non seulement faire usage de la langue, mais aussi acquérir des connaissances, lexicales ou syntaxiques, sur celle-ci. Dans ce mémoire, nous faisons une large utilisation des termes tâche et activité. Nous définissons le terme tâche comme une prescription, une contrainte de départ et le terme activité comme la réalisation effective de la tâche.

Deux problématiques se croisent dans notre travail :

- ⇨ La mise en place d'activités pédagogiques : partant de travaux en didactique des langues étrangères, il s'agit de créer et modéliser des activités pédagogiques interactives.

- ⇨ Le support à l'interaction : il s'agit de définir les traitements adaptés de la langue dans l'interaction et d'organiser la tâche applicative choisie comme prétexte à l'interaction

Ces deux problématiques ne sont pas au même niveau : la première, plus conceptuelle, définit le fonctionnement souhaité, imaginé, de l'environnement. La seconde se rapproche des problèmes informatiques qui vont se poser lors de la mise en oeuvre.

Travail réalisé

Nous sommes partis de travaux en didactiques des langues étrangères et de leurs prises en compte dans le développement d'environnements informatiques d'apprentissage des langues :

- ⇨ Pour identifier les approches propices à la conception des activités pédagogiques de l'environnement.
- ⇨ Pour comprendre l'acquisition en langue seconde et identifier les phénomènes qu'il s'agit de supporter dans l'interaction.

Nous avons étudié le champ de la conception des environnements d'apprentissages des langues, dans le but de saisir les enjeux de celui-ci, de collecter des expériences proches des nôtres, et pour comprendre les limites et les problèmes rencontrés tant au niveau pédagogique qu'au niveau technique. De cette étude préliminaire, nous avons dégagé trois dimensions qui vont structurer notre approche et ce manuscrit : tâche, interaction et langue.

- ⇨ **Tâche** : Nous nous intéressons à trois niveaux de tâche : pédagogique, applicative et interactive. Les tâches pédagogiques sont les exercices proposés à l'apprenant. La tâche applicative est la tâche du monde réel dans la réalisation de laquelle nous désirons impliquer l'apprenant. La tâche interactive est la tâche que réalise effectivement l'apprenant en interaction avec le système.
- ⇨ **Interaction** : Nous nous intéressons aux types d'interaction employés pour résoudre la tâche applicative et permettre des séquences interactives susceptibles de faciliter l'apprentissage en langue dans le cadre des activités pédagogiques (i.e la réalisation de

la tâche pédagogique). À partir de ces types d'interaction, nous proposons un modèle global rendant compte de la dynamique de l'interaction.

- ⇨ **Langue** : Nous nous intéressons aux traitements de la langue naturelle qui vont aider à supporter l'interaction. Ces traitements nécessitent la création d'un modèle du lexique et la modélisation des mécanismes d'analyse des énoncés de l'apprenant comme de la génération des énoncés du partenaire.

Nous avons mis en oeuvre nos modèles au sein d'un environnement informatique appelé « Sampras ». Cet environnement reprend les trois dimensions abordées ci-dessus :

- ⇨ **Tâche** : Sampras propose à l'apprenant de s'impliquer dans la réalisation d'une activité au sein d'un micromonde : la réalisation interactive d'une recette de cuisine.
- ⇨ **Interaction** : Sampras propose des types d'interaction dédiés à l'exercice de la compétence de production dans le cadre de la tâche interactive.
- ⇨ **Langue** : Sampras intègre un analyseur tolérant à la non-normativité des énoncés et orienté vers la gestion de l'interaction.

Nous avons « mis à l'essai » cet environnement deux fois auprès d'un même public : des apprenants thaïs et chinois grands débutants en français langue étrangère qui passent une première année d'adaptation à l'Université du Maine. Cette première « mise à l'essai » (fin 2004/début 2005) avait pour but, premièrement de tester l'utilisabilité de l'environnement, deuxièmement d'avoir une première vision des échanges pouvant avoir lieu dans le cadre d'une utilisation concrète, et troisièmement de relever les avis que pouvaient avoir les apprenants sur cet environnement. La seconde campagne (début 2006) poursuit plus en avant ces tests dans le but de constituer un corpus et d'identifier des phénomènes caractéristiques d'usage de la langue.

Nous ne prétendons pas apporter de façon identique sur les différents thèmes de recherche que nous croisons. Les aspects traitement de la langue (lexique, analyse) et ergonomie (tâche interactive) ne sont pas développés comme ils l'auraient été s'il s'agissait d'une thèse de ce domaine. La thèse est une thèse en EIAH et les contributions au domaine des EIAH et en particulier à celui des EIAH en langues concernent :

- ⇨ La définition de **modèles** et d'une **architecture** pour un environnement informatique, c'est-à-dire la façon dont les trois dimensions s'articulent.

⇨ La conception et le développement d'un **prototype** informatique dont certains composants peuvent être réutilisés ou transposés pour d'autres applications.

Approche méthodologique

L'approche techno-pédagogique [Tchounikine, 2002] fournit un cadre pratique pour la conception d'un EIAH. Elle implique la prise en compte de considérations pédagogiques et didactiques dans la construction des EIAH. Nous avons donc cherché à conceptualiser les dimensions pédagogiques et didactiques pour ensuite construire les modèles à réifier.

L'élaboration de Sampras a débuté par une première version créée autour d'un jeu de reconnaissance de couleurs [Michel, 2001] stabilisant l'interface Java et rendant compte des possibilités du moteur d'inférence Jess. En parallèle s'est déroulée la constitution d'un corpus d'énoncés d'apprenants étrangers pour l'identification des erreurs possibles [Lehuen *et al.*, 2002]. Le modèle de résolution de la tâche séparant le niveau interaction du niveau tâche a lui été créé plus tard [Delorme, 2002].

Notre étude s'est déroulée de la façon suivante :

- Identification des caractéristiques des EIAH langues en se fondant sur la didactique
- Confrontation des caractéristiques identifiées à l'approche « micromonde »
- Identification des trois dimensions de l'étude (tâche / interaction / langue)
- Définitions des modèles liés aux trois dimensions
- Développement du prototype informatique (Sampras)
- Mises à l'essai avec des apprenants étrangers
- Correction / adaptation de l'environnement

Organisation du mémoire

Le premier chapitre présente la problématique de la thèse. Partant d'un contexte général situant les travaux au confluent de deux courants de recherche au LIUM, ce chapitre fixe les objectifs de recherche, explore les domaines de référence de la thèse et propose des modèles à concevoir et à mettre en oeuvre, ainsi que la méthode adoptée pour les réaliser.

Nous partons ainsi du cadre général de l'apprentissage des langues et des EIAH pour définir nos objectifs de recherche. Ce chapitre débouche sur la formulation de propositions de modélisations.

Le second chapitre se focalise sur les modélisations réalisées lors de la conception de l'environnement Sampras. Ces modélisations concernent les trois dimensions identifiées dans les propositions : tâche, interaction et langue. La partie tâche décrit l'intention didactique qui motive le projet, les tâches pédagogiques, la tâche applicative qui est prétexte à l'interaction et la tâche interactive. La partie interaction s'attache à la modélisation de l'interaction dans l'environnement, les types d'interaction et le modèle global y sont détaillés. La partie langue présente les mécanismes d'analyse des énoncés et de génération des interventions du partenaire.

Le troisième chapitre, plus technique, présente le prototype développé, qui réifie les modélisations proposées dans le chapitre deux. L'interface homme-machine (écrite en Java) est présentée au travers de diagrammes UML. La présentation des différents modules de la base de connaissances (écrite en Jess, implémentation Java du moteur d'inférences CLIPS) se fait à partir d'exemples et d'éléments du code source qui aident à comprendre leur rôle et leur fonctionnement.

Nous terminons par des conclusions et perspectives.

En annexe figurent quelques traces d'utilisation de l'environnement, différents éléments du code source, le lexique de la recette et le document donné aux apprenants lors des mises à l'essai.

Chapitre 1
Problématique

1.1 ORIGINES ET CADRE DE NOS TRAVAUX DE RECHERCHE 17
1.1.1 TRAVAUX DE RECHERCHE SUR LE DIALOGUE HOMME/MACHINE AU LIUM 17
1.1.2 TRAVAUX DE RECHERCHE SUR LA CONCEPTION D'ENVIRONNEMENTS INFORMATIQUES POUR L'APPRENTISSAGE HUMAIN AU LIUM 18
1.2 OBJECTIF DES RECHERCHES 19
1.3 ELEMENTS THEORIQUES : APPRENTISSAGE DES LANGUES ET CONCEPTION D'ENVIRONNEMENTS INFORMATIQUE D'APPRENTISSAGE DES LANGUES 20
1.3.1 APPRENTISSAGE DES LANGUES 20
1.3.1.1 Approche communicative 20
1.3.1.2 Acquisition d'une seconde langue 21
1.3.2 CONCEPTION DES EIAH LANGUES 25
1.3.2.1 Apprentissage des Langues Assisté par Ordinateur ? 25
1.3.2.2 Migration des théories des sciences humaines aux EIAH langues 26
1.4 ENVIRONNEMENTS INFORMATIQUES D'APPRENTISSAGE DES LANGUES 29
1.5 NOS PROPOSITIONS 34
1.6 METHODOLOGIE 36
1.6.1 DEMARCHE ET HISTORIQUE 36
1.6.2 MISES A L'ESSAI 37
1.7 SYNTHESE 38

Ce premier chapitre expose la problématique de notre thèse : l'origine de notre recherche et le cadre dans lequel elle s'inscrit. Nous définissons ensuite les objectifs de notre recherche et les éléments théoriques qui les étayent. Nous délimitons ensuite les recherches proches des nôtres et exposons nos propositions. Le dernier point du chapitre porte sur la méthode que nous avons adoptée pour mener à bien ce projet.

1.1 Origines et cadre de nos travaux de recherche

Nos travaux se trouvent au confluent de deux thèmes de recherche au Laboratoire d'Informatique de l'Université du Maine (LIUM) : le thème dialogue homme machine et le thème ingénierie des EIAH. La thèse que nous présentons est clairement inscrite dans le second thème mais se sert des quêtes et résultats du premier. Le thème dialogue homme machine s'attache à développer des modélisations du dialogue, du langage employé dans le dialogue ou de la tâche à accomplir par l'utilisateur. Ces modélisations sont plus ou moins non dédiées, elles évitent de s'appuyer sur une planification figée et intègrent la dynamique du langage. Le thème ingénierie des EIAH se focalise sur l'élaboration d'un ensemble de connaissances permettant de fonder le processus de conception des EIAH. Nous exposons dans cette section les recherches de ces deux courants.

1.1.1 Travaux de recherche sur le dialogue homme/machine au LIUM

Les travaux menés au LIUM concernant le dialogue homme-machine ont débuté sous l'impulsion de Daniel Luzzati et ont donné lieu à plusieurs thèses dont celles de Jérôme Lehuen et de Thierry Lemeunier. Les travaux de Daniel Luzzati opèrent le dépassement du modèle genevois de représentation formelle du dialogue en y intégrant une composante dynamique [Luzzati, 1995b]. Le dialogue est alors vu comme un processus se déroulant sur deux axes : un axe régissant et un axe incident. L'axe régissant représente le déroulement de l'activité à accomplir par l'utilisateur du système de dialogue et l'axe incident représente la résolution des problèmes interactifs (questions, clarifications) qui apparaissent. Le but est de repérer les phénomènes bloquants (c'est-à-dire qui empêchent l'activité de progresser) et de revenir le long de l'axe régissant (c'est-à-dire de continuer l'activité à accomplir). La thèse de Jérôme Lehuen visait à l'élaboration d'un modèle de dialogue dynamique et générique

intégrant l'acquisition de sa propre compétence linguistique et à sa réification dans un prototype (nommé COALA). L'originalité de cette recherche était que le système informatique construisait son propre lexique et sa propre syntaxe en dialoguant avec les utilisateurs finaux [Lehuen, 1997]. La thèse de Thierry Lemeunier portait sur l'intentionnalité communicative dans le dialogue homme-machine en langue naturelle et proposait une contribution originale permettant au système informatique d'avoir ses propres intentions de communication avec ses utilisateurs finaux [Lemeunier, 2000]. À l'heure actuelle, ces chercheurs, sans quitter le champ du dialogue homme-machine, inscrivent désormais leurs recherches dans le courant de l'ingénierie des EIAH au sein des projets Dialects et Mepa. Dialects porte sur la conception d'un environnement d'apprentissage de la modélisation orientée objet en UML, où l'utilisateur-apprenant conçoit des modèles UML et s'il commet des erreurs de modélisation, des dialogues en langue naturelle prennent place pour remédier aux erreurs. Mepa, conduit par Jérôme Lehuen, a pour objectif principal, la conception et le développement de systèmes informatiques *client-serveur* qui permettent à des apprenants, à des tuteurs et à des agents virtuels, d'agir et d'interagir dans des espaces virtuels partagés, dans le cadre de scénarios pédagogiques.

1.1.2 Travaux de recherche sur la conception d'environnements informatiques pour l'apprentissage humain au LIUM

Le LIUM s'est focalisé depuis sa création sur l'élaboration d'environnements informatiques pour l'apprentissage humain [Bruillard *et al.*, 2000]. Il existe deux grands types d'environnements informatiques pour l'apprentissage humain : les environnements orientés « pédagogie » qui se focalisent sur l'organisation et l'enseignement de contenus bien identifiés et les environnements orientés « performance » qui mettent en place des situations d'apprentissage fondées sur la réalisation d'activités pédagogiques avec des contenus à apprendre dans l'interaction avec ces environnements [Tchounikine, 2002]. Les environnements créés au LIUM appartiennent principalement à la seconde catégorie. Nous trouvons des travaux sur l'instrumentation du tuteur et l'aide portée à l'autonomie d'un apprenant en FOAD (Formation ouverte et à distance), [Despres, 2001], [Coffinet-Rasseneur, 2004], [Laperrousaz *et al.*, 2005], [Gueye, 2005]. D'autres travaux se focalisent sur les apports des normes à la ré-ingénierie des EIAH [Barré *et al.*, 2004], sur le support d'activités collaboratives à distance [Betbeder M-L *et al.*, 2003], [Taurisson *et al.*, 2005]. Les travaux, plus anciens, qui ont donné lieu à des réalisations informatiques et qui nous concernent le

plus sont ceux autour de la conception des situations d'interaction dans les EIAH [Delozanne, 1992], [Dubourg, 1995]. D'un point de vue plus théorique, la réflexion entreprise au sein de la communauté française quant à une approche générale de la conception des EIAH aide à orienter et caractériser notre travail [Tchounikine *et al.*, 2004].

1.2 Objectif des recherches

Notre étude porte sur la conception d'un type particulier d'environnements informatiques pour l'apprentissage des langues étrangères. L'idée première est de repartir des travaux en dialogue homme-machine et de voir si leur exploitation permettait des avancées quant à la réalisation d'environnements originaux. L'objectif est la création de tâches pédagogiques interactives nécessitant le dialogue, la plus évidente de ces tâches étant la simulation énonciative. Les simulations énonciatives sont des situations dans lesquelles l'apprenant interagit avec un interlocuteur virtuel en utilisant directement la langue. Les simulations, dans le cadre des environnements informatiques d'apprentissage des langues, s'instancient sous la forme de micromondes qui fournissent à l'interaction entre l'apprenant et son partenaire un cadre de référenciation.

Les simulations énonciatives permettent des apprentissages en langue, apprentissage principalement de type implicite. L'objectif général est de contribuer à la conception et modélisation de ces simulations énonciatives. Les simulations s'organisent autour d'une tâche à accomplir selon certaines modalités, tâche et modalités caractérisant le langage à employer.
Nous devons donc disposer de connaissances sur la tâche applicative dans la réalisation de laquelle l'apprenant est impliqué, sur les interactions qu'il réalise pour ce faire, sur les caractéristiques du langage qu'il est amené à employer dans ce cadre. Ces connaissances induisent des modèles qu'il s'agit de mettre en oeuvre.

Plus généralement, nous observons ici le problème classique en conception d'EIAH qu'est la définition de l'intention didactique de cet environnement et les conséquences en termes de modélisation puis d'articulation avec les spécifications de l'artefact créé [Tchounikine *et al.*, 2004].

Cet objectif se décompose en plusieurs sous-objectifs qui nourrissent les trois dimensions du projet. Le premier objectif est de définir un cadre théorique en apprentissage des langues, tant pour la conception de la tâche pédagogique que pour la définition des séquences interactives susceptibles d'induire des apprentissages. La définition de l'intention didactique passe par la migration de modèles préexistants en apprentissage des langues vers les EIAH langues. Le deuxième objectif est d'identifier les recherches en dialogue homme-machine susceptibles d'apporter des modèles et concepts intéressants vis-à-vis de cadre théorique défini en apprentissage des langues, en particulier les travaux réalisés dans le domaine des EIAH sur le concept d'interaction. Le dernier objectif est d'étudier les implications du traitement automatique de la langue dans le domaine des environnements d'apprentissage des langues et plus particulièrement dans le cadre des systèmes qui « dialoguent » avec leur utilisateur. Ces objectifs atteints permettent la modélisation des dimensions tâches, interaction et langue, à l'articulation de ces trois dimensions et à la mise en oeuvre de l'environnement informatique.

1.3 Eléments théoriques : Apprentissage des langues et conception d'environnements informatique d'apprentissage des langues

Ce paragraphe présente les éléments théoriques sur lesquels reposent certaines de nos propositions et permettent de caractériser nos objectifs du point de vue de la didactique des langues étrangères.

1.3.1 Apprentissage des langues

1.3.1.1 Approche communicative

La notion de compétence communicative [Hymes, 1972] émerge en addition à la notion de compétence linguistique [Chomsky, 1975] et affirme que « *communiquer langagièrement ne se réduit pas à une connaissance des formes et règles linguistiques mais suppose aussi une maîtrise des conventions d'utilisation, socialement et pragmatiquement définies* » [Cuq, 2003]. Le cadre défini par l'approche communicative juxtapose à la compétence linguistique les compétences sociolinguistique (inscription des faits de langage

dans une pratique sociale) et pragmatique (connaissance des effets de la forme des actes de langage).

Le postulat psychologique qui sous-tend cette approche est « *qu'apprendre une langue, c'est s'entraîner dès le départ à penser et à parler en langue étrangère* » [Puren, 2001]. L'apprentissage se passe alors dans et par l'interaction. Les exercices de référence de cette approche sont les simulations et les jeux de rôle où il faut communiquer dans la langue étrangère de manière authentique ou simulée pour développer la compétence communicative.

Continuant ces recherches, le Cadre Commun Européen de référence [l'Europe, 2001] synthétise l'approche communicative en s'inscrivant dans une modélisation actionnelle, « *centrée sur la relation entre les stratégies de l'acteur et la ou les tâche(s) à réaliser dans un environnement et dans des conditions données* ». L'acteur est ici le sujet apprenant, les stratégies se matérialisent sous la forme des actions (langagières ou non suivant la nature de la tâche) que produit l'apprenant dans l'accomplissement d'une tâche. La compétence en langues se décline alors selon quatre catégories d'activités langagières : activité interactive, activité de production, activité de compréhension, activité de médiation.

Les tâches pédagogiques communicatives (contrairement aux exercices formels hors contexte) visent à impliquer l'apprenant dans une communication réelle, ont un sens (pour l'apprenant), sont pertinentes (ici et maintenant dans la situation formelle d'apprentissage), exigeantes mais faisables (avec un réajustement de l'activité si nécessaire) et ont un résultat identifiable (ainsi que d'autres moins identifiables dans l'immédiat). Dans une tâche communicative, l'accent est mis sur le succès de l'exécution de la tâche et, en conséquence, le sens est au centre du processus tandis que les apprenants réalisent leurs intentions communicatives. C'est à ce niveau qu'interviennent les problèmes liés à la langue et leur implication en termes d'acquisition en langue.

1.3.1.2 Acquisition d'une seconde langue

Une première théorie, baptisée théorie interactionniste, provient de recherches issues des travaux de Krashen [Krashen, 1982] sur la notion d'input (traduit par le terme « apport langagier » dans le texte qui suit). Ces travaux correspondent à un courant de recherche majoritairement Nord-Américain. Cette théorie affirme que c'est le processus psycholinguistique à l'origine de la communication qui est responsable du développement

linguistique. S'appuyant sur le schéma réducteur input-ouput-feedback-uptake[1], elle propose cependant un cadre pratique d'utilisation, surtout dans le cadre de la classe de langues, où ont été effectuées ces recherches. Ce modèle s'organise selon le schéma suivant (Figure 1). Les concepts d'apport, saisie et production, sont des termes qui renvoient à des actes de langue alors que compréhension, intégration, création sont des termes qui se réfèrent à des mécanismes linguistiques.

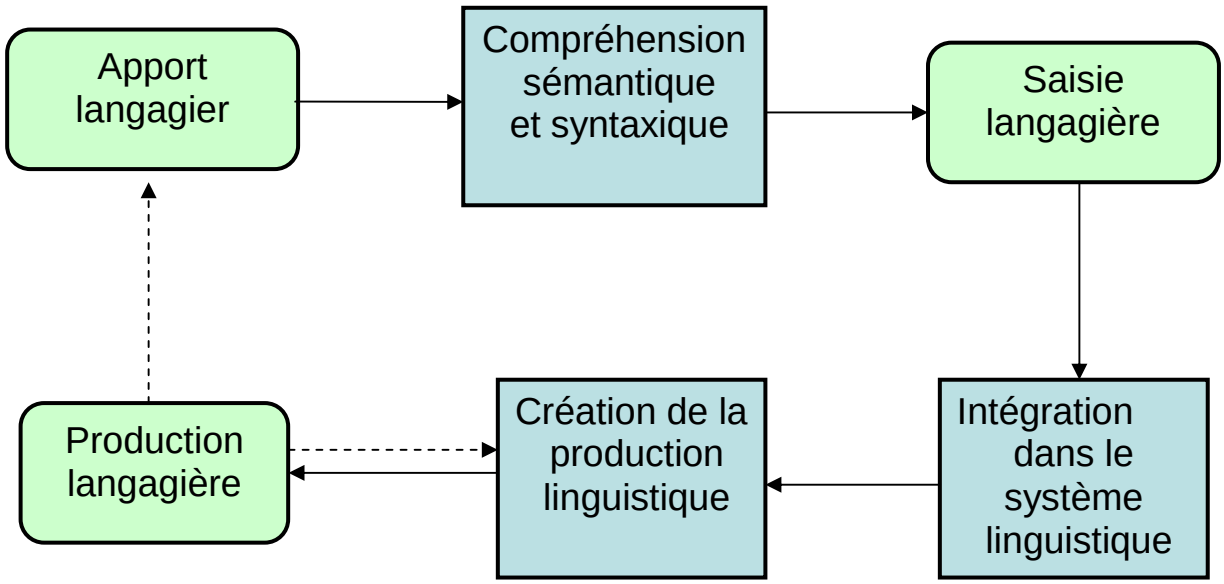


Figure 1 : Le modèle interactionniste

L'apport langagier est la langue cible parlée ou écrite à laquelle les apprenants sont exposés. La saisie langagière est la part de l'apport langagier qui est compris par l'apprenant. Cette part est ensuite intégrée dans le système linguistique de l'apprenant. La production langagière, c'est-à-dire les énoncés qu'il produit dans la langue cible, résulte de la création linguistique élaborée par l'apprenant. Sur la figure, les flèches indiquent la direction probable du processus d'acquisition linguistique, cette orientation repose sur l'idée que l'on arrive à comprendre la langue cible avant de savoir la manipuler pour créer ses propres énoncés.

Ces recherches expliquent alors leur modèle par les points suivants :

- La double compréhension de la syntaxe et de la sémantique est nécessaire pour que l'apport langagier active le processus d'acquisition. La compréhension sémantique seule

[1] Que l'on peut traduire par : « apport langagier - production langagière – retour – reprise »

n'offre que très peu d'assistance à l'apprenant, qui a besoin d'acquérir le système syntaxique pour organiser la réalité.

- Lorsqu'il y a traitement à la fois sémantique et syntaxique au cours du processus de compréhension, les caractéristiques linguistiques (donc ces éléments syntaxiques et sémantiques) de l'apport langagier sont alors saisies.
- L'intégration de la saisie langagière dans le système linguistique de l'apprenant dépend de mécanismes cognitifs utilisés pour retenir cet intrant dans la mémoire à court terme, antichambre du développement linguistique. Le système linguistique, sujet à de constantes révisions, constitue la base des connaissances nécessaires à la production langagière.

La production langagière est cruciale pour deux raisons (flèches pointillées). Tout d'abord, elle incite les apprenants à utiliser leur système syntaxique d'une façon plus appropriée qu'à l'étape de compréhension, incitant à une concentration sur les problèmes linguistiques. Ensuite, cette production sollicite un apport langagier de la part des interlocuteurs vers qui cette production aura été dirigée. Dans le cadre d'un tel échange, il est très probable que les réponses des interlocuteurs fournissent des indices qui puissent aider les apprenants à résoudre leurs problèmes linguistiques, à formuler de nouvelles hypothèses et à s'engager dans de nouvelles productions langagières [Swain, 1985], [Pica, 1994], [Pica *et al.*, 1996]. Ce processus, appelé « négociation de sens » [Long, 1985], apparaît lorsque la structure discursive normale est modifiée, du fait d'une demande de répétition, de clarification ou de reformulation de l'apport langagier initial.

Cette hypothèse importante constitue à nos yeux l'apport fondamental de la théorie interactionniste : « *La modification d'une structure interactive au cours d'une conversation, ou la modification d'un discours écrit à l'occasion d'une lecture […] est une [bonne] condition, nécessaire (mais insuffisante) pour l'acquisition d'une langue cible. Une telle négociation de sens vise la compréhensibilité de l'apport langagier même si celui-ci maintient des éléments linguistiques encore inconnus. Cette compréhensibilité émanant de la négociation permet donc une saisie langagière nécessaire à l'acquisition*. » [Larsen-freeman *et al.*, 1991]. Dans [Chapelle, 2002], nous trouvons le résumé des fondements pour l'application de la théorie :

- L'apport langagier est nécessaire mais pas suffisant pour apprendre en langue.

- L'apprenant doit observer la langue, affirmation que l'on retrouve chez Skehan [Skehan, 2002] : l'étudiant doit se concentrer sur la forme de l'apport langagier.
- L'apprenant doit comprendre l'apport langagier (du point de vue syntaxique comme du point de vue sémantique)
- L'apprenant doit avoir des occasions de produire de la langue
- L'apprenant doit remarquer ses erreurs de langue
- L'apprenant doit corriger ses erreurs de langue lui-même
- Une manière de profiter de la communication est la « négociation de sens », l'apprenant se concentre sur la forme lorsque la communication échoue.

Nous revenons sur ces hypothèses et les développements qu'ils suscitent chez Chapelle dans la partie liée au passage dans le monde informatique de cette théorie.

Les recherches européennes envisagent l'acquisition en milieu dit « naturel », valorisant les techniques de « bricolage » d'un apprentissage sur le « tas » [Gajo *et al.*, 2000]. Ces recherches se basent sur des analyses ethnographiques, l'idée étant que l'interprétation des processus de communication et d'acquisition se trouve dans l'événement langagier lui-même. L'objet de recherche est lui-même redéfini par rapport aux recherches américaines, puisque recentré sur les processus de gestion et de construction du répertoire langagier dans l'interaction [Matthey, 1996]. Le but est de repérer des moments-clés pour l'acquisition sous la forme de séquences typiques, ces séquences permettent la négociation de données linguistiques, souvent déclenchée par un problème de communication. La plus connue est la « séquence potentiellement acquisitionnelle » [De Pietro *et al.*, 1989], [Veronique, 1992], [Marcelli, 2004]. Cette dernière est caractérisée par trois phases: dans la première phase (d'auto-structuration), le non-natif dévoile un état de son interlangue (la langue telle que parlée par l'apprenant [Voguel, 1995]) et ses éventuels problèmes de communication ; dans la seconde (d'hétéro-structuration), le natif propose son aide ou son guidage ; dans la dernière, le sujet apprenant fait une reprise de l'énoncé, c'est de nouveau un moment d'auto-structuration qui permet au sujet apprenant la construction d'un objet linguistique et l'appropriation de données nouvelles. On note ici aussi le rapprochement avec la théorie interactionniste quant à la négociation de sens en tant que potentiel catalyseur d'acquisition au cours de l'interaction lors d'un problème de communication.

1.3.2 Conception des EIAH langues

Cette section est dédiée à la présentation du domaine de recherche que constitue la conception d'environnements informatique d'apprentissage en langue. La première partie présente succinctement le domaine et la seconde détaille la migration des concepts d'apprentissage/acquisition en langue vers ces environnements.

1.3.2.1 Apprentissage des Langues Assisté par Ordinateur ?

Ce champ de recherche commence à se structurer dans les années 1980 avec une première thèse spécifique à ce domaine [Demaiziere, 1986] et l'apparition des premiers didacticiels d'apprentissage des langues. Les années 1990 voient l'arrivée du Web et l'explosion des possibilités (apprentissages en réseau), et aussi le renforcement des apports au niveau méthodologie de conception, grâce en particulier aux apports des travaux sur l'acquisition en langue. Ce domaine donne lieu à une très grande variété de travaux de recherche et donc d'outils informatiques issus de ces travaux. Les disciplines connexes sont nombreuses et la figure suivante, présente ces disciplines (Figure 2).

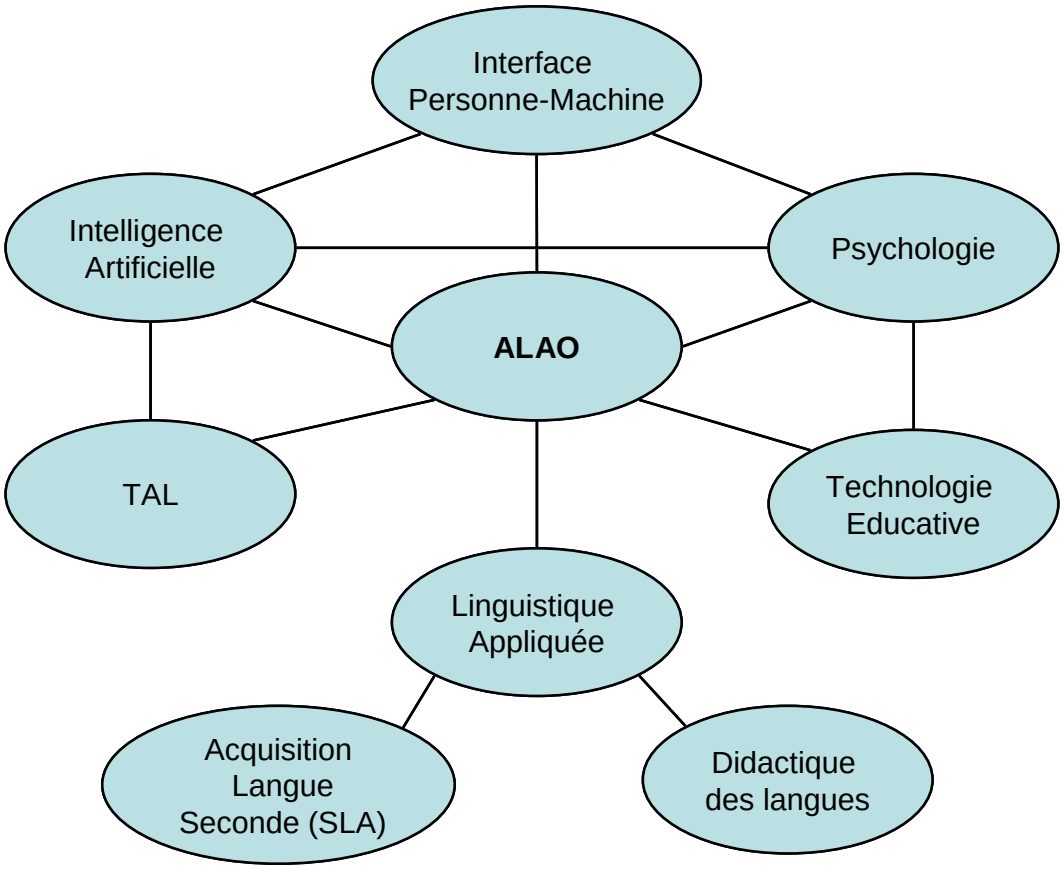


Figure 2 : Disciplines connexes de l'alao, extrait de [Chanier, 1998]

Chanier propose une classification des types d'environnements informatique pour l'apprentissage des langues [Chanier, 1998] : les outils linguistiques de présentation de matériaux verbaux ou non-verbaux, les environnements de tests de connaissance, les environnements d'apprentissage finalisés, les environnements permettant d'effectuer des recherches sur l'acquisition. Nos travaux se situent dans la classe des environnements d'apprentissage finalisés, « *environnements conçus pour l'acquisition de compétences dans la langue cible à partir d'approches multiples* ».

1.3.2.2 Migration des théories des sciences humaines aux EIAH langues

Cette section expose les différents travaux effectués sur la transposition des travaux de sciences humaines vers les EIAH langues. Nous exposons les travaux réalisés sur ce thème par Chapelle [Chapelle, 2002]. Elle reprend les hypothèses des travaux de la théorie interactionniste et émet des suggestions quant à leur mise en place dans un EIAH en langues :

- Les caractéristiques linguistiques particulières de l'apport langagier doivent être mises en évidence, même si c'est l'apprenant qui le percevra et en tiendra compte ou pas. Dans un environnement informatique, il s'agit de mettre en avant les caractéristiques linguistiques

particulières présentes dans l'apport langagier (nouveaux mots, nouvelles structures de la langue). La méthode de mise en avant est laissée au choix du concepteur de l'environnement informatique ou du professeur utilisateur.

- Les apprenants devraient recevoir de l'aide dans la compréhension sémantique et syntaxique de l'apport langagier. Dans un environnement informatique, il s'agit de donner à l'apprenant, si l'apport langagier n'est pas compris, la possibilité d'obtenir l'énoncé modifié ou des aides lui permettant de comprendre cet apport.
- Les apprenants doivent avoir l'opportunité de produire des actes de langage dans la langue cible. Dans un environnement informatique, Il s'agit de faire en sorte que les apprenants produisant des énoncés corrects soient compris. Le travail de production reflétant le travail (par demande ou accès à des renseignements) de création de la part de l'apprenant qui ne doit pas être oublié non plus. Ce point soulève le problème de la conception d'analyseurs qui permettent la compréhension des structures correctes.
- Les apprenants doivent pouvoir remarquer (le phénomène de remarquer se nomme *noticing*, terme employé au sein de la communauté didactique des langues pour désigner ce phénomène) les erreurs dans leurs productions. Dans le cadre d'un environnement informatique, il s'agit de faire en sorte que l'apprenant puisse être mis au courant de ses erreurs. Les moyens à employer pour le faire vont dépendre des moyens d'interaction qui existent entre l'apprenant et le système informatique.
- Les apprenants ont besoin de pouvoir corriger leurs productions langagières. Les apprenants doivent pouvoir modifier leurs interventions dans l'environnement informatique et avoir accès à des aides relatives à leurs erreurs.
- Les apprenants ont besoin d'être engagés dans des interactions en langue étrangère, interactions dont la structure puisse être modifiée pour permettre des séquences de négociation de sens. Dans un environnement informatique, Chapelle indique qu'il faut qu'il y ait possibilité de sortir de la réalisation de la tâche pédagogique donnée à l'apprenant lorsqu'il y a des problèmes de communication et se concentrer sur la résolution de ces problèmes de communication qui ont lieu à ce moment de l'interaction. Ces problèmes résolus, l'interaction normale reprend.
- Les apprenants devraient être engagés dans des tâches pédagogiques mises en place pour avoir un nombre maximal d'interactions profitables d'un point de vue acquisitionnel. Dans un environnement informatique, l'apprenant se consacre plus à la réalisation de la

tâche communicative que celles des problèmes linguistiques. Les problèmes linguistiques mis en évidence sont ceux qui bloquent la résolution de la tâche.

Très peu d'autres auteurs s'intéressent à la migration des théorie de l'apprentissage des langue vers les EIAH langues, on en trouve des références chez Chanier notamment dans l'usage des séquences potentiellement acquisitionnelles dans le projet Eléonore [Chanier *et al.*, 1996], où se trouve la formalisation de ces séquences mais ceci dans le cadre d'activités collaboratives mettant en scène des acteurs humains, donc déjà plus proches d'interactions entre humains même si celles-ci sont médiatisées par des machines.

Le problème principal dans un EIAH langue est de définir une tâche pédagogique. Mangenot [Mangenot, 2000], dans le cadre plus général des produits multimédia, mentionne Furstenberg [Furstenberg, 1997] laquelle insiste sur le double rôle de la tâche pédagogique dans un produit multimédia :

- faciliter l'exploration et permettre la construction d'un sens par l'utilisateur,
- évaluer ce qu'il aura compris et retiré.

Mangenot définit, ce qu'il appelle une tâche linguistique profitable dans un environnement multimédia : « *Une tâche linguistique profitable est celle qui :*

- *part de données riches et authentiques,*
- *propose des activités d'un bon niveau cognitif (liens données/activités pertinents, situations-problèmes, appel à la créativité),*
- *prévoit des interactions variées (notamment - mais pas exclusivement - évaluatrices) pendant et après l'exécution de la tâche.* »

Cette section résume les travaux que nous avons jugés pertinents pour notre étude quant à la transposition informatique de théories provenant de travaux de sciences humaines. Nous ne discutons pas pour l'instant des différentes approches didactiques et comment elles sont passées dans le monde informatique ou plus largement multimédia, tant de multiples travaux, la plupart sans rapport avec notre approche, ont été faits. Une partie de ces travaux, les plus proches de notre approche sont présentés dans ce qui suit.

1.4 Environnements informatiques d'apprentissage des langues

Cette section présente différents environnements informatiques d'apprentissage des langues dont la conception et les usages relèvent de problématiques proches. Nous nous limitons à quatre grands systèmes qui nous permettent aussi de situer notre champ de recherches plus précisément. Tout d'abord, nous présentons trois environnements issus de la première vague d'utilisation de micromondes pour l'apprentissage d'une langue selon l'approche communicative. Ces environnements furent dénommés ICALL (Intelligent Computer Assisted Language Learning) [Harrington, 1996], en référence aux techniques d'Intelligence Artificielle qui y sont utilisées, en particulier pour le traitement automatique des langues. D'ailleurs, la description de ces environnements par leurs auteurs est souvent centrée sur le composant de traitement automatique de la langue [Holland *et al.*, 1993]. Le premier système présenté est le système FLUENT développé par Henry Hamburger et son équipe. Il met en jeu une interaction bimodale entre un apprenant et un système autour d'actions simples à réaliser dans un micromonde dans le cadre de la résolution d'une tâche applicative (une recette de cuisine). Le second MILT développé par Holland et son équipe a fait le choix d'une action uniquement langagière pour l'apprenant qui commande un agent animé dans un micromonde dans le cadre d'une mission militaire (recherche de renseignements). Le troisième développé par Boylan et Micarelli est un environnement où l'apprenant est impliqué dans une communication avec un agent conversationnel. L'apprenant est plongé dans un exercice de reconstruction de conversation où il essaie de trouver les bonnes formulations de ses énoncés pour communiquer avec l'agent conversationnel. La situation mise en scène est celle de l'apprenant discutant avec un commerçant (un fleuriste). Le dernier présenté est TLTS, développé par Johnson et son équipe, un environnement plus récent. Il est plus centré sur les compétences de communication orale mais son existence est symptomatique de l'intérêt toujours gardé pour ce type d'environnements micromondes. L'apprenant y interagit avec des compagnons virtuels au cours de missions militaires (recherches de renseignements, contact avec la population) à accomplir.

Proche de nos travaux se trouve donc l'environnement FLUENT qui se caractérise par l'utilisation de techniques d'intelligence artificielle et des idées éducatives relativement novatrices à son époque [Hamburger, 1994]. S'appuyant, tout comme nous sur la démarche d'une acquisition par et dans l'interaction langagière dans la langue cible, l'apprenant est

impliqué dans une interaction avec un système (que les auteurs nomment tuteur la plupart du temps) autour d'actions à accomplir dans un micromonde (le même « monde », à savoir la cuisine, a été utilisé sans doute pour les champs lexicaux et les actions simples que cet univers génère). Les interactions y sont bimodales : le langage naturel (écrit pour l'apprenant, écrit et parlé pour le système), en l'occurrence la langue cible de l'apprentissage et des manipulations graphiques dans la cuisine. Lorsque le système agit à l'écran, une main personnifie ses actions mais il n'y a pas de représentation sous la forme d'un compagnon à l'écran.

Dans FLUENT, l'enseignant peut paramétrer les « schémas tutoriaux » qui vont décider de la coordination des différentes activités et du style conversationnel du système. La représentation des tâches applicatives à accomplir repose sur des plans et actions qui caractérisent comment le « tuteur » organise l'activité dans le micromonde, comment l'apprenant initie ses actions et comment l'état du micromonde est changé. Les « vues » sont un ensemble de structures qui fournissent les particularités de la langue, les contraintes en termes de variété conversationnelle, et permettent de calibrer la difficulté de l'apport langagier présenté à l'étudiant. La gestion de ces tâches applicatives est assurée par un raisonneur qui instancie les plans et actions en fonction de la situation courante. Les plans regroupent les sous-plans et les actions à effectuer pour accomplir un but. L'analyse des énoncés de l'apprenant et la génération des énoncés reposent sur un lexique et une grammaire et sont caractérisées par le schéma de tutorat prescrit par l'enseignant. Le composant analyseur provient du projet Athena et fut adapté spécialement pour le projet FLUENT. Le lexique est composé de cinq cent cinquante mots et contient des informations pour trouver les flexions de ces différents mots ainsi que les formes irrégulières qui peuvent exister. La grammaire est composée d'expressions régulières et de fonctions de réduction pour réduire les structures, confirmer les bonnes constructions et construire les arbres d'analyse. L'état du micromonde est géré par un module qui maintient l'état du monde et exécute les actions graphiques dans le micromonde.

Les différentes tâches pédagogiques dans FLUENT sont spécifiées dans les schémas tutoriaux qui permettent trois types d'interaction [Schoelles *et al.*, 1997]. Le plus simple est le « Tourguide » (traduisible en français par « visite guidée ») où le système accomplit des manipulations d'objets du micromonde et les verbalise. Plus interactif est le « Commander » (l'interrogateur, le « dictateur ») où le système demande à l'apprenant d'effectuer une action sur un objet du micromonde. Le système réagit alors quant à la correction de la manipulation

effectuée. Le dernier type est le « Quizmaster » où le système accomplit une action et pose une question à l'apprenant à propos de cette action. Les schémas tutoriaux consistent en l'association d'une action ou un plan avec un type d'interaction et un ensemble de « vues ». Ainsi une leçon est constituée d'un enchaînement de schémas tutoriaux. Les « vues » servent d'intermédiaires entre les fonctions pédagogiques du système et le composant TAL. De même, lors de la création de schémas tutoriaux, l'utilisateur professeur peut modifier les préconditions et argument des plans et actions les plans et actions. Ceci permet la création de séquences interactives dans le micromonde.Ce projet, instancié pour l'apprentissage de l'anglais en tant que langue seconde pour des adultes, fut expérimenté avec succès dans une classe d'apprentissage de l'anglais pour adultes. Les apprenants, travaillant en binôme ou triade, devaient réaliser les trois types d'activité à la suite. Les résultats prouvèrent la robustesse du système et aussi l'intérêt de la démarche vu l'adhésion remportée auprès des apprenants.

Les derniers questionnements autour de ce projet tournent autour de la constitution d'une taxonomie d'erreurs en vue de trouver la « meilleure » façon d'y répondre et sur les problèmes de caractérisations des interactions en cours [Reeder, 2000]. Le problème de la caractérisation des interactions et de leur organisation fait l'objet d'une discussion entre l'analyse conversationnelle et le DISCOUNT pour comprendre comment caractériser les interactions dans une situation d'apprentissage. L'analyse conversationnelle décrit l'interaction comme une séquence de paires d'actes de langage [Coulthard, 1985]. Les séquences sont caractérisées par la nature des actes de langage (par exemple, la séquence initiative/réactive/reprise). DISCOUNT décrit et évalue les dialogues éducatifs, plus spécialement les dialogues entre un apprenant et un tuteur. Grâce à DISCOUNT, des séquences des interactions peuvent être marquées. Les marqueurs posent les rôles des apprenants et du tuteur dans cette séquence [Pilkington, 1999]. Ici s'arrêtent les questionnements autour de FLUENT, Reeder trouvant l'analyse conversationnelle trop descriptive et et difficile à concilier avec SCHEME.

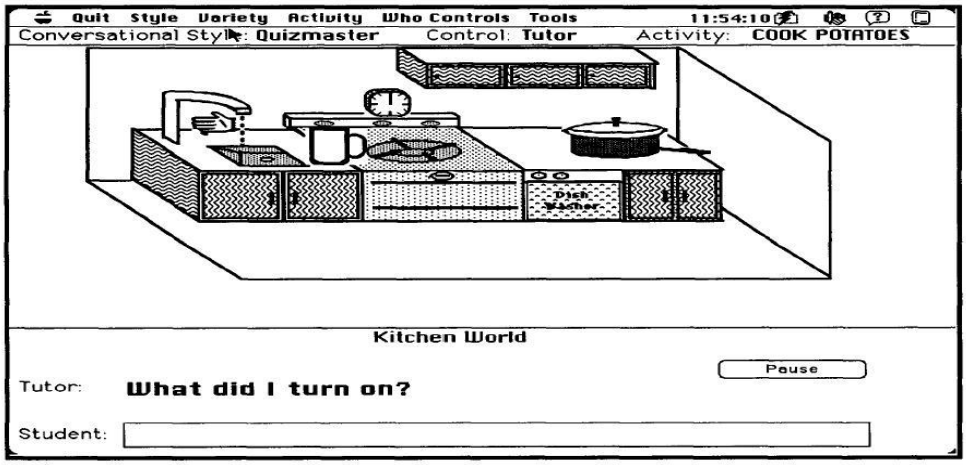


Figure 3 : L'environnement FLUENT, extrait de [Schoelles *et al.*, 1997]

Le second environnement auquel il est intéressant de se référer est l'environnement MILT (Military Language Tutor) développé par l'armée américaine tout au long des années 1990 [Holland *et al.*, 1999]. Cet environnement propose des activités de communication autour d'activités que peuvent réaliser des soldats en mission dans un pays étranger. Les apprenants sont aussi immergés dans un micromonde et peuvent soit parler soit écrire uniquement dans la langue cible à l'intention d'un agent animé (le compagnon) qui accomplit les actions que lui dicte l'apprenant. Ces activités se centrent autour d'exploration de pièces dans lesquels l'apprenant doit découvrir des documents et les consulter. La langue cible est l'arabe moderne. Les instructeurs peuvent créer leurs scénarii en les décrivant dans un environnement auteur, où ils peuvent aussi modifier l'apparence du micromonde (réarrangement des objets du micromonde, redéfinition des écrits des lettres, livres et cartes, changement du décor). La définition du composant de traitement automatique de la langue naturelle leur posa un grand problème et ils durent se résoudre à une approche par reconnaissance de mots-clés dans les énoncés de l'apprenant, approche qui s'est révélée suffisante au cours de leurs tests. Ils simplifièrent même l'interaction en allant jusqu'à proposer un ensemble de phrases déjà construites pour une situation d'un scénario donné (ce qui donna par la suite leurs premières idées pour une implémentation de techniques de reconnaissance vocale).

Le troisième environnement significativement proche de nos travaux est l'environnement développé par Boylan et Micarelli [Boylan *et al.*, 1998]. Ces chercheurs se basent sur l'approche culturo-communicative définie par eux-mêmes. La caractéristique

communicative se retrouve dans l'interaction des activités où les actions langagières de l'apprenant sont dirigées intentionnellement vers un locuteur étranger (réel ou sensé l'être). La caractéristique culturelle est le fait de comprendre le contexte culturel dans la communication, les usages de son interlocuteur sont à respecter pour que l'interaction puisse aboutir. Ils prennent comme exercice de référence la reconstruction de conversation, le principe étant de deviner la suite d'une conversation qui est lancée. Dans ce système, l'étudiant dialogue dans une situation simulée avec un interlocuteur sous la surveillance d'un tuteur virtuel qui conseille l'apprenant dans sa langue maternelle. L'étudiant écrit des phrases à l'intention d'un personnage. La situation de communication est, par exemple, la classique scène où l'apprenant s'adresse à un commerçant. L'apprenant est représenté par un avatar et peut inscrire ses phrases dans une bulle. Le personnage fait de même. Le tuteur renseigne l'apprenant sur les tournures à employer vis à vis de la situation et de l'interlocuteur, le ressort pragmatique est la correction du langage vis à vis de son interlocuteur. Les deux auteurs insistent sur le fait que de nombreux étudiants trouveraient intrinsèquement plus intéressant que l'usage de la langue se manifeste par des réalisations dans un monde réel ou virtuel. L'environnement est conçu autour d'un moteur d'intelligence artificielle qui gère l'interaction.

L'environnement TLTS (Tactical Language Training System) peut se présenter comme la somme de nombreuses recherches et constitue un projet de grande envergure, tant par ses prétentions scientifiques que par le nombre de chercheurs impliqués. Les idées des micromondes, des compétences culturelles, de la communication avec un partenaire virtuel, d'un agent pédagogique qui intervient dans la conversation, de la reconnaissance vocale y sont reprises et donnent naissance à un environnement original [Johnson *et al.*, 2004]. Ce système développé pour l'armée américaine se donne trois objectifs principaux. Le premier objectif est de permettre l'apprentissage de « langues rares », l'arabe libanais en l'occurrence, en impliquant l'apprenant dans des tâches permettant de se concentrer sur des échanges communicatifs particuliers. Ces échanges permettent d'accomplir le second objectif, celui d'immersion culturelle. L'apprenant communique avec des partenaires animés, communication passant par l'oral mais aussi par le choix de gestes de la part de l'apprenant à l'intention des personnages animés. Le dernier objectif est de combiner un retour sur la forme auprès de l'apprenant et une pratique « authentique » de la langue, ceci est fait au travers de la présence dans les scènes d'un agent animé à mission pédagogique qui peut aider l'apprenant dans ses activités. L'environnement intègre de nombreux composants. Il existe ainsi un outil

auteur pour définir les scènes. Il y a une représentation de l'agent pédagogique qui possède lui-même une représentation de l'apprenant. Un composant langue naturelle qui inclut une partie reconnaissance vocale, une partie traitement automatique de la langue naturelle, une partie d'analyse des erreurs. Les tâches proposées dans TLTS se rapprochent des tâches que les soldats auront sur le terrain : par exemple, des missions de reconstruction de villages où ils sont tenus de communiquer avec les habitants. Le système a été testé avec succès et un développement avec l'arabe iraquien comme langue cible est prévu.

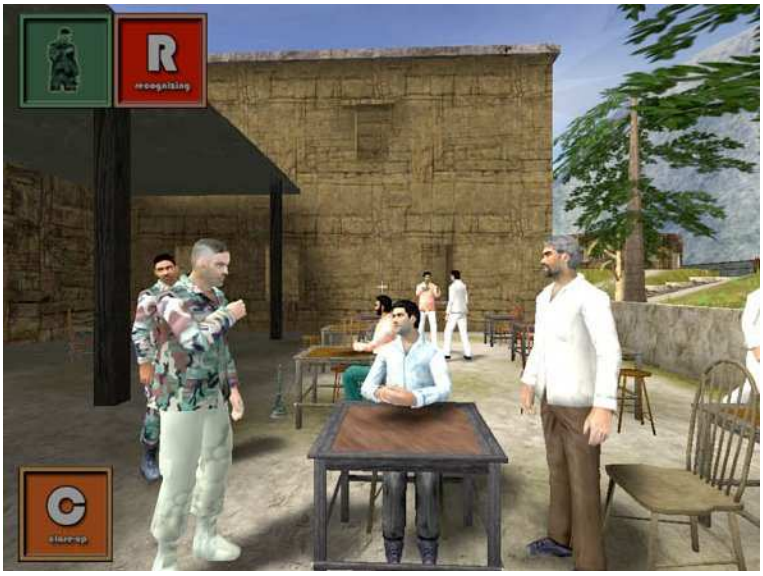


Figure 4 : L'environnement TLTS, extrait de [Johnson *et al.*, 2004]

1.5 Nos propositions

Les systèmes présentés dans la section précédente s'attachent aux question suivantes : quelles tâches pédagogiques donner aux apprenants, quelles interactions concevoir pour effectuer les activités interactives, comment réagir aux énoncés et actions des apprenants. Peu de ces travaux mentionnent les théories en acquisition en langue pour caractériser les interactions possibles dans ces environnements. Notre positionnement est de reprendre ces théories et d'étudier comment les adapter dans le contexte de micromonde d'apprentissage en langue.

Le problème posé est la création de simulations énonciatives qui puissent permettre d'acquérir en langue. L'hypothèse pédagogique de la situation est que la conversation avec un agent virtuel dans une langue est susceptible de créer des situations dans lesquelles

l'apprenant puisse non seulement user de la langue mais aussi acquérir des connaissances sur cette langue, fussent elles lexicales ou syntaxiques. Notre proposition est d'impliquer l'apprenant dans une tâche pratique (la réalisation d'une recette), en interaction avec un partenaire virtuel (un chef) au sein d'un monde virtuel (une cuisine), tâche prétexte à l'usage et l'éventuelle acquisition de la langue. L'hypothèse théorique sous-jacente est que l'interaction, générée par la réalisation de la tâche, va créer des séquences potentiellement acquisitionnelles [Demaiziere *et al.*, 2005]. Le partenaire réagit aux énoncés corrects ou incorrects de l'apprenant, poursuivant la tâche ou s'engageant à discuter d'éventuels points problématiques des énoncés de l'apprenant. Les réactions aux énoncés incorrects se font grâce à des stratégies de rétroaction, en fonction des éléments reconnus dans les énoncés de l'apprenant; ces stratégies sont pensées pour continuer l'interaction autour de la tâche.

Nous voyons ici l'émergence de trois dimensions selon lesquelles vont s'organiser la conception de l'environnement : tâche (quelle est la tâche dans laquelle l'apprenant est impliqué), interaction (quelles sont les interactions qui président à l'accomplissement de cette tâche), et langue (quelle sont les formes lexicales comme syntaxiques que prend la langue utilisée dans un tel contexte).

- *Une tâche pédagogique impliquant l'usage de la langue selon des types d'interaction dans un micromonde*

Nous avons retenu l'approche micromonde qui permet de contextualiser l'interaction entre l'apprenant et le système. La tâche est alors à définir en trois autres niveaux : le niveau pédagogique, le niveau interactif et le niveau applicatif. Le niveau pédagogique s'intéresse à comment est mise en jeu la langue dans une optique d'apprentissage et sur quelles théories et concepts cette mise en jeu s'appuie. Le niveau interactif définit les interactions qui prennent place dans le micromonde, tant par leurs modalités que par leur structuration. Le niveau applicatif est défini par les actions de réalisation de la recette. Il permet de caractériser le micromonde et les formes du langage naturel utilisées.

- *Un modèle d'interaction inscrit dans une activité dont la langue est l'une des modalités*

Ce modèle décrit l'articulation entre les différents types d'interaction au cours d'une même activité et rend compte de la dynamique de l'interaction. Les types d'interaction permettent de caractériser les échanges possibles dans une activité mais de façon statique. Lorsqu'il s'agit de gérer dynamiquement l'interaction, il faut croiser les différents types d'interactions en un même modèle. La modélisation et l'organisation de l'interaction repose sur les travaux accomplis en dialogue homme-machine et sur les théories et concepts concernant le problème de l'interaction dans les EIAH.

- *Un modèle de la langue support à l'interaction dans le cadre d'une tâche qui délimite le lexique et la syntaxe*

La tâche applicative délimite le lexique et la syntaxe qui sont utilisés dans l'accomplissement de la tâche interactive. Ces lexique et syntaxe permettent le traitement automatique de la langue naturelle (TALN), tant pour l'analyse des énoncés de l'apprenant que pour la génération des interventions verbales et non-verbales du partenaire, permettant la continuation de l'interaction. Ce modèle s'inscrit dans le champ du TALN et en particulier le TALN en tant que support à l'apprentissage des langues.

À partir de ces trois modèles imbriqués, nous avons construit un environnement informatique mettant en place une simulation énonciative. L'activité mise en place est une activité de production d'énoncés de la part de l'apprenant à l'adresse du partenaire. L'apprenant dicte une recette au partenaire, lequel réagit aux énoncés de l'apprenant suivant la correction linguistique et la pertinence de l'énoncé dans le déroulement de l'activité.

1.6 Méthodologie

1.6.1 Démarche et historique

Notre cheminement est parti d'une étude des travaux théoriques à la fois en apprentissage des langues, en dialogue homme-machine, en interaction homme machine et en

conception des EIAH. Ces travaux permirent l'élaboration de premiers modèles d'activité et d'interaction. Une expérience de récupération de corpus d'énoncés d'apprenants fut aussi menée pour identifier les erreurs possibles des apprenants. Un site web fut créé[2]. L'activité proposée aux apprenants était de traduire les consignes de la recette de leur langue vers le français (une page web contenait l'énoncé de la consigne dans leur langue, une photographie montrant l'action, et un champ de texte où ils pouvaient entrer leur traduction). Ceci a permis d'obtenir un corpus de plus de 700 énoncés [Lehuen *et al.*, 2002]. Ces énoncés ont permis d'avoir une idée des erreurs des apprenants et d'avoir des pistes pour l'analyse des énoncés à mettre en place. Venant de travaux annexes [Delorme, 2002] et reprenant en partie nos modèles, deux modèles simplifiés de représentation de la tâche applicative et de représentation de l'interaction (sans prise en compte des problèmes de langue) et un moteur de dialogue furent conçus. Une première réalisation de l'environnement fut alors effectuée intégrant alors une interface graphique et un moteur d'interaction permettant la réalisation de la tâche applicative. Le modèle du lexique fut créé à partir de recettes et la conception de l'analyseur s'appuya sur le corpus des énoncés. Ensuite ce premier environnement donna lieu à de premiers tests, lesquels soulevèrent des problèmes de type informatique sans vraiment remettre en cause les modèles créés. Ce travail n'est évidemment pas aussi séquentiel : l'étude des travaux théoriques et le raffinement des modèles se poursuivirent au cours des phases même de mise en oeuvre.

1.6.2 Mises à l'essai

Les mises à l'essai ont eu lieu en décembre 2004 et se sont poursuivies en 2005 et 2006 auprès d'étudiants thaïs et chinois qui suivaient à l'Université du Maine une première année d'intégration. Cette année d'intégration leur permet d'apprendre le français. Les premiers tests nous permirent d'effectuer des tests d'utilisabilité et aussi de recueillir les impressions des apprenants sur le prototype et son fonctionnement. Ces tests mirent en évidence des problèmes de programmation de certains modules (analyse, génération, tâche) et le côté stéréotypé des énoncés du partenaire. Les apprenants ont souligné des problèmes liés à l'interface (aspect du compagnon, éléments de la cuisine peu reconnaissables) et à la tâche qui leur était proposée (manque de repères au cours de l'exécution, désir de plus de possibilités d'erreurs dans la cuisine).

[2] http://www-lium.univ-lemans.fr/~lehuen/webclips/presampras/

1.7 Synthèse

Ce premier chapitre expose la motivation du projet, ses racines, ses sources et son déroulement. Une première partie expose les racines du projet au croisement de deux champs d'études en informatique que sont l'EIAH et le dialogue homme-machine. Une seconde partie présente le cadre théorique dans lequel nous nous situons : travaux en didactique des langues, travaux en conception des EIAH langues. Une troisième partie expose des environnements proches du nôtre de par leurs idées et leurs conceptions. Une quatrième partie présente nos propositions de modèles qui caractérisent trois dimensions de conception de l'environnement à mettre en place. Le chapitre s'achève sur les considérations méthodologiques et sur les mises à l'essai de cet environnement.

Chapitre 2
Modélisations

2.1 TACHE 43
2.1.1 TACHES PEDAGOGIQUES 43
2.1.1.1 Intention didactique 44
2.1.1.2 Tâches pédagogiques dans notre environnement 46
2.1.2 TACHE APPLICATIVE 47
2.1.2.1 Choix de la tâche applicative 48
2.1.2.2 Définition des éléments de l'interface du monde et constitution du lexique 48
2.1.2.3 Description de la tâche applicative 49
2.1.3 TACHE INTERACTIVE 52
2.1.3.1 Possibilités interactives dans le micromonde 53
2.1.3.2 Activité de compréhension 55
2.1.3.3 Activité de production 57
2.2 INTERACTION 60
2.2.1 DU CONCEPT D'INTERACTION DANS LES EIAH 60
2.2.1.1 Dialogue homme-machine et tuteurs intelligents 61
2.2.1.2 Dialogue homme-machine et apprentissage des langues 62
2.2.2 TYPES D'INTERACTION 64
2.2.2.1 Nature des échanges 64
2.2.2.2 Description des interactions 64
2.2.2.3 Types d'interaction au cours de l'activité de compréhension 65
2.2.2.4 Types d'interaction au cours de l'activité de production 68
2.2.3 MODELE D'INTERACTION 70
2.2.3.1 Représentation des échanges 70
2.2.3.2 Déroulement des échanges 79
2.3 LANGUE 86
2.3.1 TRAITEMENT AUTOMATIQUE DES LANGUES ET APPRENTISSAGE DES LANGUES ASSISTE PAR ORDINATEUR 86
2.3.2 ANALYSE ET GENERATION EN CONTEXTE 89
2.3.2.1 Rétroaction 89
2.3.2.2 Analyse 90
2.3.2.3 Génération de la réponse 92
2.3.3 LEXIQUE 93
2.3.4 UNE ANALYSE, SUPPORT A L'INTERACTION : COMMENT ? 95
2.3.4.1 Observation de l'énoncé 96
2.3.4.2 Comparaison des énoncés à la description de la tâche 106
2.3.4.3 Génération 109
2.4 SYNTHESE 112

Ce chapitre présente les modélisations qui soutiennent l'environnement Sampras. Le chapitre porte sur l'établissement des modèles de tâche, d'interaction et de la langue. Il présente le passage d'une description d'intentions en termes de concepts et théories didactiques à une description en termes de modèles informatisables. Trois dimensions caractérisent la modélisation de cette activité pédagogique en vue de sa réification : la tâche, l'interaction et le langage. La description de la dimension tâche recouvre la description des tâches pédagogiques proposées aux apprenants, des tâches interactives créées à partir des spécifications des tâches pédagogiques et de la tâche applicative dont la résolution sert de prétexte à l'interaction. La description de la dimension interaction présente les types d'interaction attendus entre l'apprenant et son partenaire au cours des activités pédagogiques et le modèle d'interaction qui articule ces types d'interaction. La présentation de la dimension langue décrit le modèle du lexique, les mécanismes d'analyse des énoncés de l'apprenant et de génération des réactions du partenaire.

Trois dimensions de lecture

La figure suivante représente les liens entre les trois dimensions définissant la conception de notre environnement. Cette vision conceptuelle de l'environnement articule trois pôles complémentaires et interdépendants (Figure 5).

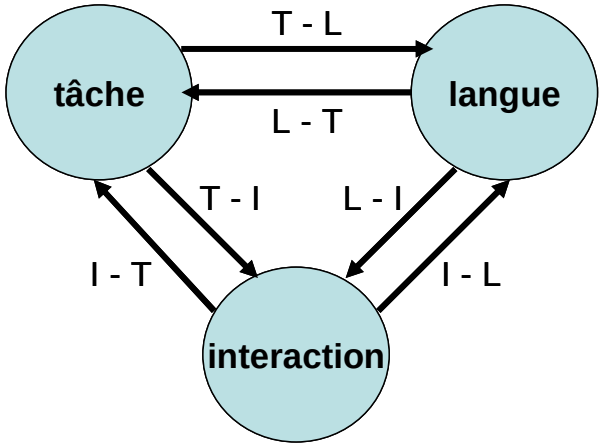


Figure 5 : Tâche, interaction et langue

Les trois pôles, présentés Figure 5, se déclinent ainsi. Les pôles étant interdépendants, leurs définitions se croisent.

- Le pôle tâche :
  - tâche pédagogique : Partant de l'approche communicative, les tâches pédagogiques visent à impliquer un apprenant dans une interaction avec un partenaire virtuel. Cette interaction est l'accomplissement d'une tâche applicative selon des types interaction. Les types d'interaction dépendent des compétences en langue à exercer.
  - tâche applicative : la tâche applicative est la tâche pratique prétexte à l'interaction. Elle organise l'interface du micromonde et nécessite une description informatique précise pour permettre de suivre son avancement.
  - tâche interactive : les tâches interactives sont les instanciations informatiques des activités pédagogiques. Elles permettent la progression de l'accomplissement de la tâche interactive dans le micromonde par la communication entre l'apprenant et le partenaire. Les interactions comprennent celles utilisées pour accomplir la tâche applicative selon les compétences en jeu (compréhension ou production) et celles liées aux rétroactions à donner à l'apprenant.

- Le pôle interaction décline deux aspects :
  - représentation des interactions : à partir des éléments en acquisition des langues et des tâches pédagogiques créées, des types d'interaction ont été imaginés.
  - gestion de l'interaction : en fonction des énoncés et actions des apprenants comme du partenaire, des modèles d'interaction articulent l'ensemble des types d'interaction au sein de l'activité de production.

- Le pôle langue :
  - conception du lexique : à partir de la tâche applicative et en fonction de l'analyse à effectuer, le lexique est modélisé.
  - mécanisme d'analyse et génération : à partir des besoins en support de l'interaction, un modèle d'analyse et de génération a été conçu.

Organisation du chapitre

Les trois dimensions se déclinent selon trois points de vue : théorisation, modélisation et réification. Ainsi la définition des différents pôles s'appuie sur des théories qui permettent la création d'un modèle, lequel se réifie dans l'environnement informatique.

Dans ce chapitre, les dimensions théorie et modélisation sont présentées pour chaque pôle, la partie mise en oeuvre est abordée dans le chapitre environnement. En dépit du fait que ces trois pôles soient fondamentalement interdépendants, nous allons néanmoins les présenter sous la forme de trois parties qui vont chacune d'elle privilégier un point de vue tout en évoquant les autres. Ce parti pris implique certaines redondances mais permet trois entrées pour comprendre nos modèles. Le chapitre est donc divisée en trois parties : la première présente la dimension tâche, la seconde explicite la dimension interaction et la dernière présente la dimension langue.

2.1 Tâche

2.1.1 TACHES PEDAGOGIQUES 43
2.1.1.1 Intention didactique 44
2.1.1.2 Tâches pédagogiques dans notre environnement 46
2.1.2 TACHE APPLICATIVE 47
2.1.2.1 Choix de la tâche applicative 48
2.1.2.2 Définition des éléments de l'interface du monde et constitution du lexique 48
2.1.2.3 Description de la tâche applicative 49
2.1.3 TACHE INTERACTIVE 52
2.1.3.1 Possibilités interactives dans le micromonde 53
2.1.3.2 Activité de compréhension 55
2.1.3.3 Activité de production 57

Dans la création d'environnements informatiques pour l'apprentissage humain, la notion de tâche est centrale. La difficulté à aborder cette notion dans le cadre de notre étude est due au fait que la notion de tâche s'applique à trois niveaux de conceptualisation distincts mais interdépendants. Le premier niveau représente **les tâches pédagogiques** : nous décrivons l'intention didactique qui motive le projet, puis les tâches pédagogiques effectivement imaginées. Le second niveau de conceptualisation est **la tâche applicative**, c'est à dire la description de la tâche choisie pour être le prétexte à le communication entre l'apprenant et son partenaire virtuel. Le dernier niveau décrit **les tâches interactives**, cette dimension permet la création des modèles des activités à effectivement réaliser par les apprenants.

2.1.1 Tâches pédagogiques

Les tâches pédagogiques sont motivées par une intention didactique prélude à la conception de l'environnement.

2.1.1.1 Intention didactique

"*Un EIAH embarque une intention didactique et/ou pédagogique*" [Tchounikine *et al.*, 2004]. Cette phrase soulève le difficile problème de la définition de **l'intention didactique** qui précède la conception d'un artefact informatique à but éducatif. Si la genèse de ce projet est l'utilisation (ou du moins l'inspiration) de techniques de dialogue homme-machine pour concevoir un environnement informatique pour l'apprentissage des langues, la mise en place de cette utilisation ne peut se décrire sans recours à la didactique des langues. La définition et la modélisation de l'intention didactique permettent la création de modèles qui se réifieront en un environnement. La première piste qui s'offre à nous lors de la définition et la modélisation de ces intentions est de **transposer des modèles didactiques et pédagogiques** préexistants. Notre premier travail est donc d'identifier les approches et théories qui sont appropriées et d'en tenter la transposition.

Notre idée de départ est l'inscription de nos activités dans **l'approche communicative**. Cette approche, prônant l'apprentissage d'une langue étrangère au cours de la communication en cette même langue, pose comme exercice principal **la simulation** [Puren, 2001]. Comment concevoir de fait une telle simulation et comment l'organiser pour que celle-ci permette l'usage et l'apprentissage d'une langue ? Si nous reprenons les éléments fournis par le cadre théorique de notre projet concernant l'approche communicative (§1.3.1), l'apprenant se trouve engagé dans une **situation de communication plausible** par la réalisation d'une tâche, l'accent étant mis davantage sur l'accomplissement de cette tâche et moins sur la correction des énoncés et actions au regard de la langue cible. La tâche impliquant l'usage de la langue, proposée à la réalisation par l'apprenant, possède donc un but ou un état d'interaction à atteindre : c'est **une tâche finalisée**. Les tâches langagières possédant cette caractéristique sont nombreuses et la plus classique d'entre elle est celle de l'apprenant s'adressant à un vendeur dans le but d'acquérir un bien, de jouir d'un service, d'obtenir des informations (scènes que l'on retrouve dans des environnements informatiques tels que Mepa [Lehuen *et al*, 2006] ou TLTS [Johnson *et al.*, 2004] par exemple).

Notre proposition d'impliquer un apprenant ou un groupe d'apprenants travaillant en binôme ou triade, (lors des tests, ce fut des apprenants travaillant en binôme) dans une activité à réaliser en interagissant dans un **micromonde** avec un partenaire virtuel, les interactions langagières étant uniquement dans la langue cible, s'inscrit dans cette approche. Le principe

premier de la démarche micromonde est l'immersion ou du moins l'implication de l'apprenant dans un monde où ses énoncés et actions ont des conséquences appréhendables. Cette approche, dérivée du célèbre Logo de Papert [Papert, 1981], met en jeu **la propre construction de son savoir** par l'apprenant en interagissant avec les éléments d'un monde. Si, dans le cas des micromondes en mathématiques, il s'agit de formes abstraites à retrouver (telles que des considérations géométriques par exemple [Bellemain, 1992]), l'usage du concept de micromonde en apprentissage des langues est fondé sur l'usage et la (re)découverte de la langue en situation de communication simulée avec un partenaire virtuel.

Le postulat qui accompagne la création des tâches pédagogiques inspirées de l'approche communicative est qu'au coeur de l'interaction prend place **l'acquisition de la langue**. Comme nous l'avons vu précédemment (§ 1.3.1), l'acquisition en langue est vue comme un processus quasi inconscient dont des manifestations sont visibles dans l'évènement langagier, une forme d'apprentissage implicite. Les autres environnements de type micromondes pour l'apprentissage des langues, présentés § 1.4, s'appuient aussi sur une approche comparable. Ainsi Schoelles évoque "*an approach, [...]designed to foster implicit learning of language* "[Schoelles *et al.*, 1997], et Holland mentionne "*implicit feedback, intrinsic reward* "[Holland *et al.*, 1999]. Ces tâches pédagogiques communicatives reposent sur l'hypothèse que certaines interactions, qui y ont lieu, sont génératrices de **séquences potentiellement acquisitionnelles**. La structure de la communication s'en trouve modifiée par l'émergence de questionnements autour du sens d'un évènement langagier (qu'il soit énoncé ou action) au cours de l'activité. Le fil de la conversation est alors modifié, le phénomène de négociation de sens prend place. Ces séquences incluent des phénomènes de *noticing* où l'apprenant remarque ou est amené à remarquer ses propres erreurs. La transposition informatique de ces concepts, mentionnés par Chapelle [Chapelle, 1998], se fait en englobant ces possibilités dans l'organisation des interactions de l'activité. Ainsi la structure de l'interaction entre l'apprenant et son partenaire doit pouvoir être modifiable en fonction des problèmes se posant au cours de cette interaction pour permettre l'acquisition en langue.

L'intention didactique du projet Sampras est de faire pratiquer une langue étrangère dans un contexte d'interaction. Cette intention s'inscrit dans l'approche communicative et s'enracine dans un apprentissage implicite ou acquisition d'une langue. Les micromondes offrent un cadre intéressant pour l'expression de cette intention.

2.1.1.2 Tâches pédagogiques dans notre environnement

Les tâches pédagogiques sont les « exercices » dans lesquels l'apprenant va être impliqué. Elles se caractérisent par les compétences en langue que l'on voudrait voir les apprenants exercer, et la situation d'interaction qu'est l'accomplissement d'une tâche applicative dans un micromonde. La tâche applicative sert de prétexte à l'interaction. La conjonction entre les compétences à exercer et la tâche applicative définit les interactions qui peuvent avoir lieu au cours des activités et par là même le lexique et la syntaxe des interventions de l'apprenant comme du partenaire. Deux compétences se dégagent clairement en apprentissage des langues : la compréhension et la production [Chapelle, 2000]. La compréhension est la capacité de l'apprenant à interpréter ce que son partenaire veut signifier. La production est la capacité de l'apprenant à s'exprimer dans la langue cible. Le choix de la tâche applicative a été celui de la réalisation d'une recette de cuisine. Cette tâche finalisée délimite un lexique et des structures syntaxiques bien particulières [Michel *et al.*, 2004] qui influent sur la modélisation du micromonde et sur celles des possibilités interactives.

Les activités à accomplir se présentent donc comme la réalisation d'une recette de cuisine dans un micromonde. Cette réalisation est faite en interaction avec un partenaire. Les interactions qui ont lieu sont caractérisées par la compétence à exercer au cours de l'activité. Dans le cas de la **compétence de compréhension**, l'activité consiste pour l'apprenant à effectuer la recette sous la direction du partenaire, il manipule les objets du micromonde dans le but de réaliser les instructions du partenaire. Dans le cas de la **compétence de production**, l'apprenant, partant d'une description sommaire de la recette (listes des objets et actions) tente de la dicter au partenaire sous la forme d'énoncés écrits à l'intention de ce dernier. Le partenaire manipule alors les objets du monde en fonction des énoncés de l'apprenant. Ici se dessinent les types d'interaction aux travers desquelles les activités sont réalisées.

Nous avons évoqué les travaux de Mangenot *[Mangenot, 2000]* qui indiquent les caractéristiques essentielles pour qu'une tâche soit profitable dans un environnement multimédia, nous pouvons comparer ces caractéristiques à notre approche. La première de ces caractéristiques et «*que la tâche part de données riches et authentiques (notamment linguistiques et civilisationnels* ». Le prétexte à l'interaction qu'est la réalisation d'une recette de cuisine fait intervenir plusieurs champs sémantiques (ustensiles, ingrédients, actions, déplacements, repérage spatial, *etc.*), et fait référence à un aspect identifiable de la culture

française. *« [la tâche] propose des activités d'un bon niveau cognitif (liens données/activités pertinents, situations problèmes, appel à la créativité) »*, le lien activité/données caractérise l'inscription des activités dans le monde de la cuisine. L'idée de dicter ou de faire dicter entraîne des manipulations d'objets et la création d'énoncés tournant autour de la réalisation de la recette. *«[la tâche] prévoit des interactions variées (notamment - mais pas exclusivement - évaluatrices) pendant et après l'exécution de la tâche. »* Ce point n'est pas complètement identifiable à notre échelle, ne connaissant pas exhaustivement les stratégies d'emploi que des enseignants sont susceptibles de faire de cet environnement. Néanmoins, les rétroactions données à l'apprenant pointent sur les problèmes linguistiques apparus au cours de l'interaction.

A partir des compétences à exercer et des caractéristiques de la tâche applicative, nous pouvons définir des activités interactives ayant les spécificités de notre approche. Les activités interactives reposent sur une tâche applicative qu'il s'agit de réaliser. La section suivante concerne la représentation et la gestion de la tâche applicative ainsi que les implications de cette tâche sur le design du micromonde.

2.1.2 Tâche applicative

La tâche applicative est le prétexte de l'interaction. Le système peut suivre l'accomplissement de cette tâche. Pour ce faire, le système nécessite une représentation de la tâche et un mécanisme de gestion pour qu'effectivement le suivi ait lieu en fonction des actions du partenaire comme de l'apprenant. Cette partie illustre le lien entre langage et activité (Figure 6).

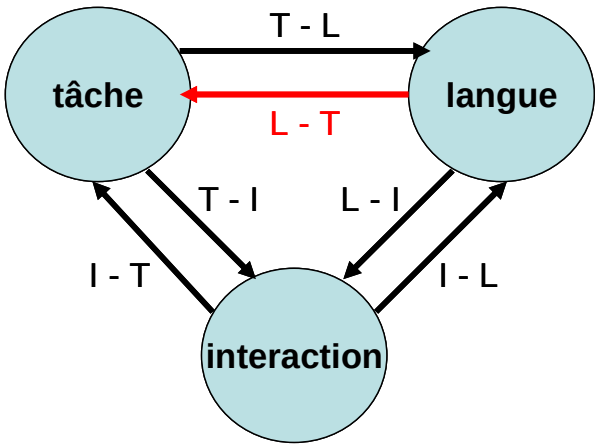


Figure 6 : Lien entre activité et langage

2.1.2.1 Choix de la tâche applicative

La tâche applicative est une tâche du monde réel dont il est possible de simuler la réalisation dans un micromonde. Cette tâche est finalisée, c'est à dire qu'elle part d'un état du micromonde pour atteindre un autre état du micromonde au travers d'étapes prévisibles. Ce choix répond à un impératif : **ne pas perdre le fil du dialogue**. Le monde clos circonscrit le lexique employé. La tâche se décompose en sous-tâches qui permettent de suivre la progression de la tâche jusqu'à atteindre le degré des actions effectuables dans le micromonde. Les tâches applicatives de ce type sont toutes celles qui suivent plus ou moins un "mode d'emploi" (une recette de cuisine, un objet à monter ou démonter, un élément à réparer). Elles se retrouvent souvent dans les environnements en 3D destinées à des apprentissages de routines de travail, comme dans le projet APLG [El Kechaï *et al.*, 2005], ou celui utilisant le compagnon Steve [Johnson *et al.*, 2000]. Mais ces environnements se consacrent à l'apprentissage même de la tâche applicative.

Dans notre étude, la tâche prétexte à l'interaction choisie est la réalisation d'une recette de cuisine. Ce choix est lié à la cuisine en tant qu'élément de la culture française et à la palette d'activités faisant intervenir plusieurs champs sémantiques (ustensiles, ingrédients, actions, déplacements, repérage spatial, *etc.)* que l'on peut exploiter à partir de cette tâche. La création de cette tâche a débuté par l'analyse de différentes recettes de gâteau au chocolat pour se fixer finalement sur une recette de flan au chocolat. À cette occasion, un site Internet[3] a été développé dans le but de constituer un corpus d'énoncés. Ce site a permis de recueillir environ 700 énoncés écrits par des apprenants hispanophones et germanophones. Ces énoncés sont les traductions de consignes, écrites en français, correspondant aux étapes de la réalisation d'une recette de gâteau au chocolat [Lehuen *et al.*, 2002]. Ce recueil d'énoncés était destiné à la définition de l'analyse des énoncés des apprenants dans Sampras.

2.1.2.2 Définition des éléments de l'interface du monde et constitution du lexique

Concernant la définition et l'organisation des éléments présents à l'interface du micromonde, ces éléments sont directement extraits du lexique employé dans la recette (voir annexe C). Ainsi nous identifions des ustensiles (balance, doseur), des ingrédients (beurre, sucre, chocolat, crème, farine), des récipients (moule, saladier), des appareils électro-

[3] http://www-lium.univ-lemans.fr/~lehuen/webclips/presampras/

ménagers (four), des rangements (placards, frigo). Après cette définition, le premier point d'organisation est celui concernant la répartition spatiale de ces éléments. Ainsi, il y a des éléments "dans" d'autres (exemple : des ingrédients dans un rangement), des éléments situés sur la gauche, au milieu, sur la droite, en haut, en bas (exemple : un placard en bas à gauche). Le second point d'organisation concerne les actions possibles avec ces éléments : ouverture de rangement, prise d'ingrédients ou d'ustensiles, pose d'ingrédients ou d'ustensiles, mise en fonctionnement d'appareils électroménagers, actions mises en scène sur l'interface de ce monde (par manipulation directe à la souris pour l'apprenant dans le cas de l'activité de compréhension, par réalisation directe à l'interface pour le partenaire dans le cas de l'activité de production).

La définition des éléments et des actions possibles à l'interface pose la question de la constitution du lexique mais cette fois-ci du point de vue de sa représentation machine en vue de traitement automatique. Le premier niveau de constitution du lexique employé est d'identifier, en plus de leur identité et placement, les liens entre les objets. Nous trouvons alors les notions de contenu (un ingrédient, par exemple la crème) et contenant (pot). Les structures des types d'énoncés employés dans l'activité sont les assertions, les ordres et les questions. Nous obtenons alors les catégories lexicales suivantes : ingrédient, récipient, localisateur, électroménager, contenant, contenu, ustensile, rangement, attribut, action, ordre, question. Ces catégories sont utilisées pour définir les motifs syntaxiques utiles au traitement automatique des énoncés (décrit au paragraphe 2.3.3).

La décomposition de la tâche applicative en sous-tâches permet la gestion par le système informatique.

2.1.2.3 Description de la tâche applicative

Les tâches applicatives dans un micromonde tel que le nôtre, c'est à dire des tâches finalisées, sont modélisées pour concevoir une représentation en machine qui permettent leur gestion efficace. Le partenaire doit savoir où en est l'avancement de la tâche, qui est le contexte d'interprétation des actions de l'apprenant. Le formalisme doit s'adapter au type de tâches que nous avons choisi [Delorme, 2002], permettant de passer de la représentation généralement linéaire des tâches applicatives à une représentation machine. Le travail à réaliser concerne la description de la tâche applicative et sa réalisation. Le contexte dans

lequel s'opère cette réalisation (c'est à dire les activités interactives proposées à l'apprenant) relève de l'interaction. Le modèle créé permet de connaître quelles tâches sont réalisées, réalisables ou à venir. Nous ne détaillons pas les travaux de DEA de Fabien Delorme mais retenons les éléments importants de ce travail pour la compréhension de la conception de l'environnement. Le cadre conceptuel utilisé pour décrire la tâche applicative est le paradigme « tâche-méthode » [Trichet *et al.*, 1999]. Ce paradigme, issu des recherches sur les systèmes à base de connaissance, se propose de modéliser la résolution de problème en représentant la méthode de résolution du problème grâce à des tâches et des méthodes. La différenciation entre les tâches et les méthodes tient dans la différence entre le quoi et le comment. Les tâches désignent ce qui va être fait et les méthodes comment cela va être fait. A une tâche, une ou plusieurs méthodes permettant de l'effectuer peuvent être adjointes.

Pour la tâche applicative, les multiples tâches et sous-tâches sont représentées par cinq champs. Un champ Identificateur permet de repérer la tâche dans l'arbre des tâches. Un champ Texte correspond à l'instruction telle qu'elle serait énoncée. Un champ Définition caractérise la tâche par un motif syntaxique issu du lexique. Le champ Post-conditions donne les éléments qui changent dans le micromonde et les modifications de l'état de la tâche applicative. Le dernier champ, celui des Méthodes, énumère les méthodes qui réalisent la tâche. Ces champs sont présentés sur le Tableau 1. Des exemples de ces tâches et méthodes sont détaillés dans la section concernant l'environnement informatique.

Champ	***Description succincte***
Identificateur	Nom donné à la tâche
Texte	Énoncé qui correspond à la tâche
Définition	Définition de la tâche par les éléments du lexique
Post-conditions	Effets de la tâche, nouvelles réalités de l'activité après réalisation de la tâche
Méthodes	Les méthodes qui permettent de réaliser cette tâche

Tableau 1 : Tâche de l'arbre des tâches

Les tâches de l'arbre des tâches n'ont pas de pré-conditions. Une tâche peut être réalisée par différentes méthodes. Les pré-conditions portent sur la pertinence de ces méthodes dites opérationnelles. Il existe deux types de méthodes : les méthodes de décomposition qui génèrent des buts (c'est à dire des sous-tâches à accomplir), et les

méthodes opérationnelles qui génèrent des opérations (des actions sur les objets du monde dans notre cas). Les méthodes de décomposition permettent de diviser une tâche en sous-tâches. Elles se caractérisent par trois champs : un champ d'identification de la méthode, un champ contexte d'activation listant un certain nombre de faits à vérifier permettant d'élire cette méthode parmi les autres méthodes qui pourraient accomplir la tâche, et un champ tâches décrivant les tâches à réaliser pour que la tâche mère dont la méthode dépend soit accomplie (Tableau 2).

Champ	***Description succincte***
Identificateur	Nom donné à la méthode
Contexte d'activation	Liste des faits permettant la sélection de cette méthode par rapport à ses concurrentes si elles existent
Tâches	Liste des tâches devant être exécutées pour que la méthode soit déclarée accomplie

Tableau 2 : Méthode de décomposition

Les méthodes opérationnelles permettent l'exécution effective d'une tâche (Tableau 3). Elles possèdent quatre descripteurs : un champ d'identification de la méthode, un champ contexte d'activation listant un certain nombre de faits à vérifier permettant d'élire cette méthode parmi les autres méthodes qui pourraient accomplir la tâche, un certain nombre de pré-conditions, et une opération qui représente ce qui sera effectivement exécuté à l'interface (manipulation des objets mais aussi création d'une partie des énoncés du partenaire).

Champ	***Description succinte***
Identificateur	Nom donné à la méthode
Contexte d'activation	Liste des faits permettant la sélection de cette méthode par rapport à ses concurrentes si elles existent
Pré-conditions	liste des pré-conditions devant être vérifiées pour que la méthode soit exécutée
Opération	identificateur de l'opération qui va être exécutée et ses arguments

Tableau 3 : Méthode opérationnelle

Ce modèle permet de décomposer une tâche en sous-tâches selon différentes méthodes, et ainsi de générer dynamiquement et de façon contextuelle des arbres de tâches. Le modèle tâche-méthode permet la construction dynamique d'arbres comme celui de la Figure 3.

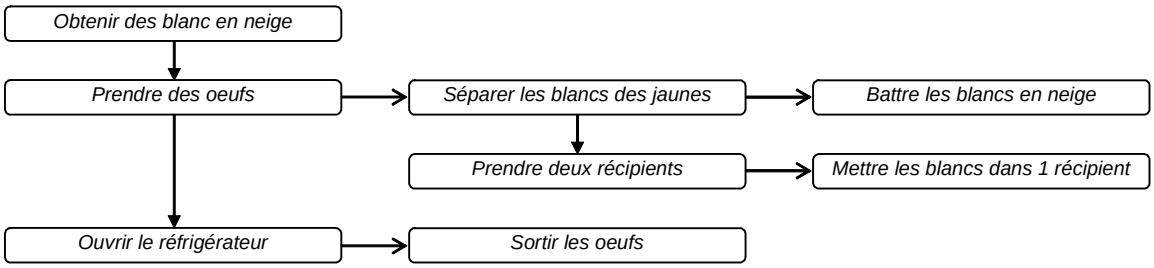


Figure 7 : Exemple d'arbre de tâches

Grâce à ce modèle, le système est en mesure, non seulement de contrôler l'évolution de l'activité, mais aussi d'interpréter les actions (données sous forme verbale ou non verbale) de l'apprenant en fonction de la tâche et/ou des consignes qui lui sont données.

Le paragraphe suivant s'attache à la définition de l'organisation des interactions qui peuvent alors lieu lors de ces activités.

2.1.3 Tâche interactive

La définition de la tâche interactive prend en compte toutes les interactions qui peuvent survenir au cours de la réalisation de la tâche applicative selon la compétence à exercer. La tâche pédagogique permet de caractériser les interactions qui ont lieu au cours de l'activité (le lien T-I, flèche rouge sur la Figure 8). Plusieurs types d'interaction prennent place dans une activité. Le type principal est celui qui fait progresser l'activité. D'autres types d'interaction liés au *feedback* donné à l'apprenant en cas d'erreurs ou aux questions que se pose l'apprenant co-existent avec celui-ci et permettent les séquences potentiellement acquisitionnelles. Deux activités interactives ont été conçues, l'une pour l'entraînement de la compétence de compréhension, l'autre pour celle de production.

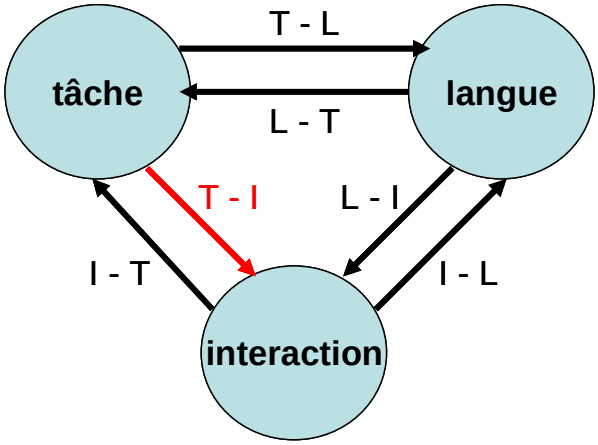


Figure 8 : Lien tâche et interaction

2.1.3.1 Possibilités interactives dans le micromonde

La question des modalités de communication dans un environnement micromonde pour les langues pose débat, surtout celle de la dimension orale et/ou écrite de la partie de la communication se déroulant en langue. Ainsi dans les environnements MILT [Holland *et al.*, 1999] et TLTS [Johnson *et al.*, 2004], la modalité orale seule a été choisie. Dans MILT, l'apprenant prononce des phrases qui lui sont présentées dans le micromonde à l'intention du système, le problème de reconnaissance est réglé car les phrases à reconnaître sont déjà connues. La communication entre l'apprenant et le système au travers d'énoncés écrits s'est heurtée à des problèmes de robustesse de l'analyse. Dans TLTS, la marge d'utilisation de la possibilité orale est plus grande : à chaque scène correspond un ensemble de phrases qui pourraient être prononcées et ces phrases sont attendues par le partenaire. Au bout de plusieurs essais infructueux de la part de l'apprenant, l'aide suggère un énoncé. Dans l'autre sens, c'est à dire du système vers l'apprenant dans le micromonde, les personnages des scènes sont dotés de la parole et communiquent ainsi avec l'apprenant. La modalité écrite n'existe pas du tout dans TLTS. Dans FLUENT [Schoelles *et al.*, 1997], les interactions sont bimodales. L'apprenant écrit dans un champ de texte alors que le système use de l'écrit comme de l'oral pour communiquer avec l'apprenant. Les phrases à prononcer étant connues d'avance pour chaque scène, le problème de la génération est ainsi contourné.

Dans le cadre de notre environnement, seule la modalité écrite a été considérée, autant pour l'apprenant que pour le partenaire. Pour la communication des apprenants, ce choix s'explique par le public que nous avions choisi de cibler : un public de grands débutants sans nationalité précise. Connaissant la diversité des accents, il nous semblait impossible de

pouvoir les traiter dans un même environnement, l'écrit permettant lui une relative unicité des énoncés (même si le corpus réalisé en début d'étude montre des régularités d'erreurs suivant l'origine des apprenants dans leurs écrits). Le partenaire communique aussi de façon écrite, mais tout comme dans FLUENT, l'oral est envisageable voire souhaitable d'après les commentaires qu'a suscité l'environnement lors de ses diverses présentations et lors des entretiens avec les apprenants l'ayant utilisé.

Dans Sampras, l'apprenant et son partenaire peuvent agir de deux façons dans le micromonde : soit avec des actes verbaux, soit avec des actes non-verbaux. L'apprenant peut produire des actes verbaux via un champ de saisie. Les énoncés du compagnon apparaissent dans une bulle style bande dessinée au-dessus de la tête du personnage, ce personnage étant un chef cuisinier dans le micromonde créé. Les actes non-verbaux sont les manipulations des éléments graphiques du micromonde. Ils peuvent être la désignation d'un objet ou d'un endroit, le déplacement d'un objet, la mise en relation de deux objets (dans le cas de la cuisine : ouverture du frigo, prise de la boîte à oeufs et déplacement de celle-ci). Les actes non-verbaux de l'apprenant sont effectués à l'aide de la souris. Le partenaire les effectue de manière transparente, les objets se mettant en action par eux-mêmes dans le micromonde. L'interface du micromonde est présentée Figure 9.

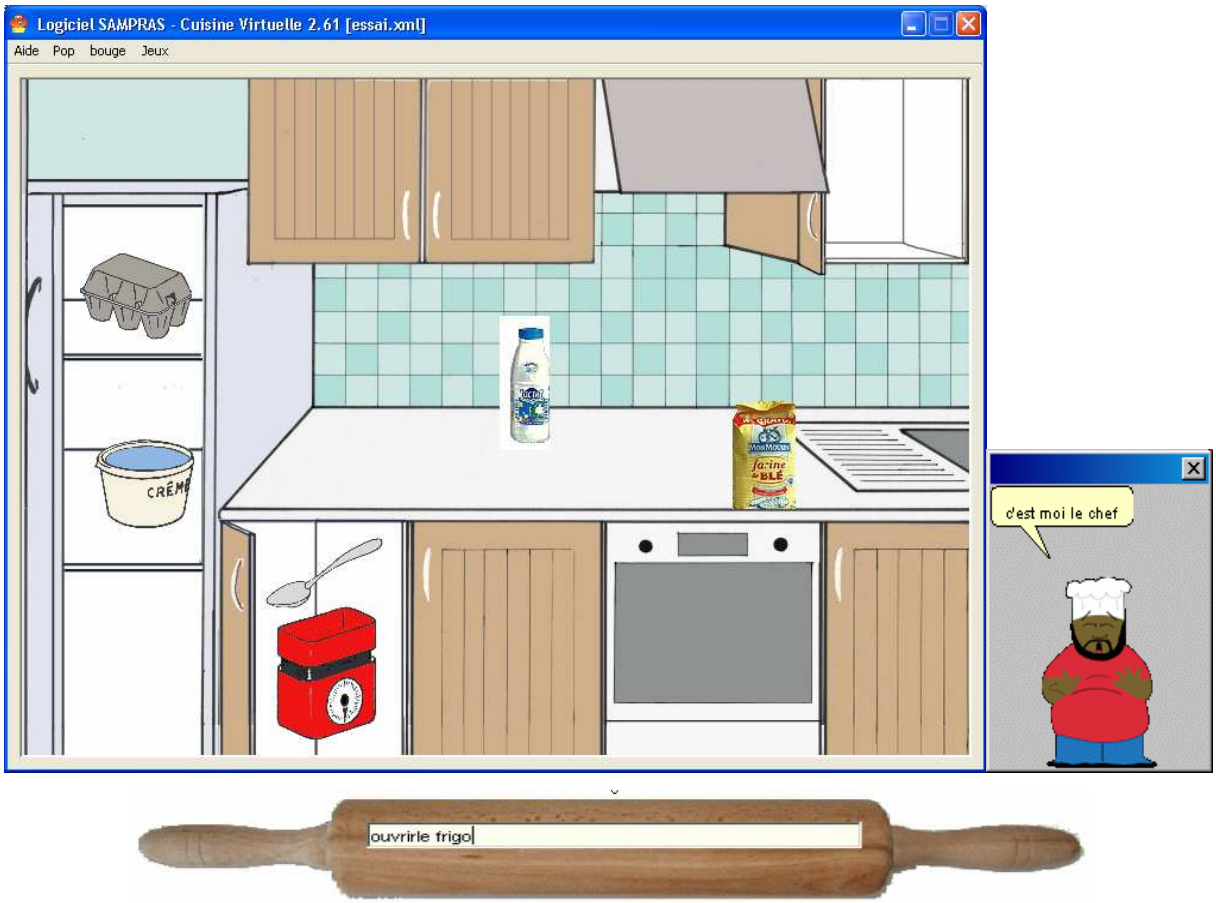


Figure 9 : Interface du micromonde

La mise en place des activités pédagogiques s'appuie sur ces possibilités interactives. Nous donnons pour chaque activité les types d'interaction qui y sont possibles et les assemblons en un modèle d'activité général décrit selon le formalisme UML (diagrammes d'activité).

2.1.3.2 Activité de compréhension

Dans l'activité de compréhension, le partenaire essaie de faire accomplir la tâche applicative par l'apprenant. Le partenaire donne des consignes à l'apprenant, celui-ci essaie de les suivre en manipulant les objets du monde virtuel. Dans le cadre de notre tâche applicative qu'est la réalisation d'une recette, le partenaire donne des actions de la recette à accomplir, l'apprenant les effectue (Figure 10).

Partenaire: « Prendre le moule dans le placard du bas à gauche »

Apprenant: *[déplace le moule du placard vers le plan de travail]*

Figure 10 : Exemples d'échange au cours de l'activité de compréhension

Le seconde type d'interaction concerne la rétroaction faite à l'apprenant lorsqu'il n'effectue pas l'action demandée. Ce *feedback* de type implicite se limite à des marques de non-compréhension du partenaire, remarques reprenant l'action de l'apprenant (Figure 7).

Partenaire: « Prendre le moule dans le placard du bas à gauche »

Apprenant: *[déplace la farine du placard vers le plan de travail]*

Partenaire: « Ceci est la farine. Prendre le moule dans le placard du bas à gauche »

Figure 11 : Exemple de feedback au cours de l'activité de compréhension

Le dernier type d'interaction de cette activité concerne la possibilité donné à l'apprenant de poser des questions au partenaire (ou du moins d'écrire des énoncés à son intention). Si la question est reconnue, le partenaire y répond (Figure 12).

Partenaire: « Prendre le moule dans le placard du bas à gauche »

Apprenant: « Le moule ? »

Partenaire: « C'est un plat. »

Figure 12 : Exemple de question réponse dans l'activité de compréhension

Transposé à l'activité de compréhension, nous obtenons le fonctionnement suivant : le partenaire donne des instructions que l'apprenant complète jusqu'à ce que la tâche soit terminée. Lorsque l'apprenant commet une erreur, le partenaire fait une remarque à ce propos. L'apprenant dispose aussi de la possibilité de poser des questions au partenaire. Nous obtenons alors le modèle de l'activité de compréhension (Figure 13) : le type d'interaction principal est représenté par les flèches bleues, le problème de la non correspondance entre instruction et action qui engage la création d'une rétroaction par les flèches noires, le type d'interaction correspondant aux questions réponses est représenté en rose.

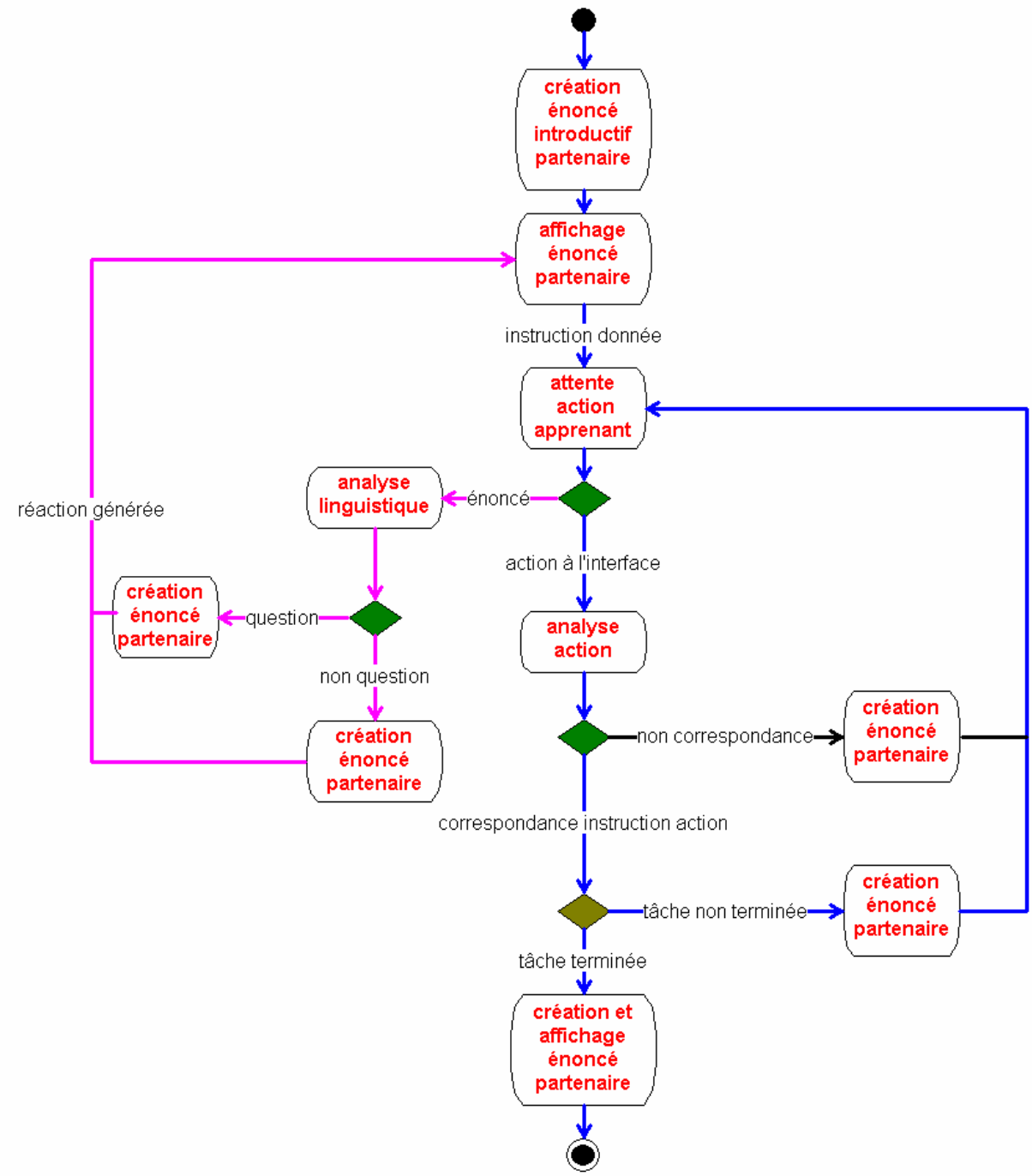


Figure 13: Modèle de l'activité de compréhension

2.1.3.3 Activité de production

Dans l'activité de production, l'apprenant tente de faire exécuter la tâche applicative par le partenaire. Le type d'interaction principal de l'activité pédagogique est l'apprenant qui donne des instructions correctement formées et réalisables au partenaire. Dans notre environnement actuel, l'apprenant, à partir d'une description sommaire de la tâche applicative

et des objets du monde, donne des instructions au partenaire qui les exécute lorsqu'elles sont reconnues. Nous présentons un exemple de ce type d'interaction (Figure 14).

Apprenant : « Prendre le moule dans le placard du bas à gauche »

Partenaire : « J'ai bien compris ton ordre et je l'exécute. Je prends le moule. »
[le moule est pris par le partenaire et mis sur le plan de travail]

Figure 14 : Exemple d'échange au cours de l'activité de production

Le second type d'interaction concerne les cas où l'instruction n'est pas réalisable au vu de l'avancement de la tâche et de l'état du monde (Figure 15).

Apprenant : « Prendre le moule dans le placard du bas à gauche »

Partenaire : « J'ai bien compris ton ordre mais je ne peux pas l'exécuter»

Figure 15 : Exemple d'échange où l'instruction est irréalisable

La rétroaction sur les énoncés incorrects de l'apprenant reprend des éléments de son énoncé et commente ou formule une question à son endroit (Figure 16).

Apprenant : « Pendre les oeufs »

Partenaire : « Je ne comprends pas lorsque tu écris "pendre les oeufs". Que veux-tu faire avec les oeufs ? »

Figure 16 : Exemple de *rétroaction* au cours de l'activité de production

S'ajoute aussi la possibilité pour l'apprenant de poser des questions au partenaire s'il désire des renseignements sur la tâche ou les objets du monde (position, état…), reprise de la remarque individuelle que pourrait se faire l'apprenant à lui-même (Figure 17).

Apprenant : « Le moule ? »

Partenaire : « C'est un plat. »

Figure 17 : Exemple de question/réponse au cours de l'activité de production

Le modèle pour l'activité de production est présenté Figure 18.

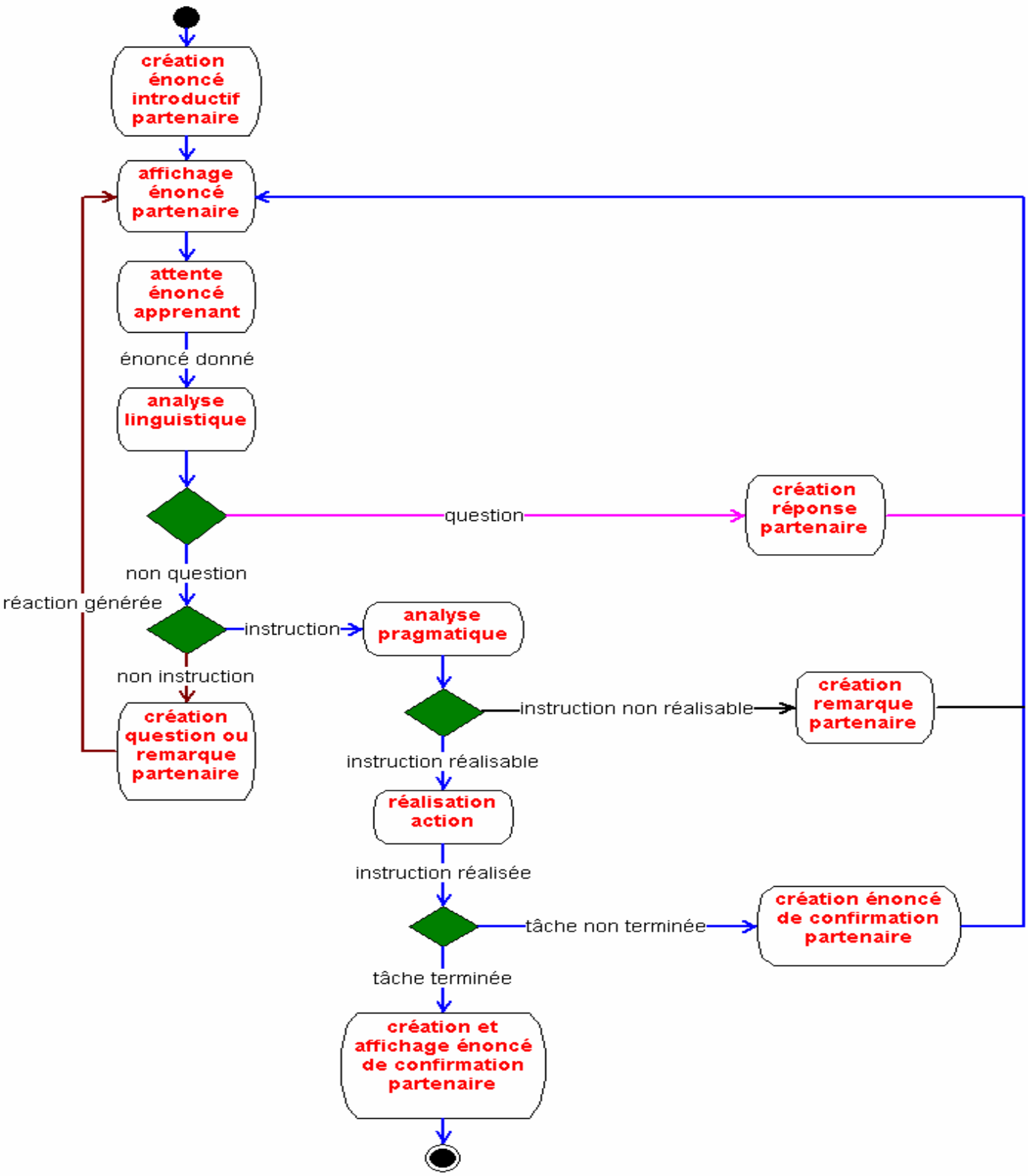


Figure 18 : Modèle de l'activité de production

2.2 Interaction

2.2.1 DU CONCEPT D'INTERACTION DANS LES EIAH 60
2.2.1.1 Dialogue homme-machine et tuteurs intelligents 61
2.2.1.2 Dialogue homme-machine et apprentissage des langues 62
2.2.2 TYPES D'INTERACTION 64
2.2.2.1 Nature des échanges 64
2.2.2.2 Description des interactions 64
2.2.2.3 Types d'interaction au cours de l'activité de compréhension 65
2.2.2.4 Types d'interaction au cours de l'activité de production 68
2.2.3 MODELE D'INTERACTION 70
2.2.3.1 Représentation des échanges 70
2.2.3.2 Déroulement des échanges 79

La problématique de l'interaction comme possible déclencheur d'apprentissages est le second pôle sur lequel notre étude se penche. Cette section se propose tout d'abord de présenter le **concept d'interaction** dans les EIAH et plus spécifiquement dans les EIAH en langue et ceux qui utilisent le dialogue homme-machine. Ensuite sont évoqués les **types d'interaction** d'accomplissement des activités, nous y voyons la nature des échanges ainsi que leur représentation influencées par les choix didactiques et la nature de la tâche sous les angles applicatif et didactique. La dernière partie est consacrée au **modèle dynamique de l'interaction** qui permet d'en avoir une représentation globale.

2.2.1 Du concept d'interaction dans les EIAH

Le concept d'interaction a toujours été un thème central de la recherche en EIAH. Historiquement, ce champ de recherche fut même appelé EIAO (Environnements Interactifs d'Apprentissage par Ordinateur) [Bruillard, 1997] dénomination sans doute due à un effet de mode autour du terme « interactivité » dans les années 1990. Ce thème se décline généralement selon deux sous-thèmes [Baker *et al.*, 2006] : les interactions homme-homme (vu sous les angles de la médiation ou de la médiatisation) et les interactions homme-ordinateur (vu sous les angles du dialogue homme-machine ou des relations sujet-milieu). Dans le cadre de notre étude, nous nous situons clairement dans le champ des interactions homme-ordinateur et plus précisément du dialogue homme-machine (même si l'usage de l'environnement implique aussi des conséquences dans le champ de la relation sujet-milieu). Les premières utilisations du dialogue homme-machine dans les EIAH ont été vues au travers des tuteurs intelligents, puis au travers des micromondes.

2.2.1.1 Dialogue homme-machine et tuteurs intelligents

La mise en oeuvre de modèles issus du dialogue homme-machine (DHM) dans les environnements informatiques d'apprentissage humain remonte aux premiers tuteurs intelligents où le dialogue de type socratique (dialogue dont le but des participants est de faire émerger un consensus autour d'une vérité, dans notre cas, une ou des connaissances) était la norme. Le dialogue est le moyen d'expression des connaissances de la machine. Les premiers systèmes significatifs utilisant cette approche furent SCHOLAR, et « WHY ». SCHOLAR est le premier système de ce type et est consacré aux connaissances géographiques mais ne dépasse pas l'horizon du vrai/faux en termes de *rétroaction* [Carbonell, 1970]. « WHY » dépasse ce cadre et engage effectivement des dialogues « socratiques » pour essayer de faire comprendre ses erreurs à l'apprenant [Collins *et al.*, 1982]. L'architecture classique des tuteurs intelligents repose alors sur quatre modules : une base de connaissances relatives au domaine que l'environnement se propose d'explorer, un modèle de l'apprenant qui contient les informations recueillies ou présupposés sur l'utilisateur, l'interface utilisateur ou l'environnement d'apprentissage, et un module pédagogique qui contient les tactiques pédagogiques à mettre en œuvre pour permettre l'acquisition des connaissances, la composante dialogue y étant incluse [Wenger, 1987].

Cette vision est ensuite dépassée pour arriver à la séparation du module de dialogue et du module pédagogique, le dialogue (l'interaction) et son organisation devenant un problème à part entière dans la conception des EIAH. Ainsi dans [Joab, 1990], le système de dialogue est indépendant du domaine de connaissances et s'inspire de modèles provenant des sciences du langage et en particulier la « pragmatique » ([Roulet, 1985], [Moeschler, 1985]). Le module de dialogue n'est pas dédié aux connaissances à apprendre mais permet d'identifier les intentions de l'utilisateur par l'identification des actes de langages de celui-ci, permettant alors l'adaptation du dialogue aux attitudes de l'apprenant. L'environnement STUDIA, destiné à l'apprentissage de méthodes de calcul statistique, s'appuie sur un modèle de dialogue basé sur la négociation [Chevallier, 1992]. L'idée est de pouvoir négocier l'organisation des interactions apprenant-système autour de la résolution de problèmes de statistiques, permettant une certaine souplesse dans le déroulement des activités. Ces travaux d'adjonction de systèmes dialoguants à des tuteurs intelligents se poursuivent et nous pouvons citer un des plus représentatifs : Autotutor [Graesser *et al.*, 2001]. Autotutor se propose d'engager des dialogues en langage naturel avec des apprenants. Le tuteur est un agent animé

dont le comportement est une simulation de comportements que pourraient adopter des tuteurs humains. L'agent animé pose une question sur le sujet d'apprentissage et l'étudiant est amené à construire une réponse argumentée dans laquelle il doit articuler ses connaissances. Les connaissances à acquérir et les connaissances sur le dialogue sont séparées, celles-ci étant organisées dans un module à part. Autotutor n'est pas un « délivreur de connaissances » mais plutôt un composant dialoguant qui essaye de faire formuler à l'apprenant ses connaissances.

Ces environnements se concentrent sur des apprentissages qui ne sont pas en langue, c'est-à-dire où le dialogue porte sur un objet à apprendre qui n'est pas le moyen du dialogue. Les environnements d'apprentissage des langues possèdent leurs propres spécificités, liées à la nature des apprentissages qu'ils entendent susciter ou faciliter.

2.2.1.2 Dialogue homme-machine et apprentissage des langues

La particularité du dialogue en apprentissage des langues est qu'il n'a nul besoin d'être tutorial. Dans ce cas, il ressemble à une conversation entre deux interlocuteurs autour d'un thème particulier ou d'une activité à réaliser suivant la mise en scène de la conversation. Le sujet à apprendre étant le langage lui-même, faire avancer la conversation a en soi des mérites éducatifs [Slabbers *et al.*, 2005], le présupposé pédagogique étant qu'au sein de l'interaction prend place l'apprentissage de la langue. Ce modèle, dit de l'apprentissage dans l'interaction ou acquisition, postule qu'au cours de la communication entre un locuteur apprenant et un locuteur natif se déroulent des séquences interactives, appelées **séquences potentiellement acquisitionnelles**, où l'acquisition en langue peut prendre place [De Pietro *et al.*, 1989]. Les caractéristiques des activités permettant ces acquisitions sont décrites (§ 1.3.2.2). Ces caractéristiques permettent l'établissement d'un modèle d'apprentissage.

La difficulté habituelle de la conception d'EIAH reposant sur le dialogue homme-machine est le passage d'un modèle d'apprentissage à un modèle d'interaction [Baker *et al.*, 2006]. Dans notre cas, l'apprentissage reposant sur l'interaction, notre tâche est simplifiée : identifier les caractéristiques portant sur les interactions et imaginer leur mise en place dans un modèle d'interaction « informatisable ». La première caractéristique intéressante à mettre en place au niveau de l'interaction est le ***noticing***, moment où l'apprenant remarque un problème linguistique. Cette constatation peut provenir d'un *feedback* externe ou interne. La seconde caractéristique est la possibilité pour l'apprenant de **corriger et modifier ses**

interventions. La troisième, **la négociation de sens**, porte sur la possibilité de modifier la structure de l'interaction en cours pour se focaliser sur un problème qui surviendrait lors de cette interaction [Chapelle, 1998]. Le modèle d'interaction essaie donc d'intégrer ces caractéristiques.

Nous notons que ces trois caractéristiques ne sont pas de même niveau : ainsi pour le *noticing*, il s'agit de souligner le problème détecté à l'aide de questions ou affirmations, et de donner à l'apprenant la possibilité de questionner le partenaire, notamment sur le sens de l'énoncé. Le *noticing* est naturellement intégré dans la stratégie de négociation de sens car l'interaction est modifiée et on se concentre sur les problèmes linguistiques, et non sur la tâche applicative. Le *noticing* se réfère à la possibilité dans ses négociations de remarquer ou faire remarquer les points à résoudre. Selon Swain et Lapkin [Swain *et al.*, 1995]: « *En produisant de la langue, un apprenant sera occasionnellement conscient (i.e remarquera) d'un problème linguistique (apporté à son attention par un feedback externe (par exemple des demandes de clarifications) ou feedback interne). Remarquer un problème pousse l'apprenant à modifier sa production langagière.* » (traduction libre). La négociation de sens se réfère au processus de compréhension de l'apport langagier mais sans tout comprendre, et de produire des actes de langue sans atteindre la perfection. La négociation de sens caractérise des moments de communication imparfaits et s'étend à leurs résolutions [Long, 1996]. Ce processus arrive lorsque le fil de la conversation est modifié suite à des problèmes de communication. A ce propos, il faut aussi noter que, comme le dit Vasseur [Vasseur *et al.*, 1992], ces activités réflexives se déroulent aussi lorsqu'un apprenant tente de construire ses phrases et il n'y a pas forcément rupture de la communication pour autant. La négociation de sens est un phénomène se déroulant sur un enchaînement de plusieurs échanges interactifs alors que le phénomène de *noticing* se retrouve sur un seul échange entre l'apprenant et le partenaire. De même, le phénomène d'autocorrection peut se dérouler sur plusieurs échanges. La description de ces phénomènes ne s'envisage donc pas au même niveau : si la question de la rétroaction se pose sur un seul échange, celles de négociation de sens ou de correction peuvent se dérouler sur plusieurs échanges [Michel *et al.*, 2005]. Elles entrent dans des phénomènes de dynamique de l'interaction et montrent la nécessité d'une mémoire de l'interaction.

2.2.2 Types d'interaction

Cette section présente les différents types d'interaction qui peuvent prendre place au cours des activités. Nous nous intéressons tout d'abord à la nature des échanges qui peuvent avoir lieu entre l'apprenant et son partenaire, puis à la description des types d'interaction ou d'échange qui peuvent prendre place au cours des activités. La troisième partie présente les types d'interaction pouvant avoir lieu lors de l'activité de compréhension, la dernière ceux de l'activité de production. Ces types d'interaction permettent la réalisation des activités (Figure 19).

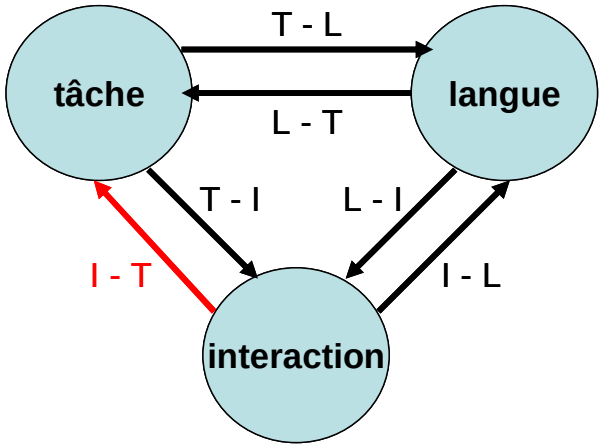


Figure 19 : Lien interaction et activité

2.2.2.1 Nature des échanges

De par les différentes modalités offertes à l'apprenant et au partenaire dans nos activités, il existe deux types d'actions: les actions verbales et les actions non-verbales. Pour simplifier la suite de notre propos, nous nous proposons de nommer « énoncé » une action verbale. Le terme « action » qualifie quant à lui une action non-verbale, c'est à dire une action de manipulation d'un objet du monde. Ainsi les échanges entre l'apprenant et le partenaire ont lieu sous forme d'échanges d'énoncés et d'actions via l'interface.

2.2.2.2 Description des interactions

La modélisation des types d'interaction repose sur la notion de fonction illocutoire décrite dans des travaux de linguistique pragmatique [Moeschler, 1985], [Roulet, 1985]. Les échanges décrits dans ce modèle s'organisent autours des fonctions initiatives, réactives et évaluatives. Une intervention initiative d'un locuteur ouvre un échange, une intervention

réactive marque la réponse de l'interlocuteur. L'intervention évaluative qui peut suivre marque la satisfaction du premier locuteur vis-à-vis de l'intervention réactive. Nous utilisons cette caractérisation des échanges possibles au cours d'une activité pour décrire les échanges attendus lors des activités de production et de compréhension. Nous ne prenons pas en compte pour l'instant les aspects hiérarchiques du modèle fonctionnel, qui peut se déployer récursivement pour décrire les changements d'initiatives.

Pour schématiser les types d'interaction, nous nous inspirons des diagrammes de séquence UML. Ces représentations n'incluent pas l'interface car cela alourdit inutilement la présentation. Sur la Figure 20, nous représentons deux schémas équivalents. Le schéma de gauche présente une vision complète de l'interaction : le système écrit un énoncé dans la bulle du partenaire à l'interface (texte), l'apprenant perçoit ce texte (perception), l'apprenant répond en manipulant un objet de l'interface (action), le système interprète cette action (interprétation). Le schéma de droite se résume à l'énoncé du système et l'action de l'apprenant. La chronologie des actes est descendante. Nous utilisons ce formalisme pour décrire les interactions.

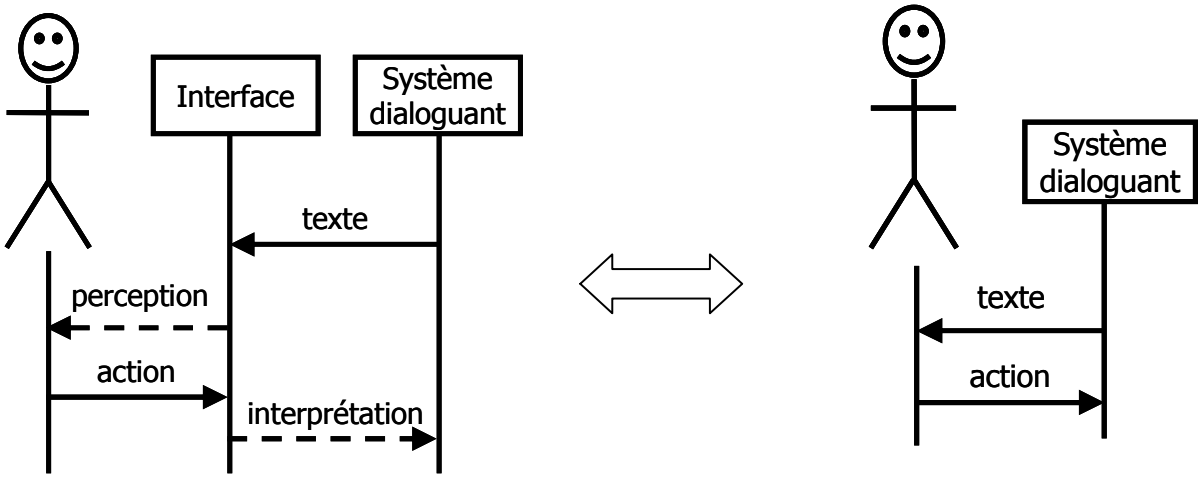


Figure 20 : Lecture simplifiée des types d'interaction

2.2.2.3 Types d'interaction au cours de l'activité de compréhension

L'activité de compréhension met en jeu la compétence de compréhension de l'apprenant, la consigne étant que l'apprenant exécute les instructions données par son partenaire pour accomplir la tâche applicative. Cette activité repose donc principalement sur

des interventions de type énoncé pour le partenaire et de type action pour l'apprenant. Trois types d'interaction existent pour cette activité.

Le premier est celui où le partenaire donne une instruction et où l'apprenant effectue l'action correspondante sur l'interface. Le partenaire est ici à l'initiative de la séquence interactive. Il donne son instruction au travers de sa bulle (intervention initiative) et l'apprenant la réalise sur l'interface en manipulant les objets de l'interface (intervention réactive). La séquence interactive s'achève alors la confirmation par le partenaire de l'action de l'apprenant (intervention évaluative) (Figure 21).

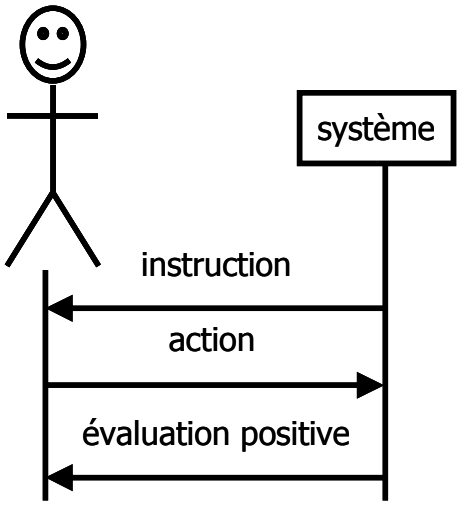


Figure 21 : Type d'interaction attendu au cours de l'activité de compréhension

Le second type d'interaction est celui où le partenaire donne une instruction (intervention initiative) et où le partenaire n'effectue pas l'action demandée (intervention réactive) mais réagit à l'action de l'apprenant en marquant son désaccord avec l'action entreprise (intervention évaluative) (Figure 22). Cette réaction constitue une rétroaction donnée à l'apprenant.

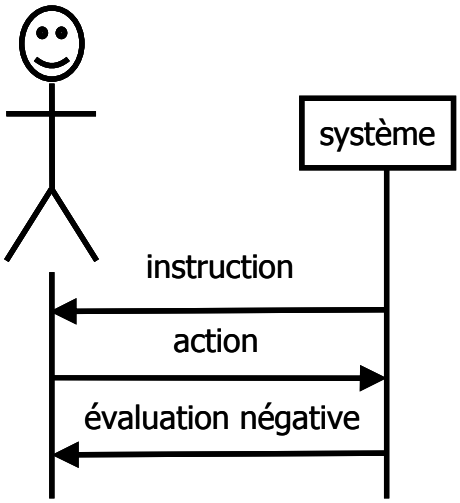


Figure 22 : Type d'interaction quand une action de l'apprenant pose problème

Le troisième et dernier type d'interaction est celui où l'apprenant pose une question au partenaire (acte de type énoncé) dans son champ de texte, le partenaire répond à la question s'il le peut. Une confirmation de l'apprenant peut suivre (autre acte de type énoncé) (Figure 23).

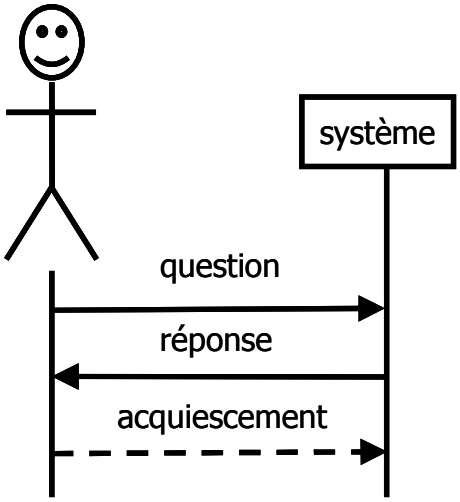


Figure 23 : Type d'interaction de question-réponse

Etant donné que nous nous sommes focalisés sur l'activité de production, nous n'avons pas exploré davantage l'activité de compréhension. Le choix de se consacrer à l'activité de production provient de l'orientation donnée à notre travail : création d'un corpus d'énoncés, travaux de DEA de Fabien Delorme fixant un modèle rudimentaire de réaction aux énoncés de l'apprenant, analyseur prenant en compte les spécificités des énoncés du corpus.

Le centre d'intérêt de nos recherches se portent sur les interactions langagières prioritairement, interactions qui sont plus importantes dans le cas de l'activité de production.

2.2.2.4 Types d'interaction au cours de l'activité de production

L'activité de production met en jeu la compétence de production de l'apprenant, la consigne étant que l'apprenant donne les instructions au partenaire pour accomplir la tâche applicative. Nous avons donc principalement des interventions de type énoncé pour l'apprenant et des interventions de type action et énoncé pour le partenaire. Le type principal d'interaction est celui où l'apprenant donne des instructions complètes ou partielles au partenaire (intervention initiative). Le partenaire répond en effectuant l'action correspondante à l'interface et en commentant l'énoncé de l'apprenant (intervention réactive) (Figure 24).

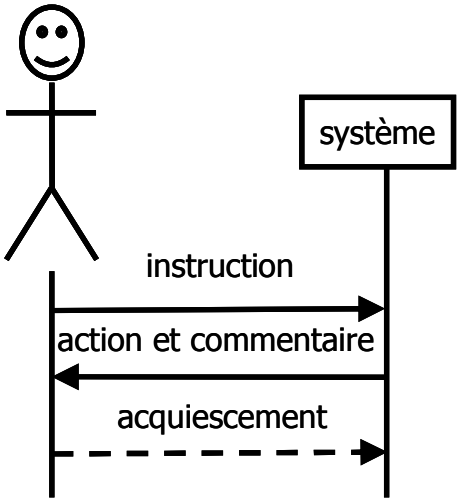


Figure 24 : Type d'interaction attendu au cours de l'activité de production

Le second type d'interaction possible concerne le cas où l'apprenant donne un énoncé qui se veut une instruction complète ou partielle (intervention initiative). L'analyse de cet énoncé révèle que cette instruction est problématique (non-pertinence de l'instruction par rapport à l'état de la tâche et/ou du monde). Le partenaire réagit par un énoncé où il peut commenter l'instruction (intervention réactive) (Figure 25). Ceci donne la possibilité dans le prochain échange à l'apprenant de corriger son énoncé.

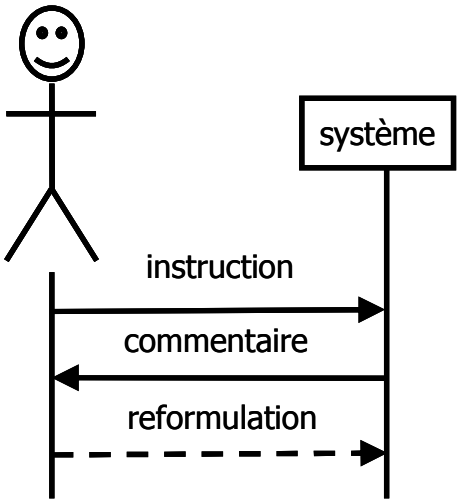


Figure 25 : Type d'interaction lorsqu'une instruction de l'apprenant pose problème

Le troisième type d'interaction est celui correspondant aux énoncés problématiques de l'apprenant (intervention initiative). L'analyse de l'énoncé montre l'existence de problèmes linguistiques dans l'énoncé. Suivant la reconnaissance ou non d'éléments dans l'énoncé qui pourraient aider à sa compréhension, le partenaire commente ou pose une question à l'apprenant (intervention réactive) (Figure 26). Ceci donne aussi la possibilité dans le prochain échange à l'apprenant de corriger son énoncé.

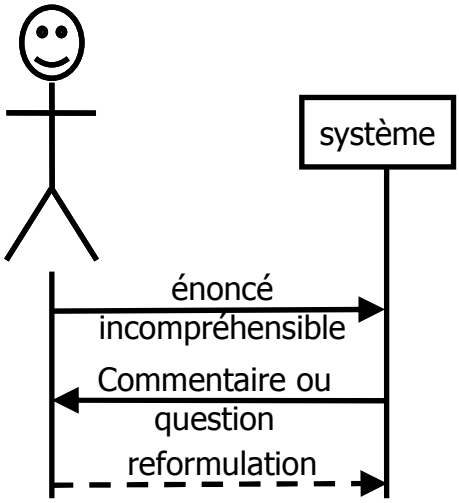


Figure 26 : Type d'interaction lorsqu'un énoncé contient des problèmes linguistiques

Le dernier type d'interaction est le même que celui offert dans l'activité de compréhension, c'est-à-dire l'apprenant qui pose une question (acte de type énoncé) en

s'aidant de son champ de texte au partenaire, lequel répond à la question s'il le peut (autre acte de type énoncé). Une confirmation de l'apprenant peut suivre (Figure 27).

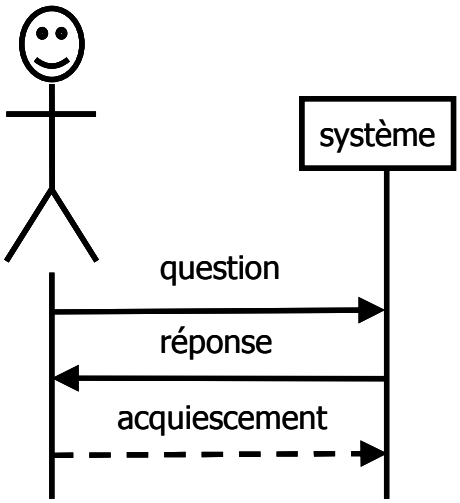


Figure 27 : Type d'interaction de question-réponse

Ces types d'interaction, très figés, marquent les échanges possibles entre l'apprenant et le partenaire. Le pas suivant dans la modélisation de l'interaction est de décrire comment s'organisent ces différents échanges entre eux au cours de l'activité.

2.2.3 Modèle d'interaction

Cette partie s'attache à montrer l'organisation de l'interaction. La démarche est de partir d'une représentation des échanges et, en déroulant ces échanges, de faire émerger un modèle dynamique de l'interaction, une représentation pour la machine de l'interaction en cours. Nous nous intéressons ici uniquement au modèle d'interaction de l'activité dite de production puisque celle-ci seulement a donné lieu à un développement informatique.

2.2.3.1 Représentation des échanges

En repartant du modèle d'activité (Figure 18) et des types d'interaction présentés dans le paragraphe précédent, nous pouvons différencier les différents cas d'interaction en fonction de ce que l'apprenant a produit comme énoncé. Des types d'interaction de l'activité de production, nous distinguons trois cas d'énoncés qui donnent lieu à des interprétations : un énoncé reconnu comme une instruction, un énoncé reconnu comme une question et un énoncé dont la compréhension pose problème (problèmes linguistiques).

En réponse à une instruction ou au complément d'une instruction, suivant l'analyse qui en est faite, trois cas sont identifiés. Si l'énoncé est une instruction qu'il est possible de réaliser (c'est-à-dire faisant partie de la tâche applicative et dont les pré-conditions de la ou méthodes qui la réalisent sont remplies), dans ce cas, elle est accomplie par le partenaire qui commente cette réalisation. Si l'énoncé est une instruction qui n'est pas comprise dans la tâche applicative, le partenaire la fait remarquer à l'apprenant. Si l'énoncé représente une instruction comprise dans la tâche applicative mais qui n'est pas réalisable au vu de l'avancement de la tâche applicative ou de l'état du monde, le partenaire le fait remarquer à l'apprenant.

En réponse à une question, le partenaire peut réagir de deux façons. Si la question est comprise, il donne la réponse exacte sinon il fait remarquer qu'il ne peut y répondre.

Si l'énoncé est incompréhensible, le partenaire a deux possibilités. S'il y a des éléments identifiés dans l'énoncé et d'autres sur lesquels sont posées des hypothèses, il peut formuler une question à l'adresse de l'apprenant en reprenant les éléments identifiés dans son énoncé. Cette question marque son incompréhension et dirige la suite du dialogue vers la résolution du problème de communication. Par contre si aucun élément n'a été identifié dans l'énoncé de l'apprenant, le partenaire le fait remarquer à ce dernier.

Nous pouvons donc grouper les réactions accomplir « une instruction », « répondre à une question », « poser une question » ou « faire remarquer les problèmes en commentant les énoncés de l'apprenant ». Nous obtenons le schéma général suivant (Figure 28). Les stratégies quant à la génération complète de la réaction selon les trois différents cas sont vues plus loin dans la section concernant le traitement automatique de la langue.

Si l'énoncé est une instruction ou le complément d'une instruction :

Elle peut être accomplie……L'accomplir et le commenter

Elle est hors-tâche……..Le faire remarquer

Elle ne peut pas être accomplie……...……..Le faire remarquer

Si l'énoncé est une question :

Réponse possible……Répondre

Réponse impossible……....Le faire remarquer

Si l'énoncé est incompréhensible :

Avec des hypothèses……...Poser une question

Sans hypothèses……..Le faire remarquer

Figure 28 : Enoncés de l'apprenant et réactions du partenaire

Pour représenter un échange du point de vue machine, nous disposons du concept d'Unité Minimale d'Interaction (UMI) [Lehuen, 1997]. Une UMI s'attache à découper l'interaction en constituants significatifs. Ces constituants se relient ensuite entre eux pour constituer la structure du dialogue. Ceci permet au système d'avoir une vision subjective de l'interaction en cours. Lehuen décrit ainsi une UMI (Tableau 4).

Champ	Intitulé	Description succincte
E_i	Etat courant de la tâche	Pointeur vers une description de la tâche
M_i	Intervention de la machine	Suite de symboles produits par la machine
Att_{mi}	Attentes de la machine	Contexte pour interpréter l'intervention H_i
H_i	Intervention de l'usager	Suite de symboles perçue par la machine
Rep_{hi}	Granules de compréhension	Eléments d'interprétations de l'intervention H_i
Hyp_i	Hypothèses de la machine	Catégorisations, implicites, anaphores

Tableau 4 : Aperçu de la structure d'unité minimale d'interaction, extrait de [Lehuen, 1997]

Dans le cadre de notre activité, une UMI peut être vue de cette manière (Tableau 5). Le premier E_i est élargie à l'état du monde. Le champ M_i représente les énoncés du partenaire, souvent des réactions aux énoncés de l'apprenant. Les attentes du système représentent un énoncé (qu'il soit instruction, question, ou énoncé incorrect). L'intervention de l'usager est son énoncé. Les granules de compréhension sont les éléments d'interprétation fournis par l'analyseur (ce que l'analyseur a « reconnu »), ces éléments étant conservés jusqu'à

émergence du sens de l'énoncé H_i (une « mémoire » de l'interaction en quelque sorte) ou passage à une autre étape de l'activité. Le dernier champ, celui des hypothèses concerne les hypothèses que l'analyseur peut porter sur des éléments non reconnus par l'analyseur. La définition de Lehuen de ces hypothèses va bien plus loin, ces hypothèses pouvant porter sur des résolutions d'anaphore par exemple, mais pour l'instant ces hypothèses ne sont pas prévues ou leur traitement est implicite par la permissivité du système sur certains énoncés.

Champ	Intitulé
E_i	Avancement de la tâche et état des objets du monde
M_i	Intervention du partenaire (de la machine)
Att_{mi}	une instruction ou question
H_i	Enoncé de l'apprenant
Rep_{hi}	éléments reconnus de l'énoncé
Hyp_i	hypothèses sur les mots inconnus de l'énoncé

Tableau 5 : Une UMI dans l'activité de production

Cette idée des UMIs qui permet de caractériser un échange du point de vue machine peut se redéfinir dans notre contexte particulier de cette façon. L'activité de production repose sur le type d'interaction où l'apprenant écrit un texte qui peut être une instruction à l'intention du partenaire et où celui-ci réalise cette instruction. Il est alors possible de penser que les UMIs donnent une structuration décalée de l'échange. Ce n'est pas le cas si l'on considère que les UMIs sont une représentation subjective du dialogue du point de vue de la machine. La machine interprète toujours un énoncé par rapport au contexte applicatif et interactionnel immédiat, c'est le rôle de l'UMI de fournir ce contexte.

Représentons la correspondance entre les types d'interaction vus antérieurement et les UMIs ainsi définies. Nous imaginons une situation d'interaction puis la dérivons selon UMIs et types d'interaction (Figure 29).

Partenaire : « Dis-moi comment réaliser un flan au chocolat. »

Apprenant : « Ouvrir le frigo »

Partenaire : « J'ai bien compris ton ordre et je l'exécute. J'ouvre le frigo. »
[le frigo est ouvert par le partenaire]

Figure 29 : Echange au cours de l'activité de production

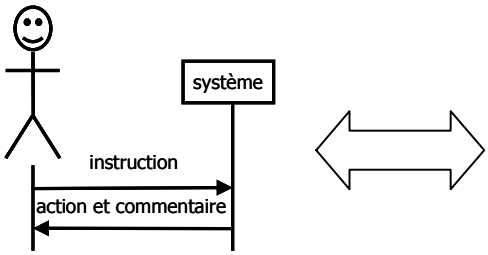
Cet échange peut représenter le premier échange possible de l'activité. En termes d'UMI, nous en obtenons la représentation suivante (Figure 27). Deux UMIs sont nécessitées pour cette représentation. Les états de la tâche et du monde de la première UMI sont leurs états initiaux E_1. La première intervention du partenaire est l'énoncé M_1. L'énoncé de l'apprenant est H_1. Ce qui est reconnu est une instruction d'ouverture du frigo (Rep_{h1}). Aucune hypothèse n'est à poser sur les éléments de l'énoncé de l'apprenant (Hyp_1). Les états de la tâche et du monde de la seconde UMI incorporent le changement réalisé où le frigo est désormais ouvert E_2. Le nouvel énoncé du partenaire est construit et dénote l'ouverture du frigo, frigo qui est effectivement ouvert à l'interface en M_2.

E_1	Tâche et monde à l'état initial
M_1	« Dis-moi comment réaliser un flan au chocolat. »
Att_{m1}	Attentes : instruction ou question
H_1	« Ouvrir le frigo »
Rep_{h1}	Instruction d'ouverture du frigo
Hyp_1	Aucune hypothèse

E_2	Frigo ouvert
M_2	«J'ai bien compris ton ordre et je l'exécute. J'ouvre le frigo.»
Att_{m2}	Attentes : instruction ou question
H_2	
Rep_{h2}	
Hyp_2	

Figure 30 : Représentation en UMI de l'échange Figure 29

La correspondance avec le type d'interaction principal de l'activité de production s'instancie ainsi (Figure 31). L'instruction de l'apprenant est le champ H_1, l'action et le commentaire du partenaire constituant le champ M_2.



E_1	
M_1	
Att_{m1}	
H_1	Instruction
Rep_{h1}	
Hyp_1	

E_2	
M_2	Action et commentaire
Att_{m2}	
H_2	
Rep_{h2}	
Hyp_2	

Figure 31 : Correspondance entre le type d'interaction principal de l'activité de production et la représentation en UMIs

Dans le cas du second type d'interaction, celui où l'apprenant commet une erreur dans son énoncé, nous avons l'exemple suivant que nous représentons sous forme d'UMIs (Figure 32).

Partenaire : « Dis-moi comment réaliser un flan au chocolat. »

Apprenant : « Ouvrir le fridge »

Partenaire : «Je ne te comprends pas lorsque tu écris « ouvrir le fridge ». Que veux-tu ouvrir ? »

Figure 32 : Un échange avec un problème dans l'énoncé

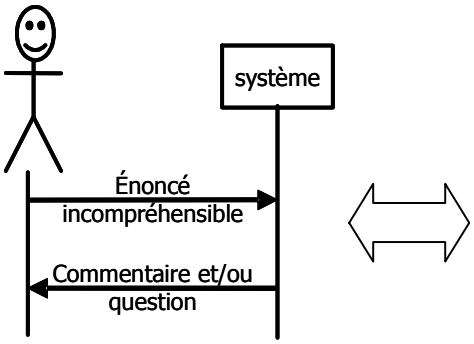
Les états de la tâche et du monde ainsi que la première intervention du partenaire E_1, M_1 sont les mêmes que dans le cas précédent. Le changement concerne les autres champs de la première UMI. Ainsi le champ H_1 contient l'énoncé « ouvrir le fridge ». Un élément représentant l'action « ouvrir » est reconnu. Des hypothèses sont posées sur l'élément « le fridge », ainsi il peut s'agit d'un contenant, d'un rangement ou d'un appareil électroménager. Dans la seconde UMI, le premier champ est le même que celui de la première, l'état du monde et de la tâche n'ayant pas été modifiés. Le champ M_2 représente la réaction du partenaire à l'énoncé de l'apprenant marquant la non-compréhension de l'énoncé et l'interrogation sur l'élément non reconnu (Figure 33).

E_1	Tâche et monde à l'état initial
M_1	« Dis-moi comment réaliser un flan au chocolat. »
Att_{m1}	Attentes : instruction ou question
H_1	« Ouvrir le fridge
Rep_{h1}	Ouvrir
Hyp_1	rangement, contenant, electro-ménager

E_2	Frigo ouvert
M_2	Je ne te comprends pas lorsque tu écris « ouvrir le fridge ». Que veux-tu ouvrir ?
Att_{m2}	Attentes : instruction ou question
H_2	
Rep_{h2}	
Hyp_2	

Figure 33 : Représentation en UMI de l'échange Figure 32

Comme dans le cas précédent, nous avons la correspondance entre type d'interaction et l'UMI suivante (Figure 34).



E_1	
M_1	
Att_{m1}	
H_1	Enoncé incompréhensible
Rep_{h1}	
Hyp_1	

E_2	
M_2	Commentaire et/ou question
Att_{m2}	
H_2	
Rep_{h2}	
Hyp_2	

Figure 34 : Correspondance entre le type d'interaction feedback de l'activité de production et la représentation en UMIs

Le cas du dernier type d'interaction, c'est-à-dire celui des questions-réponses. L'exemple suivant l'illustre (Figure 35).

Apprenant : « Où est le lait ? »

Partenaire : « Le lait est dans le frigo. »

Figure 35 : Exemple d'échange de question-réponse

La représentation en terme d'UMI est présentée Figure 36. Les deux champs sur les états du monde et de la tâche sont identiques. L'énoncé (H_1) est reconnu comme une question sur la place du lait (Rep_{h1}) sans aucune hypothèse sur un élément de l'énoncé (Hyp_1). Le champ représentant la réponse du partenaire reprend l'élément recherché et donne sa place (M_2).

E_1	Tâche et monde en un état
M_1	xxx
Att_{m1}	Attentes : instruction ou question
H_1	« Où est le lait ? »
Rep_{h1}	Question sur la place du lait
Hyp_1	Aucune hypothèse

E_2	Tâche et monde sans modification
M_2	Le lait est dans le frigo
Att_{m2}	Attentes : instruction ou question
H_2	
Rep_{h2}	
Hyp_2	

Figure 36 : Représentation en UMI de l'échange de la Figure 35

La correspondance entre type d'interaction et UMI se visualise ainsi (Figure 37).

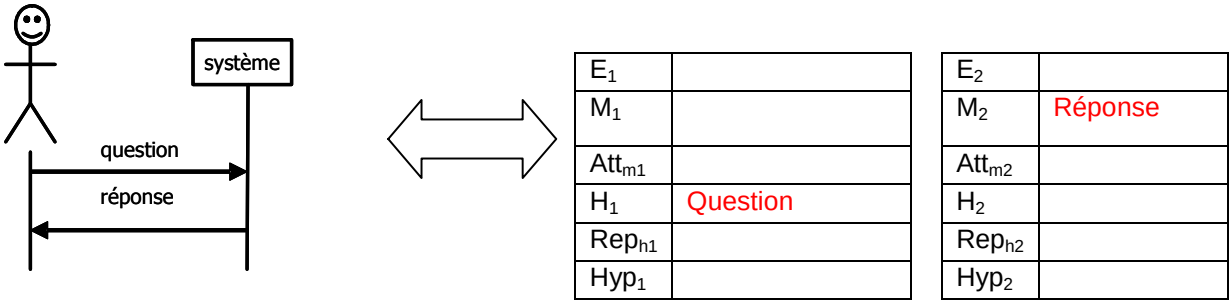


Figure 37 : Correspondance entre le type d'interaction question-réponse et la représentation en UMIs

La correspondance entre UMIs et type d'interaction prend sens ici. Les échanges se déroulent comme des alternances de types d'interaction, les UMIs suivent ces mêmes échanges. En poussant plus loin ce raisonnement, l'activité peut aussi se représenter à l'aide des UMIs en reprenant le diagramme d'activité de l'activité de production et remplaçant les énoncés effectifs par leur représentation sous forme UMI, nous obtenons alors le diagramme (Figure 38).

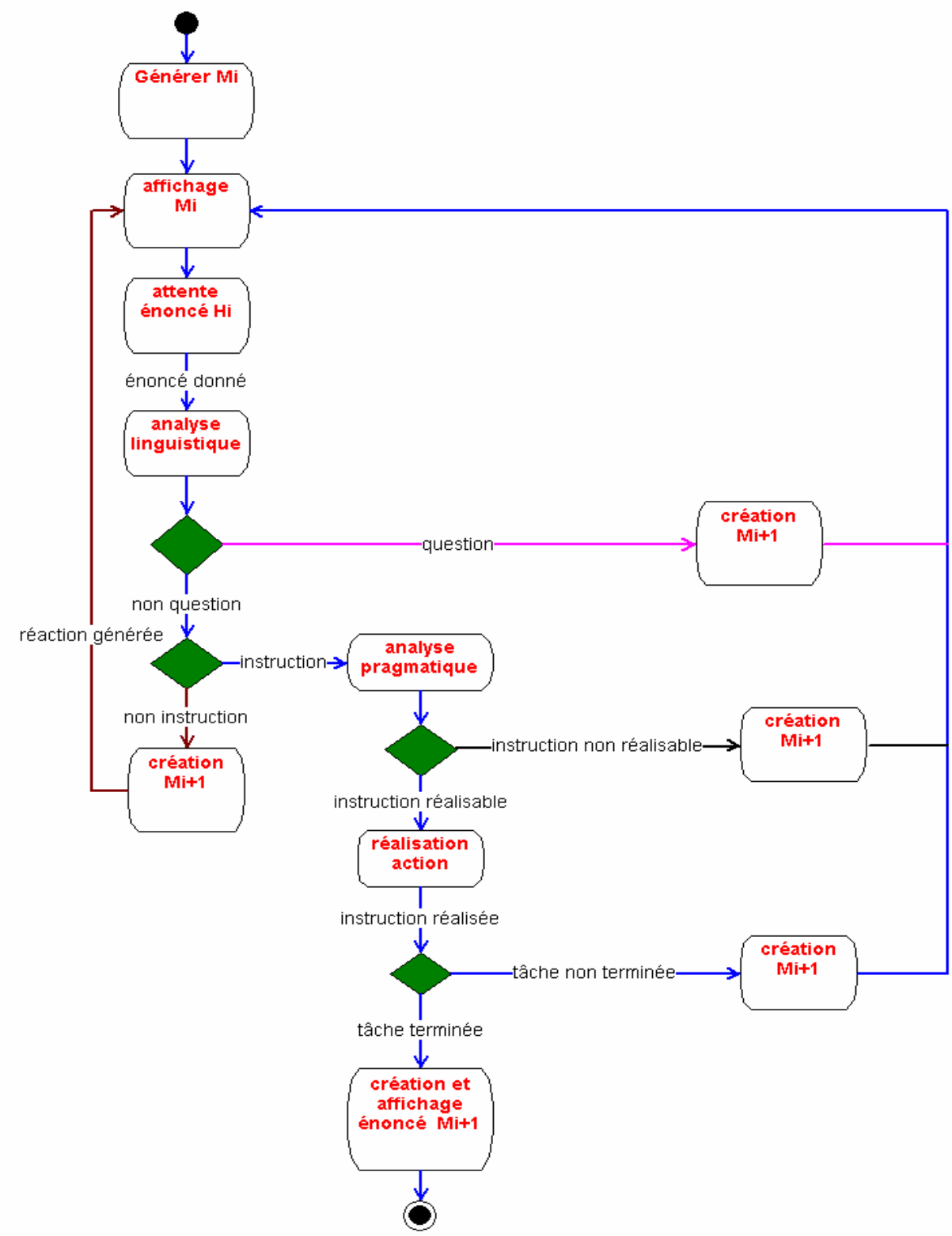


Figure 38 : Diagramme d'activité vu selon les UMIs

2.2.3.2 Déroulement des échanges

Pour rendre compte de la dynamicité de l'interaction, nous la présentons comme l'enchaînement de différentes UMIs. S'inspirant du modèle de Luzzati [Luzzati, 1995a], Lehuen présente le dialogue se déroulant selon deux axes : un axe **applicatif** qui dénote la réalisation des buts successifs des locuteurs (la réalisation de la tâche) et un axe **discursif** qui dénote la résolution des problèmes ayant lieu au cours du dialogue. Nous transformons légèrement ce modèle pour y inclure les incidences côté locuteur (ses questions). Ainsi l'axe de réalisation des buts successifs devient simplement l'avancée de la tâche applicative et l'axe discursif est l'axe de résolution des problèmes, que cette résolution soit initiée par l'apprenant comme par la machine.

La structuration dynamique de l'interaction peut prendre différentes formes [Lehuen, 1997]. La structuration linéaire permet la description selon l'axe temporel de l'interaction, l'interaction y étant vue comme le simple enchaînement des différentes UMIs, comme présenté sur les figures

Figure 31, Figure 34 et Figure 37. La seconde représentation, celle qui nous intéresse principalement, est la représentation imbriquée issue d'une structuration récursive. Elle permet de comprendre l'évolution de l'interaction en fonction des problèmes rencontrés au cours de celle-ci en plaçant les UMIs selon les axes régissant et incident. Ainsi l'interaction est vue comme suivant l'un ou l'autre axe. Lehuen introduit les notions de **schéma régissant** et **schéma incident**. Le schéma régissant correspond à une évaluation positive de l'UMI, l'interaction avance le long de l'axe régissant, la tâche s'effectue, et une UMI dite régissante est instanciée. Lorsqu'un énoncé ne permet pas de faire avancer l'interaction, soit parce qu'il comporte des erreurs linguistiques, soit parce qu'il est une question, soit parce qu'il correspond à une instruction impossible à réaliser ou déjà réalisée, nous nous trouvons dans le cas du schéma incident. Une UMI incidente est instanciée dans le but de solutionner le problème détecté. La construction de l'énoncé du partenaire de cette UMI est réalisée à l'aide de la description de la tâche et de l'instruction donnée par l'apprenant.

La construction de l'énoncé de l'UMI incidente, donc de l'intervention du partenaire, dépend de ce que Lehuen nomme des **tactiques dialogiques**. Dans le cas des énoncés problématiques où l'incidence est à l'initiative de la machine, ces tactiques cherchent à

implémenter la « bonne intervention » en fonction de l'énoncé problématique. Ces tactiques dialogiques sont ici les incarnations du *feedback* externe ou rétroaction. Ces interventions peuvent être les suivantes :

Si l'énoncé est une instruction ou le complément d'une instruction :

Elle est hors-tâche……Le faire remarquer

Elle ne peut pas être accomplie……...……....Le faire remarquer

Si l'énoncé est une question

Réponse impossible……....Le faire remarquer

Si l'énoncé est incompréhensible :

Avec des hypothèses…….. .Poser une question

Sans hypothèses……...Le faire remarquer

L'intérêt des schémas incidents est qu'ils se rapprochent de l'idée de négociation de sens, laquelle porte sur la possibilité de voir l'interaction modifiée si un problème survient dans la communication entre apprenant et partenaire. Ce rapprochement peut se voir lors de négociation sur le sens d'un énoncé, au travers des possibilités laissées à l'apprenant de se corriger. Les exemples présentés ci après ne sont pas extraits du corpus recueilli mais écrits par nous-même afin d'illustrer au mieux les problèmes traités.

Si nous prenons l'exemple suivant (Figure 39), nous observons un premier échange qui se déroule sans problème. Puis un énoncé n'est pas complètement compréhensible, « prendre les oufs », donnant lieu à une question d'éclaircissement de la part du partenaire. Le problème se résout à l'énoncé suivant de l'apprenant. Dans le cas où une suite d'énoncés ne résout pas le problème, l'apprenant peut repartir sur une nouvelle intervention de son choix.

Partenaire : « Dis moi comment faire un gâteau au chocolat »

Apprenant : « Ouvrir le frigo »

Partenaire : « J'ouvre le frigo » [ouverture du frigo]

Apprenant : « Prendre les oufs »

Partenaire : « Je ne te comprends pas lorsque tu écris « prendre les oufs ». Que veux-tu prendre ? »

Apprenant : « Les œufs »

Partenaire : « Je sors les œufs du frigo »

Figure 39 : Exemple d'énoncé donnant lieu à une négociation de sens

La représentation récursive en UMI donne le schéma suivant (Figure 40). Les flèches représentent le sens de l'interaction selon qu'elle se déplace le long de la tâche à effectuer ou selon l'axe des problèmes rencontrés. La flèche 1, orientée selon l'axe régissant, relie les deux premières UMIs, l'interaction se passe sans problème : le partenaire ouvre le frigo en réponse à l'énoncé « ouvrir le frigo » de l'apprenant. La flèche 2, orienté selon l'axe incident, dénote un problème de communication entre le partenaire et l'apprenant : l'énoncé « prendre les oufs » n'est que partiellement compréhensible et, s'appuyant sur les éléments reconnus dans l'énoncé de l'apprenant, le partenaire demande ce que l'apprenant veut lui faire prendre. La flèche 3 revient de l'UMI incidente vers la seconde UMI, le problème interactif a été résolu : l'énoncé « les œufs » vient compléter l'instruction donnée par l'apprenant qui était de prendre quelque chose. La seconde UMI est ici un point de repère qui montre la résolution. L'UMI suivante est bien celle qui suit la flèche 4 où l'action de « prendre les œufs » est effectuée par le partenaire.

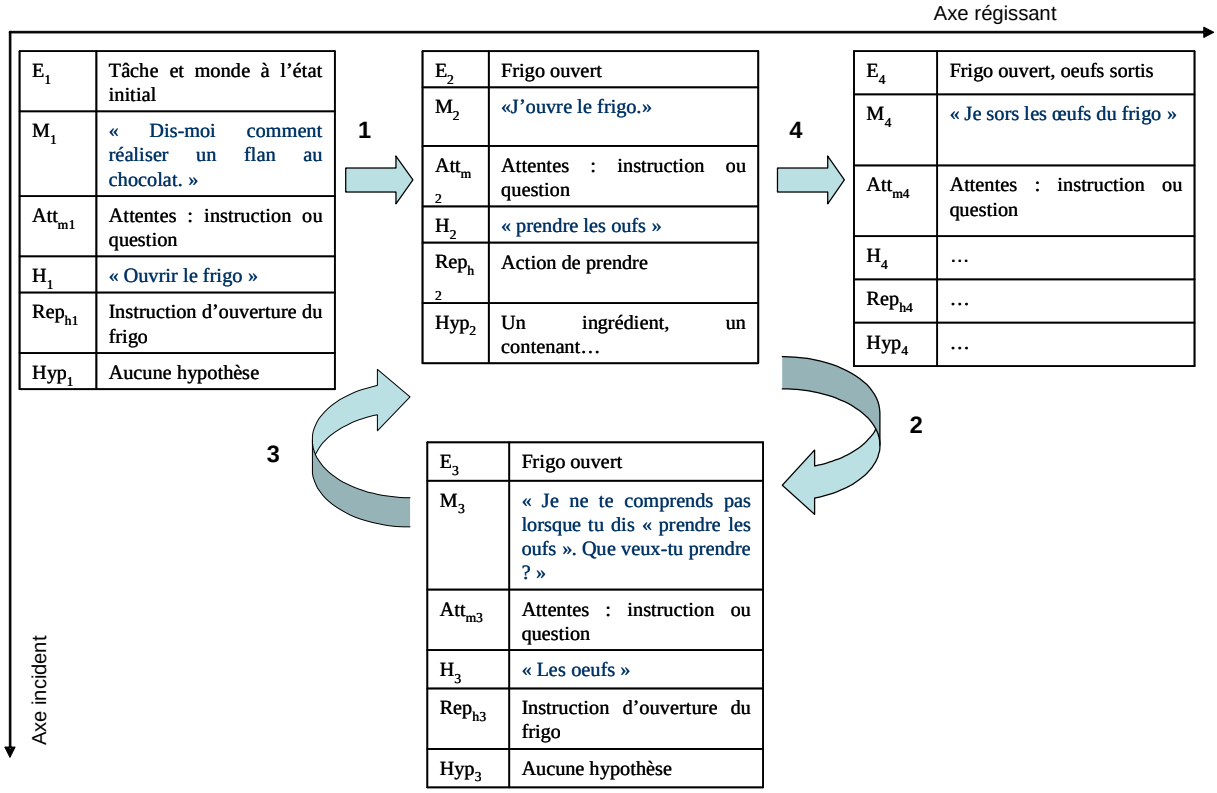


Figure 40 : Représentation en UMI de l'échange Figure 39

Les séquences incidentes sont donc provoquées par une instruction irréalisable ou hors-tâche, un problème linguistique dans l'énoncé ou à l'initiative de l'apprenant. Ces

schémas incidents peuvent se reproduire sur eux-mêmes soit par succession, soit par imbrication. [Lehuen, 1997]. Nous reprenons ces schémas en les adaptant à notre problème. Le schéma par succession est celui qui se produit lorsqu'une nouvelle incidence est créée à partir d'une UMI déjà incidente. Dans le cas de la suite d'échanges présentés Figure 43, deux échanges ayant pour but de résoudre des problèmes différents s'enchaînent. Le premier échange problématique porte sur le fait que l'apprenant désire ouvrir un placard. Cette instruction n'est pas complète car il y a six placards dans la cuisine. L'échange suivant, le partenaire lui demande quel placard il désire ouvrir. L'apprenant répond « du haut » ce qui est toujours insuffisant pour localiser le bon placard. L'apprenant produit alors l'énoncé « à droite » qui complète l'instruction commencé deux échanges plus haut. Le partenaire peut alors effectuer l'action ce qui clôt l'échange et faire repartir l'interaction pour accomplir la tâche.

Partenaire : « Dis moi comment faire un gâteau au chocolat »
Apprenant : « Ouvrir le frigo »
Partenaire : « J'ouvre le frigo » [ouverture du frigo]
Apprenant : « Ouvrir le placard »
Partenaire : « J'ai bien compris ton ordre mais je ne peux pas l'exécuter. Quel placard veux-tu ouvrir»
Apprenant : « du haut ?»
Partenaire : « J'ai bien compris ton ordre mais je ne peux pas l'exécuter. Quel placard veux-tu ouvrir ?»
Apprenant : «à droite»
Partenaire : « J'ouvre le placard » [ouverture du placard]

Figure 41 : Echange où deux incidences se succèdent

En termes de représentation en UMI dans le cas du schéma successif, nous avons l'enchaînement d'UMI suivant (Figure 42). La flèche 1 marque une interaction qui avance le long de l'axe régissant, le frigo étant alors ouvert. La flèche 2 marque une incidence liée au fait que le placard ne peut être situé. La flèche 3 repart vers la seconde UMI car le partenaire recherche toujours le placard que l'apprenant désire ouvrir et nous avons une nouvelle UMI incidente. L'énoncé de l'apprenant permet de compléter son énoncé précédent, donnant alors la position du placard (flèche 4). De même, la flèche 5 remonte vers la seconde UMI pour

marquer la résolution du problème. La flèche 6 montre l'interaction qui repart selon l'axe régissant. L'exemple présenté reprend l'état actuel du système : la réaction du partenaire est deux fois la même. Le problème des tactiques de réponse en fonction du degré d'incidence n'a pas été considéré mais constitue une piste à explorer pour obtenir un dialogue plus fluide et intéressant. De même, l'interaction reste très liée à la tâche applicative et aucune intervention de type métalinguistique n'a été pensée.

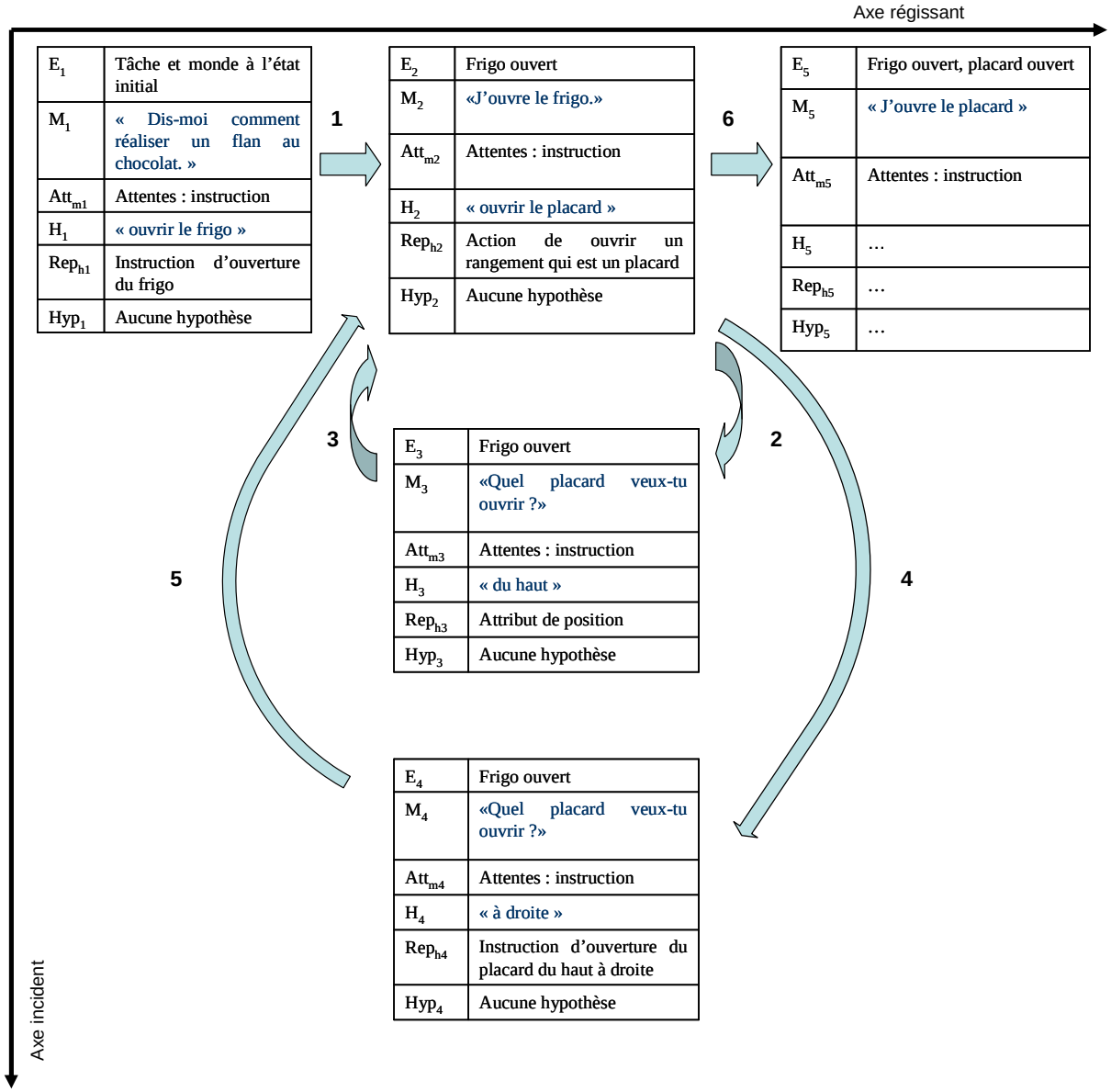


Figure 42 : Représentation en UMI de l'échange Figure 41

Le schéma successif est celui qui à partir d'un point d'avancement de la tâche, pose un problème ne le résout pas et l'UMI suivante pose un autre problème, non lié au problème soulevé par l'UMI précédemment créée. Nous en avons le cas suivant (Figure 43). Le premier

problème survient lorsque l'apprenant demande de « prendre les oufs ». Le partenaire réagit en lui demandant ce qu'il veut prendre. Ici, au lieu de répondre au partenaire, l'apprenant choisit de tenter une autre instruction sans rapport avec la question posée par le partenaire. Mais cet énoncé est aussi problématique que le premier et l'interaction n'avance toujours pas. Le partenaire souligne le nouveau problème par une question, lui redemandant ce qu'il veut prendre. L'apprenant répond « le lait » et l'interaction peut alors avancer.

Partenaire : « Dis moi comment faire un gâteau au chocolat »
Apprenant : « Ouvrir le frigo »
Partenaire : « J'ouvre le frigo » [ouverture du frigo]
Apprenant : « Prendre les oufs »
Partenaire : « Je ne te comprends pas lorsque tu écris « prendre les oufs ». Que veux-tu prendre ? »
Apprenant : « Prendre le latte »
Partenaire : « Je ne te comprends pas lorsque tu écris « prendre le latte». Que veux-tu prendre ? »
Apprenant : « le lait »
Partenaire : « Je sors le lait du frigo » [sortie du lait du frigo]

Figure 43 : Echange où deux incidences s'imbriquent

La traduction en UMI de cet échange et sa représentation de façon imbriquée sont présentées Figure 44. La flèche 1, orientée selon l'axe régissant, marque le passage entre les deux premières UMIs, aucun cas d'incidence ne se produit. La flèche 2 marque une incidence provoquée par l'énoncé de l'apprenant. La flèche 3 descend vers l'UMI suivante car l'énoncé de l'apprenant ne résout pas le problème posé. La flèche 4, revenant vers la deuxième UMI, montre la résolution du problème. La flèche 5 montre la continuation de l'interaction. Contrairement au modèle de Lehuen, nous ne remontons pas au niveau de l'axe régissant. Nous n'essayons pas de résoudre la totalité des problèmes et préférons continuer à partir du point où nous sommes dès qu'une instruction est reconnue.

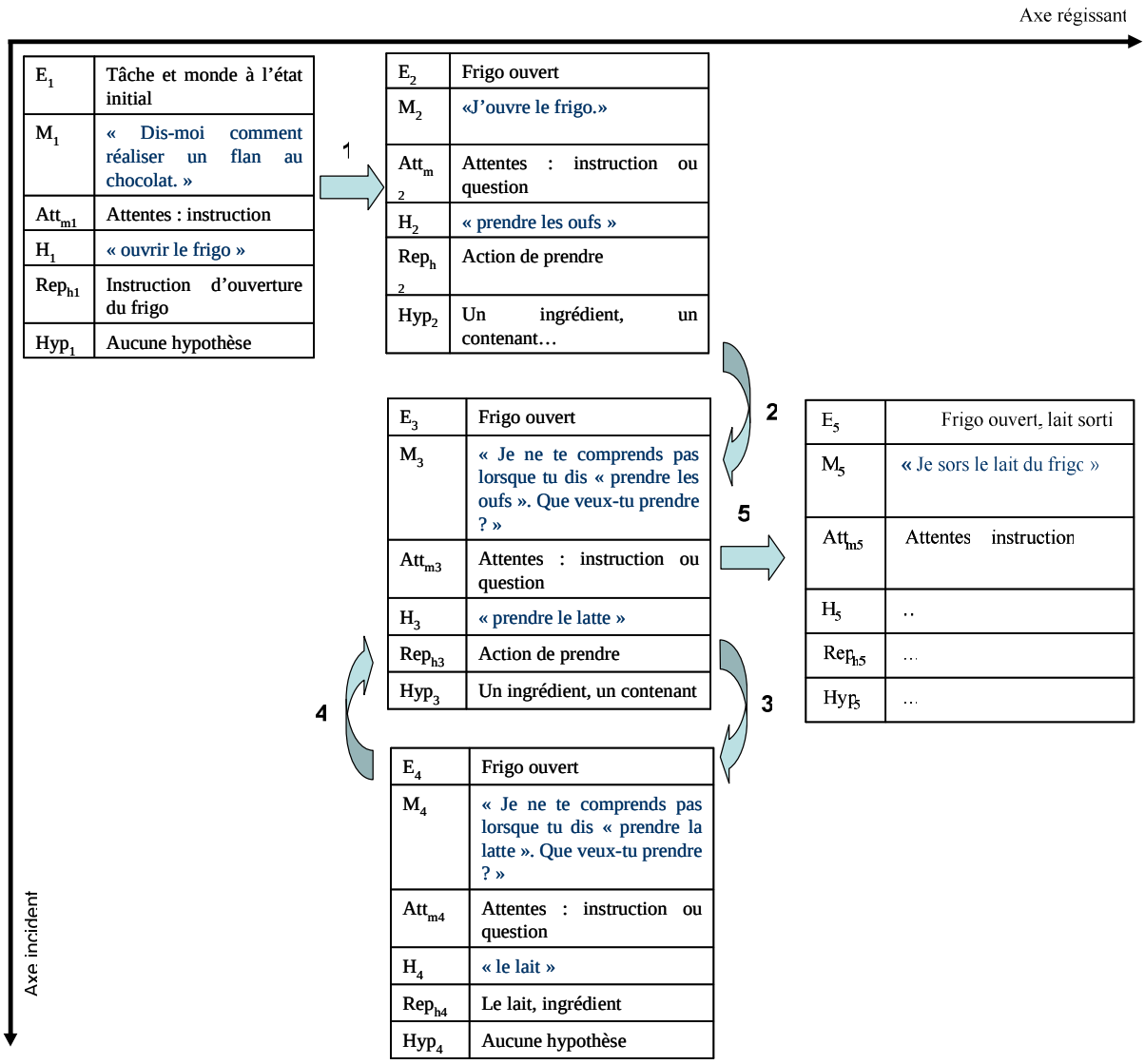


Figure 44 : Représentation en UMI de l'échange Figure 43

Nous remarquons qu'une UMI ou un groupe d'UMIs incidentes peuvent être remplacée par un seul conteneur (appelé unité minimale de résolution [Lehuen, 1997]). Ceci correspond à la résolution d'un problème interactionnel ou langagier, une structure d'interaction modifiée pour supporter la résolution d'un problème : une négociation de sens. Nous avons légèrement modifié le modèle de Lehuen pour ce genre de problèmes. Le modèle de Lehuen fait toujours remonter les incidences jusqu'à l'axe régissant, résolvant tous les problèmes rencontrés (sauf si impasse). Pour ne pas perdre le fil de l'interaction à tenter de refermer des problèmes, nous préférons reprendre l'interaction le long de l'axe régissant dès que possible. L'apprenant est responsable du changement de mouvement du dialogue en se focalisant sur une autre instruction.

2.3 Langue

2.3.1 TRAITEMENT AUTOMATIQUE DES LANGUES ET APPRENTISSAGE DES LANGUES ASSISTE PAR ORDINATEUR 86
2.3.2 ANALYSE ET GENERATION EN CONTEXTE 89
2.3.2.1 Rétroaction 89
2.3.2.2 Analyse 90
2.3.2.3 Génération de la réponse 92
2.3.3 LEXIQUE 93
2.3.4 UNE ANALYSE, SUPPORT A L'INTERACTION : COMMENT ? 95
2.3.4.1 Observation de l'énoncé 96
2.3.4.2 Comparaison des énoncés à la description de la tâche 106
2.3.4.3 Génération 109

La démarche pédagogique repose sur l'hypothèse que l'acquisition de la langue prend place au cours de la communication entre l'apprenant et son partenaire. La langue étant l'objet d'apprentissage, nombre d'énoncés sont attendus comme problématiques et une analyse robuste est nécessaire. Cette analyse est à la fois tolérante aux écarts et erreurs mais permet de les détecter. La section qui suit est dédiée au problème du traitement de la langue dans notre environnement et son contexte particulier. Elle commence par l'exploration des liens entre apprentissage des langues et traitement automatique des langues et se poursuit par la description des mécanismes d'analyse des énoncés de l'apprenant et de génération des interventions du partenaire.

2.3.1 Traitement automatique des langues et apprentissage des langues assisté par ordinateur

L'apparition du TAL dans les environnements d'ALAO date du milieu des années 1980 pour l'analyse des réponses des apprenants en vue de la création du *feedback* sur ces mêmes réponses. L'engouement pour le TAL chez les enseignants en langue, qui suivait le mouvement « tuteurs intelligents », retomba devant la complexité de celui-ci et les praticiens de la didactique des langues étrangères sont plus que sceptiques sur son usage [Rezeau, 2001], préférant des approches plus modestes telles que la simple reconnaissance de mots-clés par exemple. Le courant continua par contre chez les chercheurs en IA et EIAH. L'approche en vogue en apprentissage des langues durant les années 1990 était l'approche communicative et les environnements qui ont découlé de cette première utilisation du TAL étaient des tuteurs intelligents et des micromondes où l'activité principale se fait sous le signe de la communication entre un apprenant et un tuteur ou un compagnon s'il s'agit d'un micromonde.

Depuis les années 2000, par effet de balancier, la mode en apprentissage des langues retourne vers une nouvelle insistance sur la forme [Skehan, 2002], et le TAL perd de sa centralité, comme support d'une communication, dans les environnements au profit d'une approche « outil » [Karlström *et al.*, 2005] où celui-ci est appelé à de plus petites tâches et où sa faillibilité est moins grande. Aujourd'hui, TAL et ALAO concernent de nombreux travaux, même si ces travaux n'ont pas tous la même vision de l'emploi du TAL.

Plusieurs façons d'utiliser les traitements automatiques de la langue peuvent être envisagées pour l'apprentissage des langues en fonction de l'approche prise quant aux tâches à réaliser par les apprenants. Dans le cas d'une approche explicite, c'est-à-dire insistant sur l'apprentissage des formes de la langue, le TAL a un rôle d'expertise sur les productions de l'apprenant, délivrant les erreurs de celui-ci d'un point de vue grammatical et lexical. Cette vision se réalise au travers de l'approche « outil » du TAL. Sur des petites activités dédiées à des points particuliers de grammaire, syntaxe ou lexique, des micro-tâches [Demaiziere *et al.*, 2005], l'outil TAL fournit un cadre d'analyse de productions au contenu relativement prédéfini. Les outils permettent une expertise sur les réponses des apprenants aux exercices. L'élément traitement de la langue peut être utilisé pour montrer directement les erreurs grammaticales aux élèves comme dans le système FreeText [L'Haire *et al.*, 2003] ou German Tutor [Heift, 2003]. Les problèmes de ces systèmes se trouvent au niveau des retours à donner à l'apprenant sur ses erreurs, en particulier dans le cas des erreurs multiples dans une réponse, la question étant de savoir quelles erreurs mettre en avant. Les possibles traitements de la langue peuvent aussi être liés à des activités types que des enseignants peuvent créer comme sur la plate-forme MIRTO [Antoniadis *et al.*, 2004]. Dans cet environnement, les enseignants peuvent décrire des activités, en se basant sur des scripts décrivant des exercices (par exemple, texte à trou) et associant à ces exercices les fonctions TAL nécessitées pour construire l'activité (étiquetage morphosyntaxique et lemmatisation dans le cas du texte à trou par exemple). Les activités créées peuvent s'agencer selon des scénarii créés aussi par l'enseignant.

Dans le cas d'une approche implicite de l'apprentissage de la langue, basée sur la compréhension et la production d'énoncés dans un contexte de communication, le TAL est le support de l'interaction, son but étant de fournir les éléments qui permettent de continuer la communication et l'avancée de l'interaction. Cette vision s'accorde avec des environnements proposant des activités de communication entre un apprenant et un partenaire virtuel. Ces

activités sont des macro-tâches au sens de Demaizière [Demaiziere *et al.*, 2005], *« des formes de mise en scène de la réalité »*, où au sein de l'interaction, prend place l'acquisition en langue. Dans le cas des micromondes d'apprentissage des langues, c'est le TAL comme outil support de l'interaction qui est considéré : les résultats de l'analyse sont au service de la communication qui se déroule entre l'apprenant et son partenaire. Par exemple dans FLUENT [Schoelles *et al.*, 1997], le composant TAL a pour but de générer des descriptions, des instructions ou des questions à l'intention de l'apprenant. Il repose sur des descriptions du lexique, de la grammaire et d'un système de contrôle de la flexibilité de l'analyse suivant le degré de correction des énoncés souhaités pour l'apprenant. Le composant TAL produit une analyse structurelle de l'énoncé de l'apprenant, et crée à partir de celle-ci une nouvelle intervention du système. L'activité nécessitant cette analyse s'appelle QuizzMaster : le système accomplit une action dans le micromonde, l'activation d'un composant de la cuisine par exemple, et pose une question sur cette action. L'apprenant dispose d'un champ de texte où il répond à la question. Si celle-ci est incorrecte, un choix de réponses possibles est offert à l'apprenant. S'il se trompe une nouvelle fois, la réponse lui est donnée. Rappellons que dans FLUENT le système est considéré comme un tuteur et non un compagnon comme peut l'être le chef dans Sampras. Dans MILT [Holland *et al.*, 1999], le premier prototype supporte une tâche où le soldat apprenant accomplit une mission dans un micromonde. Les missions à accomplir sont la résolution de diverses énigmes telles que « Où attaquera l'ennemi ? », la solution se trouvant dans l'exploration du micromonde. L'apprenant commande un personnage en lui donnant des instructions à travers un champ de texte, le vocabulaire compréhensible par l'agent étant donné dans une fenêtre d'aide. Par contre, les essais d'usage d'un analyseur classique (morphosyntaxique) se révélèrent infructueux. L'analyseur n'était pas assez robuste pour appréhender la variété des énoncés des apprenants et les chercheurs durent opter pour la technique de la reconnaissance de mots-clés. Ceci les fit explorer la possibilité de donner des commandes prédéfinies à faire lire par l'apprenant, solution qu'ils conservèrent pour leur seconde version.

Il est à noter que des tentatives d'hybridation entre les deux approches existent. Ainsi nous avons le cas de BRIDGE [Holland *et al.*, 1993], un environnement d'apprentissage de l'allemand. L'apprenant y est questionné par le système autour d'activités de repérage sur une carte d'une région d'Allemagne. Les réponses de l'apprenant sont analysées et le *feedback* concerne la correction grammaticale de ces énoncés. L'analyseur utilisé est un classique analyseur morphosyntaxique. Les erreurs sont classées selon deux catégories : les erreurs

primaires et les erreurs secondaires, catégories définies par les concepteurs du logiciel. Les erreurs primaires sont signalées à l'apprenant par *feedback* immédiat, les erreurs secondaires sont signalées par alerte que l'étudiant peut ou non consulter.

2.3.2 Analyse et génération en contexte

Notre approche de l'activité de production prend un parti plus extrême que les environnements précités. Elle laisse la totale liberté à l'apprenant quant au contenu de ses énoncés, l'interaction ne s'arrête pas en cas d'erreurs répétées. Le rôle du composant TAL sera donc primordial pour assurer le bon fonctionnement de l'interaction. Sa tâche pourrait même sembler impossible vu la liberté dont dispose l'apprenant. Néanmoins, ce monde clos de la cuisine et la description de la tâche à accomplir délimitent un lexique précis sur lequel s'appuie l'analyse (la mise en scène de l'activité auprès des apprenants consigne ce lexique). Dans un contexte d'EIAH langues pour débutants, nous nous trouvons dans un contexte où les productions recèlent de nombreuses sortes d'erreurs (lexicales et syntaxiques). Le corpus recueilli en début de thèse montre l'extrême non normativité des productions [Lehuen *et al.*, 2002], un analyseur morphosyntaxique classique se heurterait vite à des graves problèmes liés à la diversité des erreurs dans les énoncés, comme ce fut le cas pour MILT [Holland *et al.*, 1999]. L'analyse prend donc un autre parti qui est d'arriver à apprécier le sens « éventuel » des énoncés en essayant de faire émerger ce sens le plus vite possible. En terme de génération, il s'agit de partir des énoncés de l'apprenant en recherchant le *feedback* adapté à la situation en cours comme pensé dans la partie interaction.

2.3.2.1 Rétroaction

La question de l'analyse et de la génération dépend des rétroactions que nous aimerions avoir dans cet environnement, rétroactions de réponse aux énoncés incorrects de l'apprenants. Les premières prescriptions en termes de conception des premiers systèmes d'ALAO vue dans une perspective communicative [Underwood, 1984] cité par [Rezeau, 2001], sous l'angle des rétroactions, sont les suivantes (traduit librement par nous mêmes):

- « *Un environnement d'ALAO communicatif devrait éviter de dire aux apprenants qu'ils ont faux. […] Si les erreurs doivent être prises en compte, le programme devrait aider l'apprenant, soit en donnant une bonne réponse soit en lui donnant des indications dans cette direction.* »

- « *Un environnement d'ALAO communicatif n'utilisera que la langue cible.* »
- « *Un environnement d'ALAO communicatif sera flexible. Les ordinateurs n'ont plus à être plus "rigide et inhumain"que les gens qui les programment.* »
- « *Par dessus tout un environnement d'ALAO communicatif sera amusant Nous avons besoin de laisser les apprenants explorer, expérimenter, apprendre.*»

A l'époque (1984), les activités dans un environnement d'apprentissage des langues selon l'approche communicative sont vues comme des aventures d'exploration des possibilités de la langue dans une situation donnée. On peut noter dans le ton général de ces propositions, l'envie de rupture avec les méthodes béhavioristes qui ont précédé et un optimisme concernant les possibilités de l'informatique (points 3, 4). L'idée de rétroaction apparaît dans le point 1 où les réactions aux énoncés incorrects sont soit la réponse correcte ou des indices vers cette solution. Cette disposition mène à une rétroaction communicative, où à partir d'éléments reconnus dans les énoncés de l'apprenant, des indices portent vers un énoncé correct. De même, l'usage de la langue cible seulement (point 2) dans l'environnement et donc dans les rétroactions données à l'apprenant est reprise dans notre environnement.

2.3.2.2 Analyse

L'orientation dominante de l'analyse est de faire émerger le sens de l'énoncé, sans insister sur une correction syntaxique parfaite, comme le préconise l'approche communicative. En effet, si une correction totale de l'énoncé n'est pas exigée, la préoccupation principale de l'analyse est d'arriver à une interprétation au niveau sémantique de l'énoncé. Les techniques que nous employons pour parvenir à cette interprétation concernent deux points : la création d'**analyses partielles** de l'énoncé et la création d'**hypothèses** sur des éléments non reconnus [Michel *et al.*, 2004]. Nous rejoignons en cela la structure en UMI : l'analyse complète les champs éléments reconnus (Rep_{hi}) et hypothèses (Hyp_i).

La première caractéristique de l'analyse est **la création d'analyses partielles** qui, recoupées, permettent ou non l'appréhension complète de l'énoncé. Dans le cas où l'appréhension complète n'est pas réalisée, les éléments reconnus permettront de réagir à l'énoncé de l'apprenant. L'idée est de découper l'énoncé en blocs décrivant des entités

sémantiques, constituer des mini-blocs de compréhension des énoncés et essayer de les agglutiner pour obtenir une compréhension globale de l'énoncé. Ceci se fait tout d'abord par un travail au niveau du lexique, où les mots et expressions utilisés dans la tâche et dans le monde sont repérés selon des catégories lexicales ou des motifs syntaxiques. Les motifs syntaxiques décrivent des structures syntaxiques « correctes ». Ces descriptions dans le lexique permettent ensuite l'identification des mots et expressions contenus dans l'énoncé. Tout d'abord en repérant les mots ou expressions qui tombent dans les catégories lexicales définies, puis en observant l'articulation de ces mots ou expressions en fonction des motifs syntaxiques décrits dans le lexique. Ce repérage des mots et formes permet de déterminer les corrections lexicales et syntaxiques de l'énoncé. De même les éléments identifiés peuvent être intervertis dans le cas où syntaxiquement l'énoncé est incorrect et que l'on recherche une structure syntaxique pouvant correspondre à un motif du lexique.

La seconde caractéristique de l'analyse est la **faculté à générer des hypothèses** au cours de l'analyse. Lors de la rencontre des mots ou expressions inconnus, des hypothèses sont posées en se fondant sur les motifs syntaxiques définis dans le lexique. Ces hypothèses permettent de donner une idée de ce que pourraient être les éléments non reconnus. Combinées aux analyses partielles, ces hypothèses permettent de choisir entre deux analyses partielles qui se recouvriraient en fonction du calcul d'un coefficient de confiance des analyses partielles. Ce coefficient de confiance est obtenu en comparant le rapport entre le marqueur d'hypothèse et la taille de couverture de l'analyse pour les deux analyses et indique quelle analyse est plus fiable dans ce cas. La figure suivante (Figure 45) présente les deux caractéristiques principales de l'analyse : les flèches bleues montrent la réalisation des analyses partielles et le processus d'agglutination, la flèche rose figure la génération d'hypothèses sur un élément non reconnu.

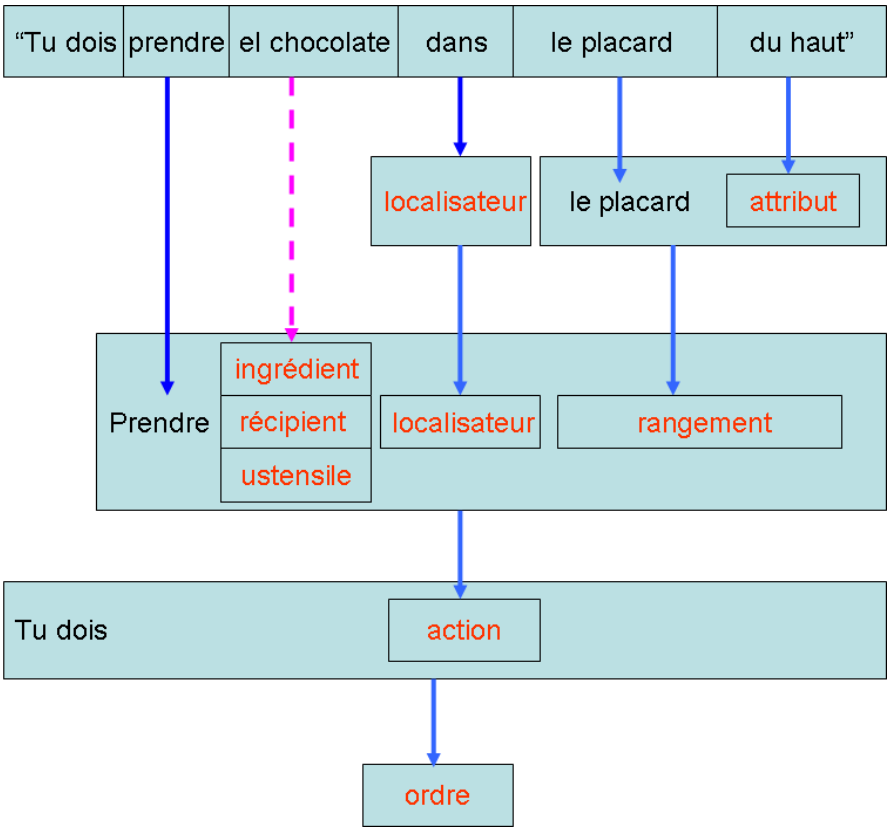


Figure 45 : Exemple d'analyse

Les résultats obtenus sont ensuite confrontés à la tâche et à l'état du monde en fonction des énoncés détectés (instructions, questions, énoncés problématiques). La phase de génération peut commencer.

2.3.2.3 Génération de la réponse

La phase de génération est sensée trouver la réaction adéquate (énoncé et/ou action) du partenaire à l'analyse de l'énoncé de l'apprenant. Dès lors, elle se préoccupe de réagir à des instructions exécutables ou non, à des questions auxquelles il est possible de répondre ou non, et aux énoncés dont le sens pose problème. Au regard des structures en UMI, la génération concerne la création de la nouvelle intervention du partenaire, c'est-à-dire le champ (H_{i+1}).

La génération concerne divers cas d'interaction. Pour réagir aux énoncés représentant des questions et interactions, elle s'appuie sur l'état de la tâche et celui du monde. Le cas où une instruction est reconnue recouvre trois cas différents. Le premier cas concerne les énoncés désignant des instructions exécutables, l'énoncé généré marque la faisabilité de l'action et son

accomplissement. A l'interface, l'instruction est exécutée. Le second cas marque un énoncé désignant une action en dehors de la tâche, le partenaire s'interroge sur la pertinence de cette action dans ce contexte et ne la réalise pas. Le troisième est le cas d'une instruction inexécutable vu l'état de la tâche ou du monde, l'énoncé généré marque cette impossibilité. Le cas des questions recouvre les questions comprises ou non comprises de l'apprenant.

Dans une optique totalement communicative, nos choix de *feedback* aux énoncés incorrects ne portent en rien sur la grammaire, mais voudraient insister sur les problèmes de compréhension sémantique des énoncés. Cette non compréhension vient essentiellement d'éléments lexicaux incompréhensibles donnés par l'analyse. L'idée est de répondre en donnant à l'apprenant des indices quant aux éléments posant problème dans son énoncé.

2.3.3 Lexique

La modélisation du lexique conditionne l'analyse, les possibilités de rétroaction, ainsi que la définition de la tâche applicative. L'idée de l'analyse est de faire émerger le sens rapidement en associant aux éléments des énoncés de l'apprenant une catégorie lexicale. Cette catégorisation conceptuelle dépend de la tâche applicative, dont la résolution est prétexte à l'interaction. La tâche applicative influe sur la création et la structuration du lexique (Figure 46).

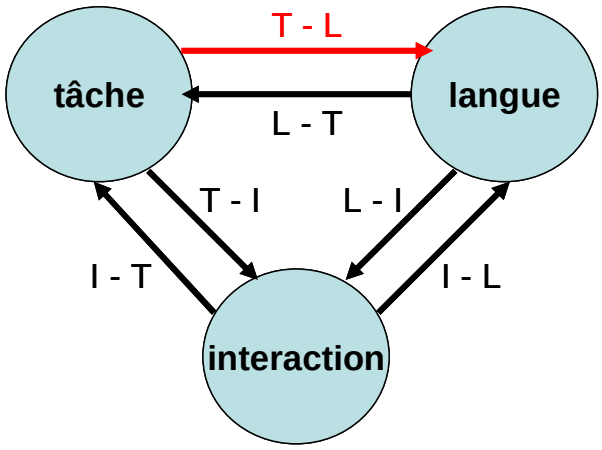


Figure 46 : Lien entre langue et tâche

Dans le cadre de notre activité, la langue utilisée est délimitée par la nature de la tâche applicative et le monde qui en découle. Cette partie de la langue, ce langage est donc un « sous-langage » au sens de Harris [Harris *et al.*, 1989], c'est à dire une partie de la langue

différent syntaxiquement et/ou lexicalement des caractéristiques d'autres parties de la même langue. Les sous-langages se caractérisent par un lexique limité et un nombre fini de schémas syntaxiques. Notre activité de production définit une situation de communication très spécifique avec une tâche applicative qui délimite très précisément un lexique (éléments de la cuisine, actions, …) mais aussi un nombre précis de motifs syntaxiques représentant les instructions à réaliser au cours de l'activité et les possibilités de questionnement de la part de l'apprenant.

La conception de la tâche applicative permet de repérer les éléments du lexique. La tâche applicative étant inspirée d'une tâche possible dans le monde réel, le lexique se trouve assez facilement. Etant donné la nature de la tâche, les formes syntaxiques employées sont relativement simples, représentant des instructions. Pour définir le lexique de l'activité, la méthode employée est de rechercher des textes de recette. En synthétisant ces textes en pensant aux interactions et donc à une représentation graphique future, nous déterminons la tâche applicative effective et par là même, un ensemble d'actions possibles et les objets associés à ces actions. Ces objets et actions déterminent le lexique et l'interface du monde. De cet ensemble d'actions et d'objets, on tire des catégories lexicales qui structurent ce lexique et permettent un certain degré de généralisation quant aux motifs syntaxiques.

Les catégories lexicales identifiées dans la tâche de réalisation d'une recette sont les suivantes : recette, contenu, ustensile, ingrédient, récipient, electro-ménager, localisateur, contenant, rangement, rangement-plat, attribut, action, ordre, question. Les catégories ingrédient, ustensile, rangement, electro-ménager, récipient, contenant se réfèrent à des objets bien identifiés dans le monde. Par exemple, les œufs sont un ingrédient. Les catégories contenant et localisateur rassemblent les mots pouvant indiquer des localisations ou la présence d'un contenu. Ainsi le mot « dans » est un localisateur. La catégorie attribut rend compte des mots en qualifiant d'autres. Nous y trouvons plus particulièrement les indications de localisation spatiale (haut, bas, gauche, droite,…). La catégorie recette renvoie aux références générales quant à la recette à accomplir, « le flan » en fait partie. Les catégories action, ordre et question sont les catégories des actes de langage attendus de la part de l'apprenant. Le mot prendre est catégorisé comme une action par exemple. La création des motifs syntaxiques est dirigée par la tâche applicative. Ainsi l'instruction "prendre les oeufs" est vue comme une action. Cette action se caractérise par l'association en un même motif

action du mot prendre et d'un élément relevant de la catégorie « ingrédient ». La figure ci-dessous représente l'organisation du lexique (Figure 47).

"le chocolat" → ⟨ingrédient⟩
"le placard" → ⟨rangement⟩
"prendre ⟨ingrédient⟩ ⟨localisateur⟩ ⟨rangement⟩" → ⟨action⟩
"prendre ⟨récipient⟩ ⟨localisateur⟩ ⟨rangement⟩" → ⟨action⟩
"prendre ⟨ustensile⟩ ⟨localisateur⟩ ⟨rangement⟩" → ⟨action⟩
"tu dois ⟨action⟩" → ⟨ordre⟩

Figure 47 : Lexique

Le lexique de la recette comprend 60 mots dont 18 représente des actions, 9 des ingrédients, 5 des ustensiles pour les catégories les plus importantes. Le lexique figure dans l'annexe C.

2.3.4 Une analyse, support à l'interaction : comment ?

L'analyse se déroule en deux grandes phases. Une première phase consiste à explorer les éléments de l'énoncé, à les catégoriser, à émettre des hypothèses sur leur nature, à essayer de les assembler pour créer des segments d'analyse plus grands. La seconde phase, à partir des données construites, cherche à dessiner les contours d'une analyse globale de l'énoncé. Le but de cette analyse est de permettre la continuation de l'interaction (Figure 48).

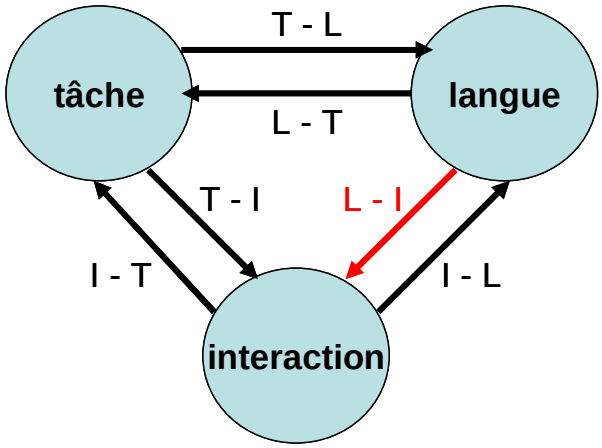


Figure 48 : Lien entre langage et interaction

2.3.4.1 Observation de l'énoncé

La première phase se déroule en quatre étapes :

- Création de « granules » à partir des mots de l'énoncé et du lexique
- Génération d'hypothèses par essai de catégorisation des groupes de mots non catégorisés sur la base des patrons syntaxiques du lexique
- Assemblage des granules en fonction des motifs syntaxiques pour découvrir la ou les interprétations des énoncés
- Application d'heuristiques afin de faire ressortir les interprétations les plus probables s'il y a des résultats concurrents

La première étape de l'analyse est donc la **granularisation des énoncés**, c'est-à-dire la génération de granules qui structurent l'énoncé à la fois d'un point de vue syntaxique et sémantique. Un granule est une structure de données qui rassemble des informations sur plusieurs mots de l'énoncé (Figure 49).

Texte :		Une suite de mots
Catégorie :		Une catégorie lexicale
Position :		La position dans l'énoncé
Couverture :		Le nombre de mots pris en compte
Hypothèses :		Le nombre de mots hypothétiques
Granules :		Liste d'éventuelles sous granules

Figure 49 : Granule

Ces informations sont les suivantes :

- Texte : différents vocables issus du lexique
- Catégorie : la catégorie lexicale à laquelle est rattachée la granule
- Position : place du premier élément reconnu dans l'énoncé
- Couverture : nombre de mots couverts par la granule dans l'énoncé
- Hypothèses : nombre de mots non reconnus
- Granules : liste des sous-granules rattachées à cette granule

Si nous prenons pour exemple « prendre les œufs dans le frigo », cet énoncé est subdivisé en quatre granules (Figure 50).

Figure 50 : Granularisation de l'énoncé « prendre les oeufs dans le frigo »

La première granule représente le texte « prendre » auquel est donné la catégorie « action ». Sa position est zéro, il s'agit du premier mot. Sa couverture est de un, un seul mot est couvert par cette granule. S'agissant d'un mot reconnu, son marqueur hypothèses est à zéro : il n'y a aucune hypothèse de portée sur ce mot. La seconde granule représente le texte « les œufs » : sa catégorie est « ingrédient », le premier mot du texte occupe la deuxième position dans l'énoncé, deux mots sont couverts par la granule, aucune hypothèse n'est portée sur ce mot. La troisième granule représente le texte « dans », la catégorie est « localisateur », la position trois et le marqueur d'hypothèses à zéro. La dernière granule représente le texte « le frigo », de la catégorie « rangement », le premier mot du texte occupe la position quatre dans l'énoncé et la granule couvre deux mots, son marqueur d'hypothèses est aussi à zéro. Les descriptions des granules un, deux et quatre sont les suivantes (Figure 46). Le champ granules n'apparaît pas car aucune de ces granules n'en possèdent.

Texte :	Prendre
Catégorie :	action
Position :	0
Couverture :	1
Hypothèses :	0

Texte :	Les oeufs
Catégorie :	ingrédient
Position :	2
Couverture :	2
Hypothèses :	0

Texte :	le frigo
Catégorie :	rangement
Position :	4
Couverture :	2
Hypothèses :	0

Figure 51 : Granules de l'énoncé « prendre les oeufs dans le frigo »

La **génération d'hypothèses** se fait à partir des granules trouvées et sur la base des motifs syntaxiques décrits dans le lexique. A partir des éléments reconnus, des hypothèses sont posées quant aux éléments les précédant, les suivant ou en observant deux granules différentes sur les éléments qui les séparent. Ces hypothèses, obtenues par comparaison les patrons syntaxiques de lexies qui comprennent des éléments reconnus, permettent de donner une ou plusieurs catégories possibles pour un élément non reconnu et de lui créer une granule correspondante. Si nous transformons l'exemple précédent avec une erreur sur « les œufs » que nous remplaçons par « les eggs », les granules correspondant à « prendre » et « le frigo »

sont inchangées. Par contre, des hypothèses sont posées sur « les eggs » (Figure 52). Dans les granules, les trois chiffres sous la barre centrale représentent respectivement la position, la couverture et le marqueur hypothèses de la granule.

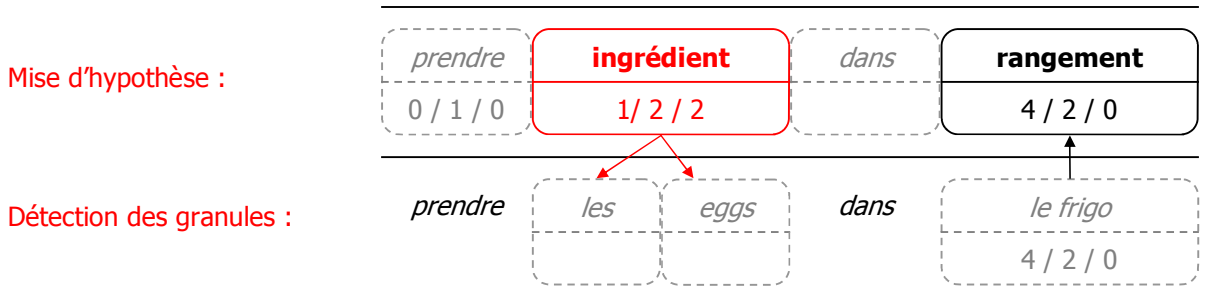


Figure 52 : Mise d'hypothèses sur « les eggs »

Trois granules hypothétiques sont créées car trois motifs syntaxiques sont applicables. En effet, « les eggs », dans ce contexte syntaxique, peuvent être un ingrédient, un ustensile ou un récipient (Figure 53).

Texte :	Les eggs
Catégorie :	ingrédient
Position :	2
Couverture :	2
Hypothèses :	2

Texte :	Les eggs
Catégorie :	ustensile
Position :	2
Couverture :	2
Hypothèses :	2

Texte :	Les eggs
Catégorie :	récipient
Position :	2
Couverture :	2
Hypothèses :	2

Figure 53 : Granules de l'énoncé « prendre les oeufs dans le frigo »

L'assemblage des granules est rendu possible par le positionnement contigu de deux granules et par la comparaison des éventuelles granules qui seraient créées avec les patrons syntaxiques du lexique. Quand une création a lieu, la nouvelle granule voit son champ texte devenir la concaténation des deux textes des granules fusionnantes, son champ catégorie prend la catégorie du patron syntaxique qu'elle décrit, son champ couverture est l'addition des deux couvertures respectives, son champ position prend la position la plus faible des deux granules, le champ hypothèse est la somme des hypothèses des deux granules. Si nous reprenons l'exemple « prendre les œufs dans le frigo », nous avons le schéma d'assemblage suivant (Figure 54). L'ensemble de l'énoncé est reconnu comme étant une action de six mots avec un marqueur d'hypothèses à zéro.

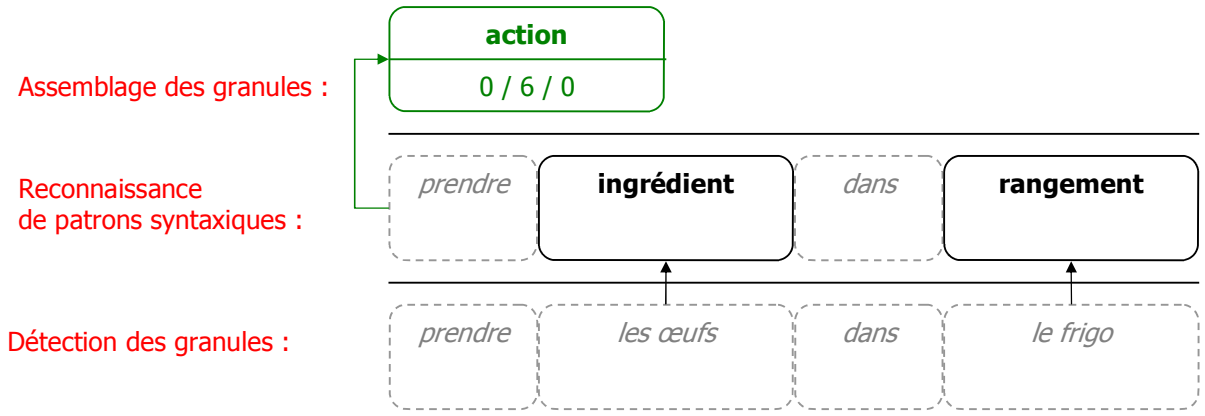


Figure 54 : Assemblage des granules de l'énoncé « prendre les oeufs dans le frigo »

De même, si nous prenons le cas de « prendre les eggs dans le frigo », l'assemblage se fait en fonction des hypothèses posées sur les éléments inconnus (Figure 55).

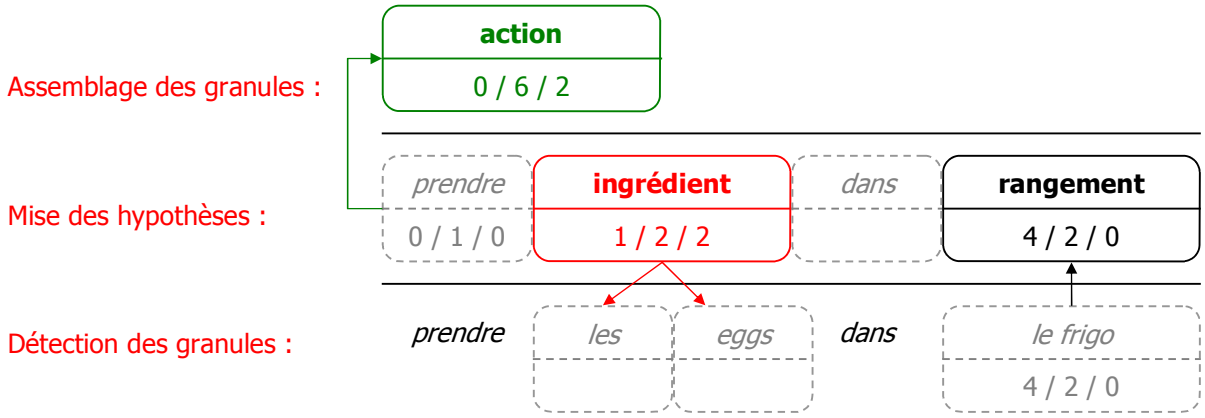


Figure 55 : Assemblage des granules de l'énoncé « prendre les eggs dans le frigo »

Les assemblages génèrent un grand nombre de résultats. Ce grand nombre de résultats est dû à la mise d'hypothèses sur des éléments qui sont en fait connus, à la création de résultats recouvrant seulement une partie de l'énoncé. Il convient alors d'effectuer la **recherche des meilleurs résultats** possibles en se fondant sur les couvertures et les hypothèses des granules obtenues. Nous avons quatre cas différents qui ont amené les heuristiques suivantes :

- Heuristique 1 :

Si deux résultats ont des couvertures égales, nous conservons le résultat avec le marqueur d'hypothèses les plus faible.

L'idée est ici de supprimer un résultat plus hypothétique qu'un autre à couverture égale. Si nous prenons l'énoncé « prendre les œufs dans le frigo », nous obtenons, parmi

l'ensemble des résultats, les deux résultats suivants (Figure 56). Le premier est le résultat obtenu avec « les oeufs » considérés comme un récipient. Ce résultat est obtenu lors de la mise d'hypothèses sur cet élément à partir des autres éléments reconnus, le marquer d'hypothèse est à 2. Le second résultat présente le même énoncé mais sur lequel aucune hypothèse n'a été posée. Le premier résultat est supprimé au profit du second.

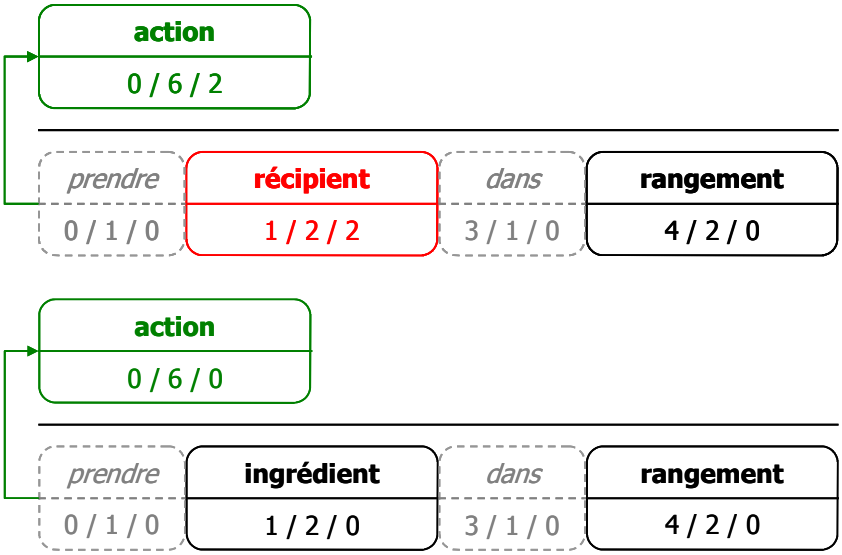


Figure 56 : Résultats à couvertures égales mais marqueurs d'hypothèse différents

- Heuristique 2 :

Si deux résultats ont des marqueurs d'hypothèse égaux, nous gardons le résultat de couverture maximale.

L'idée est de supprimer un résultat moins couvrant qu'un autre à taille égale. Avec l'énoncé « prendre les œufs dans le frigo », nous avons l'existence de ces deux résultats (Figure 57). Ces deux résultats ont le même marqueur d'hypothèse (0) mais le premier résultat couvre seulement la première moitié de l'énoncé alors que le second le couvre en entier.

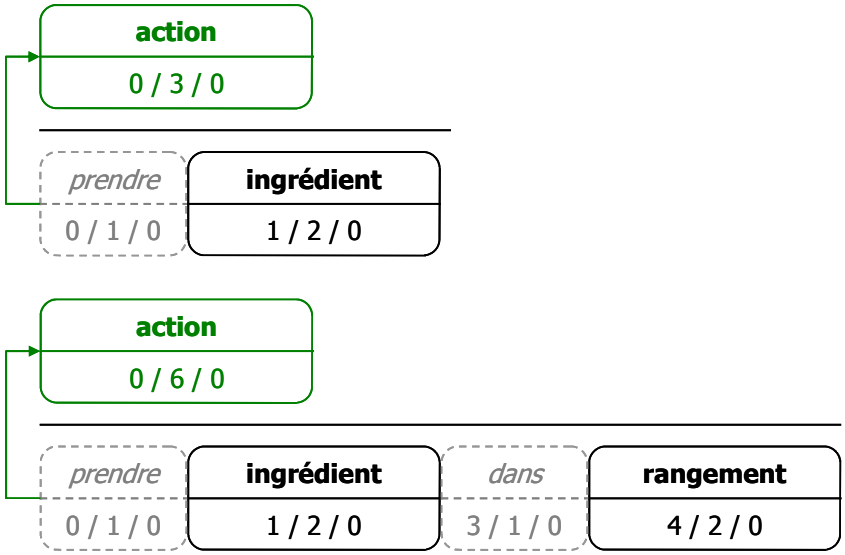


Figure 57 : Résultats avec marqueurs d'hypothèse égaux mais couvertures différentes

- Heuristique 3 :

Si un résultat a une couverture supérieure à un autre, ce dernier est supprimé si le gain en couverture est supérieur à l'addition des marqueurs d'hypothèse.

L'idée est de supprimer les résultats dont le taux d'hypothèse est faible mais qui ne couvrent pas assez l'énoncé, nous préférons le « risque » d'avoir une compréhension incomplète mais qui recouvre plus l'énoncé. Dans le cas de « prendre les eggs dans le frigo », les deux résultats qui suivent entrent dans ce cas (Figure 58). Le premier résultat est supprimé en faveur du second.

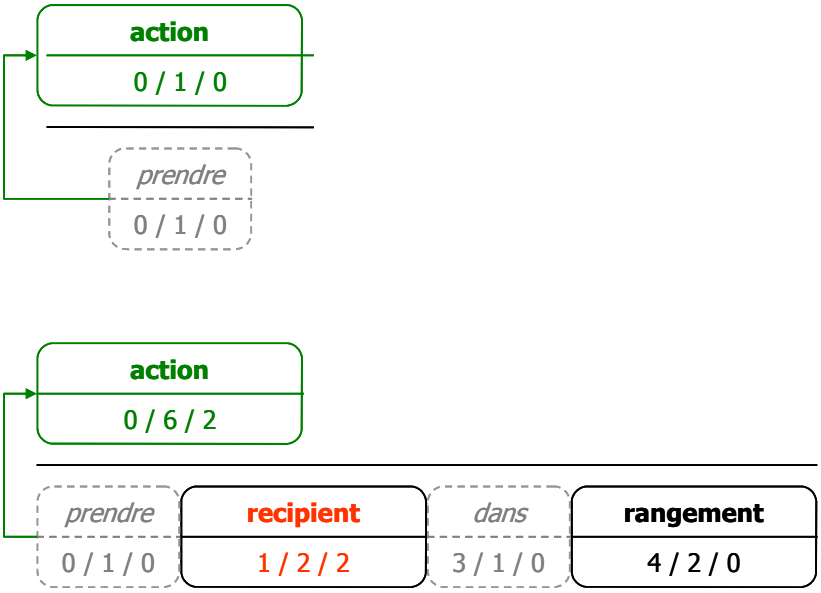


Figure 58 : Résultats mettant en concurrence couverture et hypothèse

- Heuristique 4 :

Si deux résultats sont identiques, l'un des deux est supprimé.

L'idée ici est de supprimer un des deux résultats qui présentent les mêmes couvertures et marqueurs d'hypothèse. Dans le cas de l'énoncé « prendre les eggs dans le frigo », ce cas est illustré ci-dessous (Figure 59). Un des deux résultats est supprimé dans le but de continuer l'échange conversationnel le plus naturellemnt possible.

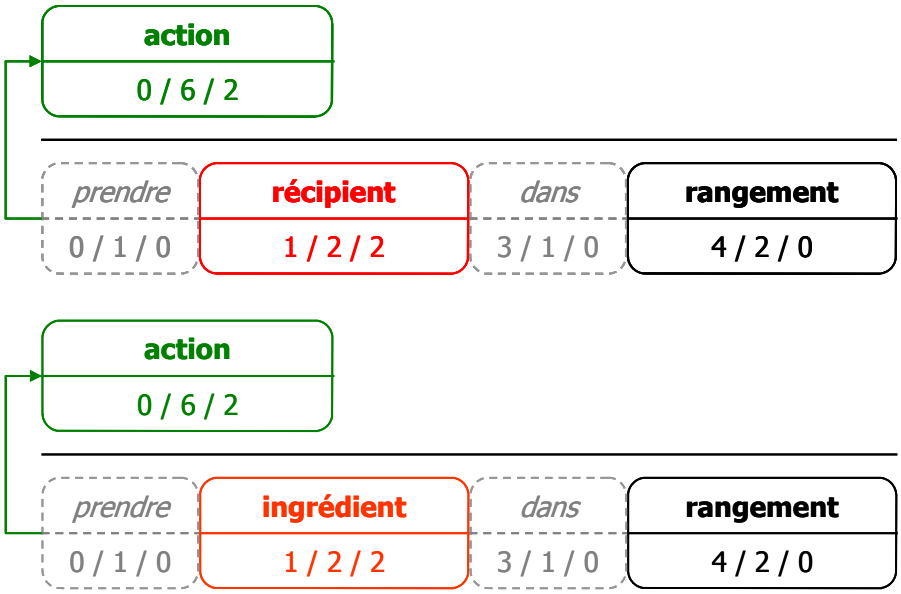


Figure 59 : Deux résultats identiques en couverture et hypothèse

Le but des différentes heuristiques est de supprimer les résultats et de faire émerger l'interprétation la plus vraisemblable, sans réarranger l'énoncé.

Nous pouvons résumer la première phase de l'analyse ainsi (Figure 60).

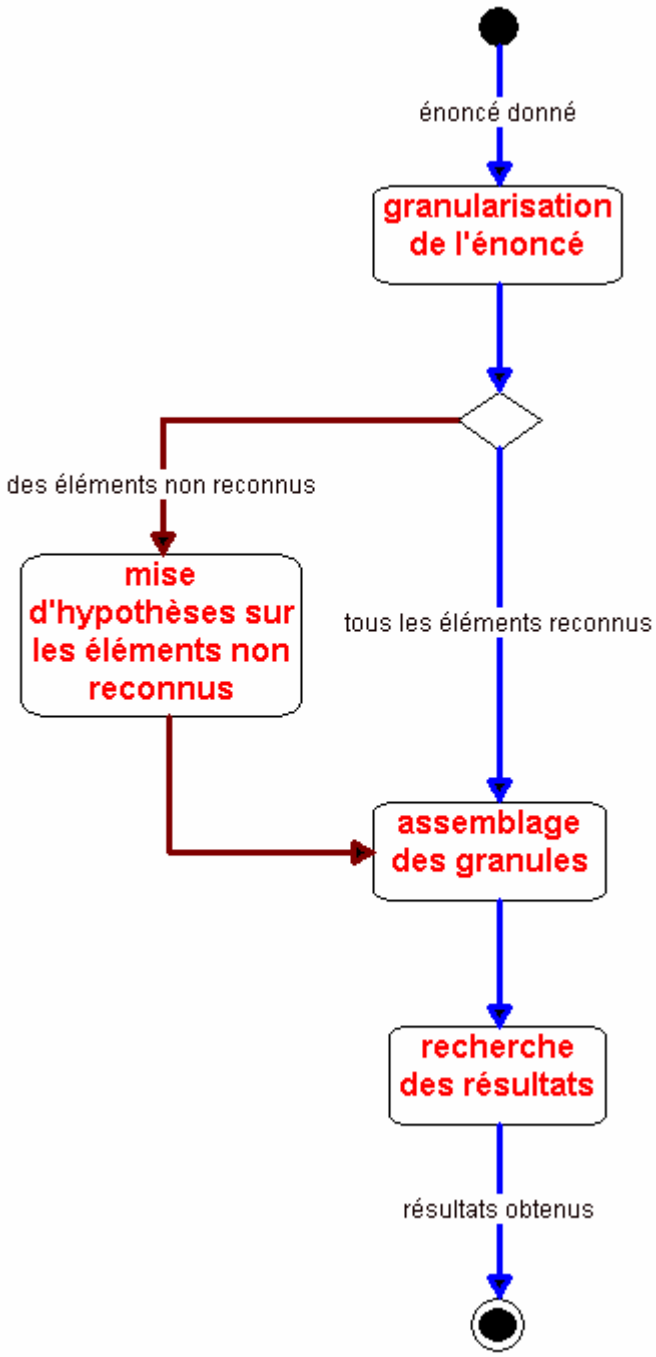


Figure 60 : Première phase de l'analyse, traitement des éléments de l'énoncé

La seconde phase de l'analyse comprend la vérification de la couverture lexicale, la vérification de la couverture syntaxique, et la recherche de correspondances syntaxiques. La **vérification de la couverture lexicale** consiste à vérifier si tous les mots de l'énoncé sont présents dans le lexique. Une bonne couverture lexicale est obtenue si tous les mots sont présents dans le lexique. Si tous les mots ne sont pas dans le lexique nous avons alors une mauvaise couverture lexicale. La **vérification de la couverture syntaxique** consiste à vérifier si l'on peut appliquer un motif syntaxique à l'énoncé. Une bonne couverture syntaxique indique qu'un motif syntaxique peut être appliqué avec ou sans hypothèse. Si un motif syntaxique ne peut être appliqué, ou que certains mots sont en dehors de la couverture syntaxique, on a alors une mauvaise couverture syntaxique. La **recherche de correspondances syntaxiques** consiste, elle, à réorganiser l'énoncé de façon à trouver une cohérence syntaxique (arriver à appliquer un motif syntaxique). Cette recherche réussit si on peut effectivement appliquer un motif après avoir réorganisé la phrase. Cette recherche échoue dans le cas contraire. Dans l'exemple suivant, nous avons une réorganisation de l'énoncé « prendre dans les œufs le frigo », cette réorganisation permet de découvrir un arrangement qui correspond effectivement à un patron syntaxique du lexique (Figure 61).

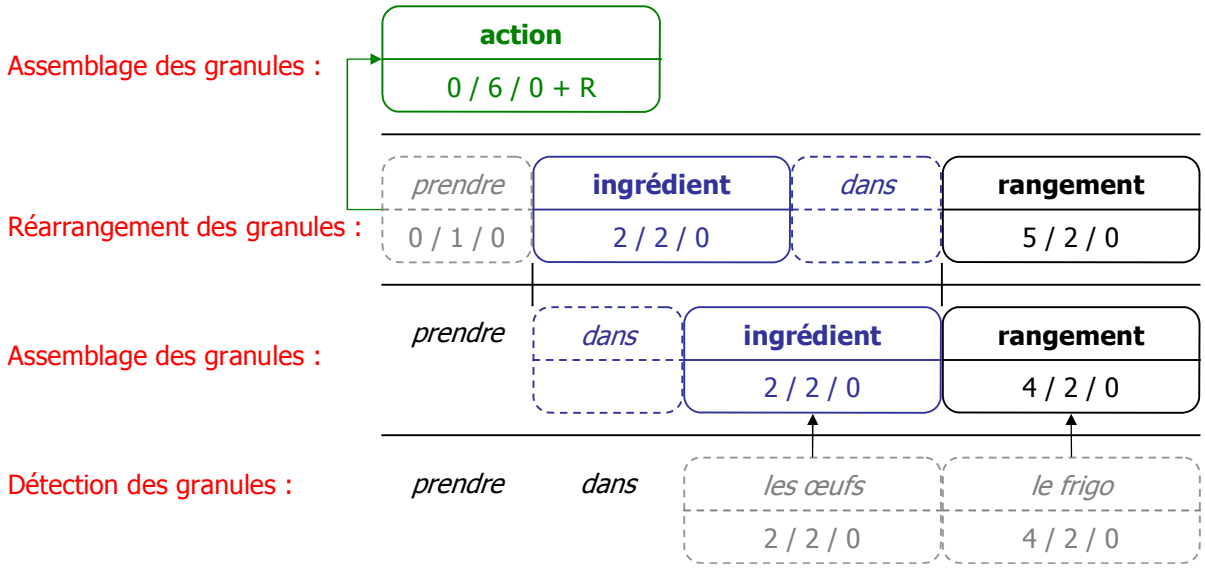


Figure 61 : Exemple de réorganisation dans le but de retrouver un patron syntaxique

Nous présentons dans le tableau suivant (Tableau 6) les différents cas auxquels est confronté l'analyseur. Nous présentons aussi un exemple pour chaque cas. Dans ces exemples, <*mot*> signifie que nous sommes en présence d'un mot inconnu qui n'est pas dans le lexique.

<table>
<tr><th colspan="3">Arbre de décision</th><th>Énoncés de l’apprenant</th></tr>
<tr><td rowspan="4">Bonne Couverture Lexicale</td><td colspan="2" rowspan="2">Bonne Couverture Syntaxique</td><td>Prendre les œufs dans le frigo (1)</td></tr>
<tr><td>Prendre (2)</td></tr>
<tr><td rowspan="2">Mauvaise Couverture Syntaxique</td><td>Avec Réarrangement</td><td>Prendre dans les œufs le frigo (3)</td></tr>
<tr><td>Sans Réarrangement</td><td>Les œufs le four (4)</td></tr>
<tr><td rowspan="5">Mauvaise Couverture Lexicale</td><td rowspan="2">Bonne Couverture Syntaxique</td><td>Avec Hypothèse</td><td>Prendre <mot> dans le frigo (5)</td></tr>
<tr><td>Sans Hypothèse</td><td>Prendre les œufs dans le frigo <mot> (6)</td></tr>
<tr><td rowspan="3">Mauvaise Couverture Syntaxique</td><td>Avec Réarrangement</td><td>Prendre dans les œufs <mot> le frigo (7)</td></tr>
<tr><td rowspan="2">Sans Réarrangement</td><td>Les œufs <mot> (8)</td></tr>
<tr><td><mot> (9)</td></tr>
</table>

Tableau 6 : Exemples de cas à traiter par l’analyseur

Les résultats obtenus sont alors dirigés soit vers l’étape de comparaison avec les tâches de l’arbre des tâches si des éléments sont reconnus, soit directement vers l’étape de génération d’une intervention du partenaire.

En résumé, l’analyse se déroule en deux grandes phases : une première phase qui permet de produire des résultats et de mettre des hypothèses sur les éléments non reconnus, travaillant à même les éléments des énoncés ; une seconde phase qui vérifie les couvertures lexicales, syntaxiques et recherche si besoin des cohérences syntaxiques pour dégager une interprétation globale de l’énoncé. Cette analyse ad-hoc suffit pour tester nos modèles d’interaction mais présente une grande limite en ce qui concerne l’apprentissage des langues : nous ne pouvons pas donner de diagnostic précis sur les compétences linguistiques des apprenants car l’analyseur ne possède pas de connaissances sur la langue en tant que système. Beaucoup de phénomènes linguistiques ne sont pas pris en compte (conjugaison, genres, pluriel, partitifs, flexions...). Ceci implique que nous ne pouvons interagir avec l’apprenant sur ces problèmes à l’heure actuelle. Ajoutons que nous avions initialement prévu d’intégrer en amont de notre analyse par « motifs syntaxiques » une véritable analyse morphosyntaxique. Malheureusement, nous n’avons pas pu récupérer le composant attendu en temps voulu. Nous

avons aussi proposé des stages sur des aspects linguistiques dans Sampras auprès de linguistes et/ou de didacticiens mais ces stages n'ont pas trouvé preneur. Nous pensons que notre approche de l'analyse n'est pas à remettre en cause mais plutôt l'absence d'un prétraitement morphosyntaxique. Une piste qui nous semble intéressante serait celle de recourir à une analyse de surface. Après une analyse morphologique, les mots d'une phrase sont regroupés pour obtenir des structures syntaxiques minimales, des *chunks*, proches des granules que nous avons créées. Ces chunks peuvent être des groupes nominaux, des verbes. Grâce à l'analyse actuelle, ces chunks pourraient être rattachés aux catégories lexicales que nous avons déterminées. La construction de la réaction pourrait alors jouer sur des éléments plus fins, éléments auxquels sont rattachées des informations concernant la morphologie, la syntaxe et une part de sémantique grâce à la catégorisation. Nous ne sommes pas allés plus loin dans cette réflexion pour l'instant.

2.3.4.2 Comparaison des énoncés à la description de la tâche

L'étape suivante consiste à établir à quelle catégorie appartient l'énoncé : instruction, question ou énoncé problématique. Les questions sont reconnues facilement de par leur structure syntaxique. Par contre, pour les instructions, il faut comparer les énoncés avec les descriptions des tâches de l'arbre des tâches. L'analyse délivrée est déjà porteuse du fait qu'elle sera comparée ou non. Le cas (9) du tableau précédent ne passe pas par cette étape car aucun élément n'a été reconnu. Par contre, l'ensemble des autres cas l'est puisque des éléments ont été reconnus. La comparaison se déroule selon deux cas précis : en ne prenant pas en compte les élément appartenant aux interactions précédentes, à la **mémoire de l'interaction,** ou en prenant en compte ces éléments.

Dans le cas où les éléments en mémoire ne sont pas pris en compte (soit parce qu'ils ne sont pas appropriés car inutiles à la comparaison de l'énoncé soit parce qu'il n'y en a pas), il existe deux possibilités. La première possibilité est celle où l'énoncé correspond à la description d'une tâche de l'arbre des tâches. Par exemple, l'énoncé « ouvrir le frigo » correspond parfaitement à la description d'une tâche de l'arbre des tâches, la tâche d'ouvrir le frigo. En observant le Tableau 6, les cas (1) (3) (6) (7) tombent dans ce cas de figure. Le traitement continue par la génération de la réaction du partenaire. La seconde possibilité est celle où l'énoncé est imparfait, cet énoncé entraîne une incidence ; les cas (4) (5) (8) du Tableau 6 correspondent à cette situation.

Dans le cas où les éléments en mémoire sont pris en compte, les deux possibilités se reproduisent avec tout de même de notables différences. Les cas (1) (3) (6) (7) ne sont pas concernés car déjà comparés avec l'arbre des tâches. La correspondance avec l'arbre de tâche peut alors parvenir à trouver des correspondances entre des éléments des énoncés ajoutés de ceux de la mémoire tels que dans les cas (2) (8). Par exemple, si dans la mémoire, nous avons un indice « le frigo », et que l'énoncé est « l'ouvrir », une correspondance se trouve entre ces éléments réunis et la tâche « ouvrir le frigo ». Par contre si l'élément en mémoire est « le chocolat », l'énoncé « ouvrir » sera sans conséquence car il n'existe pas de tâche « ouvrir le chocolat ». Dans les cas où même une recherche dans la mémoire de l'interaction ne permet pas d'identifier une tâche, une incidence prend place.

Dans le cas des incidences, avant la phase de génération se trouve la phase de création d'indices mémoires et de recherches d'indices de compréhension. La première étape, qui n'a lieu que dans le cas où les éléments en mémoire n'ont pas été pris en compte lors de la contextualisation (car il n'y en avait pas le plus souvent) est de stocker les éléments reconnus dans la mémoire de l'interaction comme des **indices mémoire**. Ces éléments serviront peut-être plus tard à la résolution d'énoncés. Ce cas peut intervenir dans l'exemple 5 issu du Tableau 6, l'énoncé « Mettre le truc au frigo » produit deux indices mémoires : l'action « mettre » et le rangement « le frigo » qui sont conservés en vue d'interpréter de futurs énoncés.

La seconde étape, celle là obligatoire, est d'effectuer une recherche d'indices quant à ce que l'apprenant a voulu dire. A partir des éléments reconnus dans les énoncés de l'apprenant et en interaction avec le contenu de la mémoire, cette recherche permet de créer des **indices de compréhension**. Ces indices sont des phrases presque formées à partir des éléments reconnus. Ce sont notamment des questions. Ces indices de compréhension sont fournis au module chargé de la génération. Par exemple, si nous avons dans un énoncé un élément non reconnu (symbolisé par le ?) et un élément reconnu comme étant un ingrédient, nous avons la création de l'indice de compréhension suivant (Figure 62).

[?] —— **[ingredient]** ⟶ **Que veux-tu faire avec [ingredient] ?**

Figure 62 : Création d'un indice de compréhension à partir d'un élément reconnu

Dans le cas de plusieurs éléments reconnus, un ensemble de formulations dépendant de ces éléments reconnus est à imaginer. Si l'on reprend l'exemple 5 issu du Tableau 6, les éléments reconnus sont l'action « prendre » et le rangement « le frigo », l'indice de compréhension généré est « Que veux-tu prendre dans le frigo ?», ce qui correspond au schéma de création d'un indice de compréhension suivant (Figure 63).

Prendre ⟨ **dans [rangement]** / **[?]** ⟶ **Que veux-tu prendre dans [rangement] ?**

Figure 63 : Création d'un indice de compréhension à partir d'une action et d'un rangement reconnus

Il existe un grand nombre de constructions d'indice de compréhension à partir d'un ou plusieurs éléments reconnus. Nous ne les décrivons pas toutes mais en voici le fonctionnement global. Dans le cas d'un seul élément reconnu, si cet élément est une action alors l'indice de compréhension sera « Que veux tu [action] ? ». Si l'élément reconnu est d'une autre catégorie alors l'indice de compréhension sera « que veux-tu faire avec [element] ? ». Dans le cas de plusieurs éléments reconnus, il existe également deux cas. Soit l'un des éléments est une action alors l'indice créé reprend l'action à laquelle elle adjoint l'autre élément reconnu si cet élément permet la création d'une question compréhensible. L'autre cas est celui où nous avons deux éléments non reconnus dont aucun n'est une action : l'indice de compréhension créé s'appuie sur l'un de ces deux éléments. Nous récapitulons ces cas dans le tableau suivant (Tableau 7).

[?] —— [ingredient]	**Que veux-tu faire avec [ingredient] ?**
[?] —— [action]	**Que veux-tu [action] ?**
Prendre — dans [rangement] / [?]	**Que veux-tu prendre dans [rangement] ?**
[?] — [rangement] / [ingredient]	**Que veux-tu faire avec [rangement] ?** **ou** **Que veux-tu faire avec [ingredient] ?**

Tableau 7 : Création des indices de compréhension

Il est à noter que le travail de construction de ces indices de compréhension n'est pas simple et, pour l'instant, n'a concerné que les énoncés utiles à la réalisation de la tâche. Il faut en effet considérer des problèmes tel que le genre et le nombre des indices, ou les arrangements possibles entre les différents éléments reconnus selon leurs catégories lexicales. Ainsi un énoncé tel que « prendre le pot » entraîne la question suivante : « quel pot veux-tu prendre ? ». Si par exemple, nous avons « prendre la bouteille », la question devient alors : « quelle bouteille veux-tu prendre ? ».

Des formes de généralisations sont sans doute envisageables. Ces généralisations permettraient à partir du genre, du nombre d'envisager des questionnements standards. Mais la langue recèle aussi d'exceptions qu'il serait peut-être difficile de gérer. Pour l'instant, nous n'avons pas été plus loin sur le sujet.

2.3.4.3 Génération

La génération des actions du partenaire vient après les phases précédentes. A partir de l'analyse, de la contextualisation et des indices d'incompréhension éventuellement générés, la nouvelle intervention du partenaire est créée. D'une part, il y a les réactions aux énoncés bien formés (ou du moins reconnus comme tels) représentant des instructions

complètes ou complétées et des questions bien formées. Ces interventions s'appuient sur les connaissances de la tâche applicative (l'état de l'avancement de la tâche) et celles du monde (l'état du monde). Si nous reprenons le mécanisme interactionnel défini dans la partie interaction (§ 2.2.3.2), nous avons les cas suivants d'énoncés :

- l'énoncé est une instruction ou complète une instruction :
    - Elle est réalisable……**La réaliser**

L'analyse pragmatique a fourni en résultat une tâche identifiée à partir de l'énoncé. L'état du monde est tel que cette tâche est réalisable, l'état de la tâche est tel que la tâche est exécutable. Dans la description de la tâche identifiée se trouve un énoncé qui indique la réalisation de cette tâche et aussi la manipulation à opérer dans la cuisine. La réaction générée est alors l'affichage de l'énoncé dans la bulle du partenaire et l'exécution de la manipulation. Si nous prenons pour exemple l'énoncé « ouvrir le frigo », si le frigo est fermé alors cette tâche est exécutable. La réaction du partenaire est l'ouverture du frigo et l'énoncé « j'ouvre le frigo » (Figure 64).

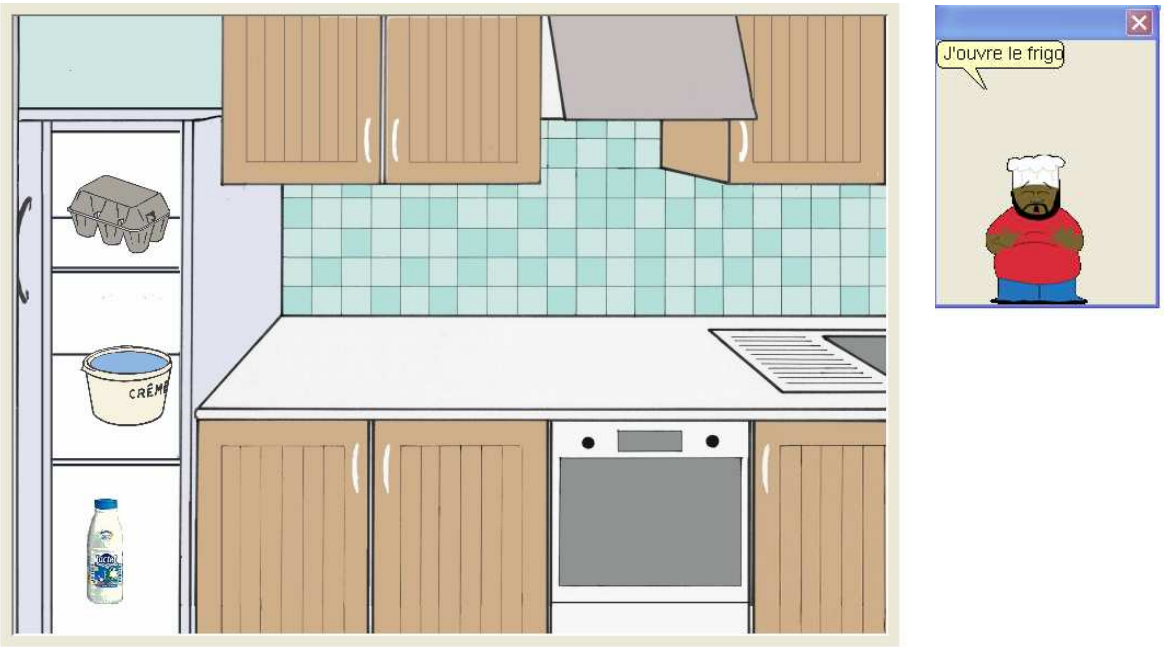


Figure 64 : Interface après la génération de la réaction du partenaire à l'énoncé « ouvrir le frigo »

- Elle n’est pas dans la tâche..... ……**Le dire**

L’analyse a fourni en résultat un énoncé correct au vu des descriptions du lexique. Mais cet énoncé ne correspond à aucune tâche. La réaction générée est l’affichage de l’énoncé « Je ne vois pas pourquoi je ferai cela ». Un exemple pourrait être l’énoncé « déposer les œufs dans le frigo », cet énoncé est correct mais ne correspond aucunement à une tâche de l’arbre des tâches.

- Elle ne peut pas être accomplie……...……**Le dire**

L’analyse pragmatique a fourni en résultat une tâche identifiée à partir de l’énoncé. L’état du monde ou l’état de la tâche est tel que la tâche n’est pas exécutable. La réaction générée est alors l’affichage de l’énoncé « Je comprends bien ton instruction, mais je ne peux pas l’exécuter » dans la bulle du partenaire. Si nous prenons par exemple l’énoncé « ouvrir le frigo » alors que le monde est dans l’état présenté Figure 64, le frigo est déjà ouvert et ne peut être ouvert de nouveau.

- l’énoncé est une question :
  - On peut y répondre..... ……**Y répondre**

L’analyse donne en résultat une question. L’état du monde permet de répondre à la question. La réaction du partenaire est la réponse à cette question. Si par exemple, l’apprenant demande « Où sont les œufs », si les œufs sont dans le frigo, alors le partenaire répond : « les œufs sont dans le frigo ».

- On ne peut pas y répondre……**Le dire**

L’analyse donne en résultat une question. L’état du monde ne permet pas d’y répondre. Le partenaire réagit par « Je ne comprends pas ta question ». Nous avons ce cas lorsque la question est, par exemple, « Où est la fenêtre ?» et qu’il n’y a pas de fenêtre dans le monde.

- l'énoncé n'est pas compris :
    - Avec des indices sur ce que l'apprenant signifie…**Poser des questions selon les indices**

L'analyse donne en résultat un énoncé non compréhensible mais des éléments de l'énoncé ont été reconnus et des indices de compréhension produits. Le partenaire marque sa non compréhension de l'énoncé mais interroge l'apprenant grâce aux indices de compréhension constitués. Par exemple, si on a l'énoncé « ouvrir », l'indice de compréhension généré est « Que veux-tu ouvrir ? ». La réaction du partenaire est alors « Je ne te comprends pas lorsque tu écris « ouvrir ». Que veux-tu ouvrir ? ».

- Sans indices sur ce que l'apprenant signifie…….**Le dire**

L'analyse ne reconnaît aucun des éléments de l'énoncé, ex : Le partenaire marque sa non compréhension par le simple « Je ne te comprends pas lorsque tu écris xxx ».

La génération permet, non seulement, de donner un énoncé et une réaction au partenaire mais influe également sur la continuité de l'action, entraînant ou non des incidences.

2.4 Synthèse

Ce chapitre s'attache à la description des modèles créés pour la conception de l'environnement informatique. Trois dimensions caractérisent cette conception : tâche, interaction et langue.

Concernant la dimension tâche, nous proposons les définitions des tâches pédagogiques, c'est à dire les activités dans lesquelles l'apprenant va être impliqué. Ces tâches permettent d'exercer des compétences en langue : compréhension et production. Nous définissons les tâches interactives, instanciations des tâches pédagogiques dans le micromonde. Nous décrivons aussi la tâche applicative, prétexte à l'interaction, en particulier les caractéristiques en termes d'interface, de lexique et de représentation en machine de cette tâche.

Concernant la dimension interaction, nous proposons des types d'interaction et une organisation de l'interaction. Les types d'interaction se regroupent dans deux catégories : ceux qui font avancer la réalisation de la tâche et ceux qui sont liés au feedback donné à l'apprenant lorsqu'un problème survient dans l'interaction. Les échanges s'organisent selon des modèles inspirés de travaux en dialogue homme-machine, modèles qui permettent de suivre et caractériser les échanges qui ont lieu. Nous présentons notamment un ensemble de tactiques dialogiques de réponse aux énoncés de l'apprenant.

Concernant la dimension langue, nous proposons un modèle d'analyse qui permet d'accéder au « sens » à partir d'analyses partielles de l'énoncé de l'apprenant et hypothèses portant sur les éléments non reconnus de ce même énoncé. Le modèle de génération proposé s'appuie sur les éléments fournis par l'analyse, l'état du monde et de la tâche pour créer les réactions du partenaire virtuel aux énoncés de l'apprenant. L'analyse ne permet pas un réel diagnostic des compétences linguistiques d'un apprenant mais supporte l'interaction en permettant sa continuation.

Le chapitre suivant présente comment ces modèles ont été implémentés dans l'environnement Sampras.

Chapitre 3
L’environnement Sampras

3.1 ARCHITECTURE 117
3.1.1 ARCHITECTURE DES SYSTEMES DE DIALOGUE 117
3.1.2 ARCHITECTURE DE SAMPRAS 118
3.2 ASPECTS LIES A L'INTERFACE 120
3.2.1 PARTIE GRAPHIQUE 120
3.2.1.1 Présentation 120
3.2.1.2 Fonctionnement 121
3.2.2 MODULE INTERFACE 124
3.2.2.1 Représenter le monde 124
3.2.2.2 Gérer les interventions du partenaire et de l'apprenant 125
3.3 ANALYSE 129
3.3.1 LES CONNAISSANCES LEXICALES 129
3.3.2 LE MODULE ANALYSE 132
3.3.2.1 Exploration de l'énoncé 133
3.3.2.2 Vérifications et manipulations 136
3.4 LA TACHE APPLICATIVE 137
3.4.1 REPRESENTER LA TACHE 137
3.4.2 GERER LA TACHE 140
3.5 MODULE GENERATION 142
3.5.1 RECHERCHE DE LA TACHE SANS INFLUENCE DES INTERACTIONS PRECEDENTES 142
3.5.2 RECHERCHE DE LA TACHE AVEC INFLUENCE D'INTERACTIONS PRECEDENTES 148
3.6 MODULE INTERACTION 151
3.6.1 DESCRIPTION 151
3.6.2 FONCTIONNEMENT 154
3.7 MISES A L'ESSAI 157
3.7.1 PREMIERES MISES A L'ESSAI, HIVER 2004-2005 158
3.7.1.1 Déroulement des mises à l'essai 158
3.7.1.2 Observations 159
3.7.2 DEUXIEMES MISES A L'ESSAI, HIVER 2005-2006 161
3.7.2.1 Déroulement des mises à l'essai 161
3.7.2.2 Observations 162

Ce chapitre décrit l'environnement développé à partir des modélisations vues dans le chapitre précédent. Il décrit aussi les mises à l'essai qui ont eu lieu. Nous présentons tout d'abord l'architecture générale choisie pour réifier nos modèles. Nous détaillons ensuite les composants de cette architecture.

3.1 Architecture

L'architecture de l'environnement Sampras articule deux grandes parties (Figure 65) : une interface graphique développée en Java, et un système à base de connaissances qui repose sur le moteur d'inférences Jess (Java Expert System Shell). Les données de l'interface, comme celles du moteur d'inférences sont codées de façon déclarative : le monde virtuel présenté à l'apprenant (décrit dans un fichier XML) et l'ensemble des connaissances qui assurent la gestion de l'interaction tout au long de l'activité. Ces connaissances, ainsi que la communication entre l'interface graphique et le système à base de règles sont explicitées dans ce qui suit.

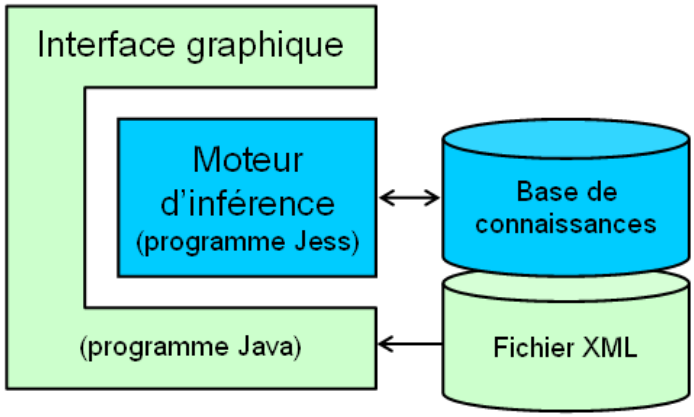


Figure 65 : Architecture en deux parties de SAMPRAS

3.1.1 Architecture des systèmes de dialogue

L'architecture la plus communément mise en œuvre pour les systèmes de dialogue est de type modulaire [Baker *et al.*, 2006], les modules étant activés de façon séquentielle de la réception d'un énoncé de la part de l'utilisateur jusqu'à la génération de la réaction à l'intention du même utilisateur (Figure 66). **Le contrôleur du dialogue** a pour but

d'organiser l'interaction sur la base d'un modèle de dialogue. A partir d'un modèle de la langue, le **module analyse** interprète les énoncés de l'utilisateur. Le module **contrôleur de la tâche** gère l'avancement de l'activité en cours sur la base d'un modèle de la tâche. Le **module génération** construit les rétroactions du système vers l'utilisateur.

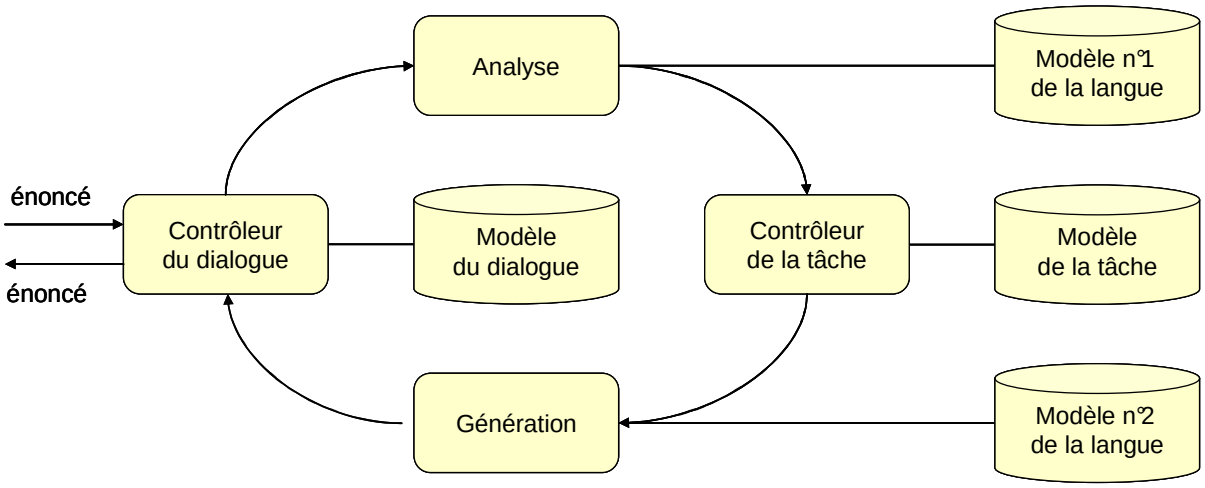


Figure 66 : Architecture modulaire, extrait de [Baker *et al.*, 2006]

D'autres architectures existent, comme par exemple les architectures distribuées ou multi-agents. Nous n'évoquons ici que l'approche modulaire que nous avons retenue dans le cadre de notre étude.

3.1.2 Architecture de Sampras

Partant de cette architecture, nous l'avons adaptée en fonction de nos besoins et de nos choix, notamment au niveau du contrôle de la tâche par le module interaction (Figure 67). En effet, ce dernier contrôle l'exécution de la tâche.

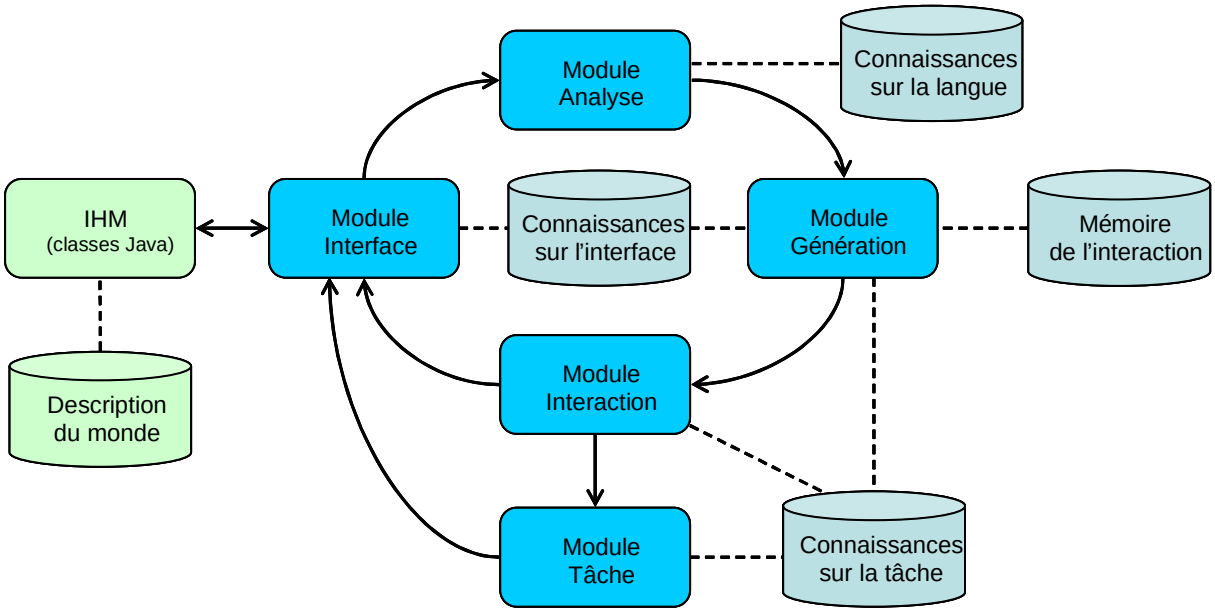


Figure 67 : Architecture de SAMPRAS

- Le **module interface** est chargé du lien entre l'interface graphique (programme Java) et les autres composants du dialogueur. Il recueille les actions (verbales et non-verbales) de l'apprenant et permet la manipulation des objets de l'interface par le partenaire, ainsi que l'affichage de ses énoncés. Il met à jour les **connaissances sur l'interface**.
- Le **module analyse** est chargé d'interpréter les énoncés de l'apprenant. Il prend en entrée une chaîne de caractères et produit une structure censée consigner le sens littéral de l'énoncé. Il repose sur des **connaissances sur la langue** décrites dans un lexique qui contient les termes et formes attendus au cours de la réalisation de l'activité.
- Le **module génération** est chargé de préparer les énoncés du partenaire (ceux-ci étant finalisés dans le module interaction) à partir de l'analyse des énoncés de l'apprenant. Il s'appuie sur des **connaissances sur la tâche**, sur des **connaissances sur l'interface** ainsi que sur une **mémoire de l'interaction**.
- Le **module interaction** gère l'interaction tout au long de l'activité. Il répartit les tours de parole et commande le module interface en lui envoyant les manipulations à faire et les énoncés à afficher. Il commande aussi le module tâche en provoquant l'exécution des tâches identifiées lors de l'étape de génération.
- Le **module tâche** gère l'avancée de la tâche applicative. Il met à jour les **connaissances sur la tâche** : un arbre de tâches qui permet de connaître les actions possibles et autorisées

à tout moment. De plus, il transmet au module interface les éventuelles manipulations à effectuer sur l'interface graphique.

La communication entre ces modules (à l'exception du module tâche) se fait par activation successive. Le module tâche est directement contrôlé par le module interaction.

Lors de l'activité de production, le module interface reçoit un énoncé de l'apprenant délivré par l'interface graphique. Cet énoncé est transmis au module analyse qui délivre une représentation à l'intention du module génération. Ce dernier donne au module interaction des éléments utiles afin de construire la réaction du partenaire. Le module interaction commande alors le module tâche, ordonnant ou non, l'exécution d'une instruction et envoie la réaction du partenaire au module interface. Le module tâche met à jour l'arbre de tâche et, le cas échéant, déclenche la manipulation d'objets à l'interface. La partie connaissance compte 250 règles.

3.2 Aspects liés à l'interface

3.2.1 Partie graphique

Comme le signifie la Figure 65, la partie graphique et la partie « connaissance » sont dissociées. L'interface graphique est donc indépendante de l'interaction et de la résolution de la tâche. L'activité que nous avons choisi de mettre en œuvre (activité de production) peut donc être remplacée par une autre activité (activité de compréhension, par exemple) sans adaptation au niveau du code de l'interface. De plus, le « moteur graphique » que nous avons développé en Java est lui-même générique : le monde est totalement décrit dans un fichier XML passé en paramètre à l'application.

3.2.1.1 Présentation

La définition de cette interface, (cf. § 2.1.3), fait apparaître trois zones distinctes (Figure 68) :

- Le monde (ici la cuisine) : zone où s'effectue l'activité. Les manipulations des objets (ouvertures, fermetures, changements d'état des objets en fonction de la tâche, *etc.*) y prennent place.

- Le partenaire (ici un chef cuisinier) : personnage qui dispose d'une bulle type bande dessinée dans laquelle les énoncés du système sont affichés.
- La zone de texte (ici un rouleau à pâtisserie) : champ de saisie qui permet à l'apprenant de rentrer ses propres énoncés.

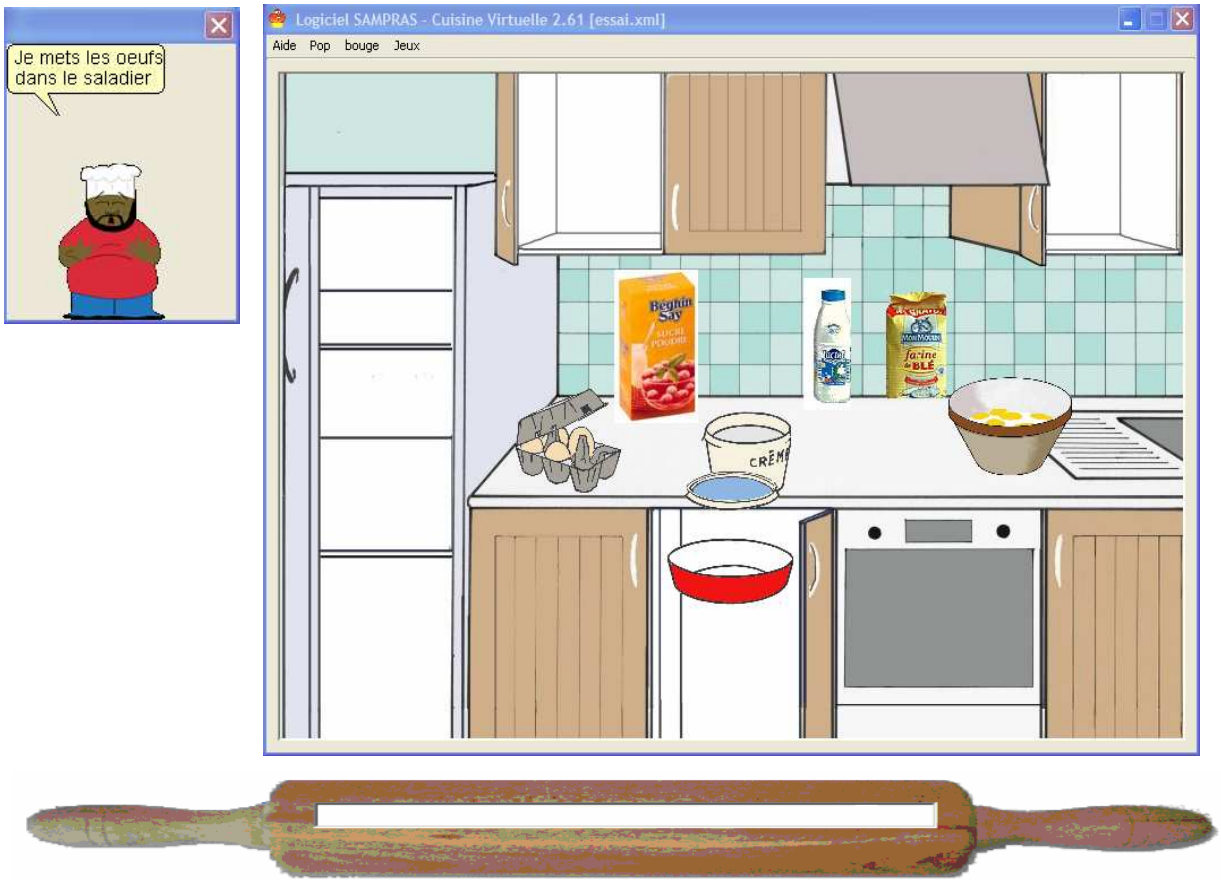


Figure 68 : Interface homme-machine

3.2.1.2 Fonctionnement

L'application construit le monde à partir des données du fichier XML dont nous proposons quelques extraits montrant la nature des différents éléments que l'on peut y trouver (Figure 69). Ainsi le monde possède un **fond** (le décor), fond sur lequel se superposent des objets et des zones aux propriétés particulières. Il existe différentes catégories d'objets : des objets **déplaçables**, des objets **ouvrables** et des objets **modifiables**. Chaque objet possède son identifiant et une position. Dans la figure précédente, nous pouvons voir des objets ouvrables (frigo, placards, four), des objets déplaçables (ingrédients, ustensiles), des objets modifiables (le saladier et le moule dont les représentations graphiques vont changer en fonction de l'avancement de la recette).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE monde SYSTEM "monde.dtd">
<monde fond="cuisine.jpg">
    [...]
    <zone ident="le bouton du four"
          x="517" y="381"
          largeur="15" hauteur="15"/>
    [...]
    <deplacable ident="le cacao"
                x="379" y="42"
                image="cacao.jpg"
                visible="non"/>
    [...]
    <ouvrable ident="le placard n°1"
              x="186" y="0"
              fond="plac1_ouv.jpg"
              porte="plac1_fer.jpg"/>
    [...]
    <modifiable ident="la balance"
                x="208" y="438"
                etat="1"
                visible="non">
          <etat ident="1" image="balance_1.gif"/>
          <etat ident="2" image="balance_2.gif"/>
    </modifiable>
    [...]
</monde>
```

Figure 69 : Extrait du fichier XML de description du monde

La classe centrale de l'application est la classe MicroWorld (Figure 70) qui possède en attributs le moteur d'inférence (classe MWJess), le partenaire (classe Partenaire) et une instance de la classe MWLayeredPane qui est la zone de manipulation qui contient tous les objets du monde. La classe Partenaire possède en attribut la bulle où sont affichées ses interventions.

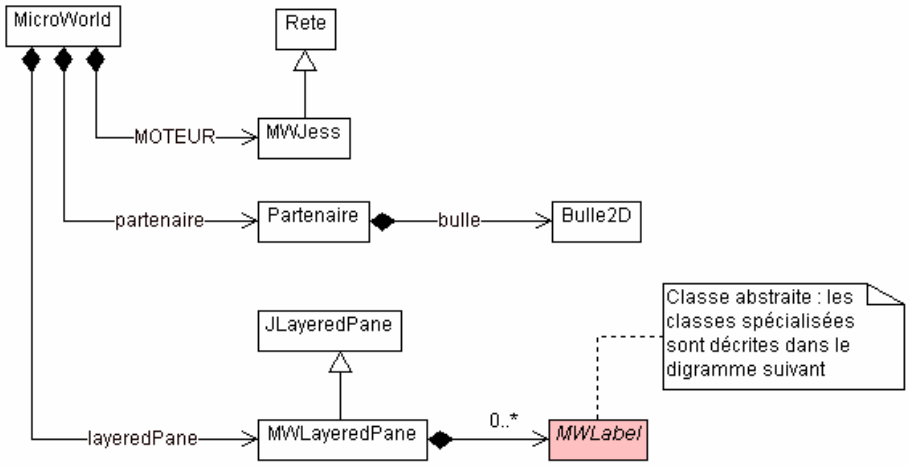


Figure 70 : Diagramme de classe simplifié de Sampras à partir de la classe MicroWorld

Aux différentes catégories d'objets correspondent des classes : Objet_zone pour ceux représentant des zones précises de la cuisine (les boutons par exemple), Objet_simple pour des objets sans particularités ou implications dans la tâche applicative, MWLabel_dep pour les déplacables, MWLabel_ouv pour les ouvrables, MWLabelState pour les modifiables.

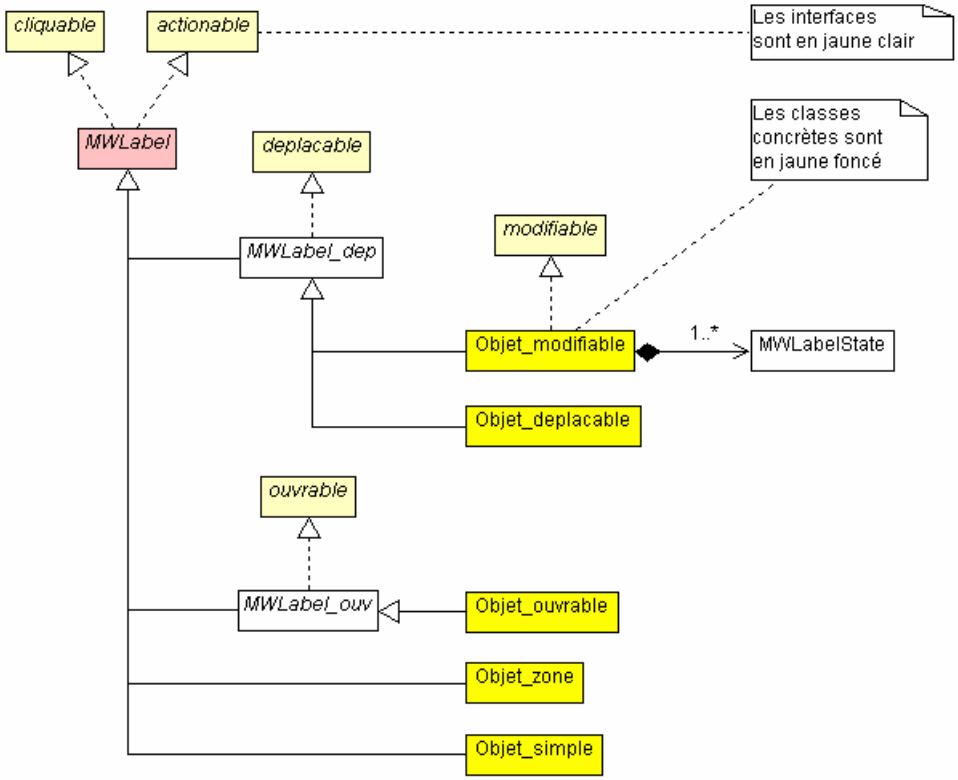


Figure 71 : Diagramme de classe de Sampras à partir de la classe MWLabel

3.2.2 Module interface

La communication entre la partie graphique et le module interface est réalisée sur le principe des *shadow facts* proposée par le package Jess. Le principe repose à la fois sur l'utilisation des *javabeans* et de la classe PropertyChangeSupport qui permet de générer un évènement lorsque la valeur d'un attribut d'un objet est modifiée. Cela permet d'associer des objets Java à des faits de la base de connaissances du moteur d'inférences. Ainsi, chaque objet de l'interface (ouvrables, déplaçables, *etc*.) possède à la fois une composante graphique et une composante symbolique, stockée dans la base de faits. Cela permet de maintenir assez facilement la cohérence entre ce qui est affiché à l'écran et la représentation du monde qui est à la base des inférences.

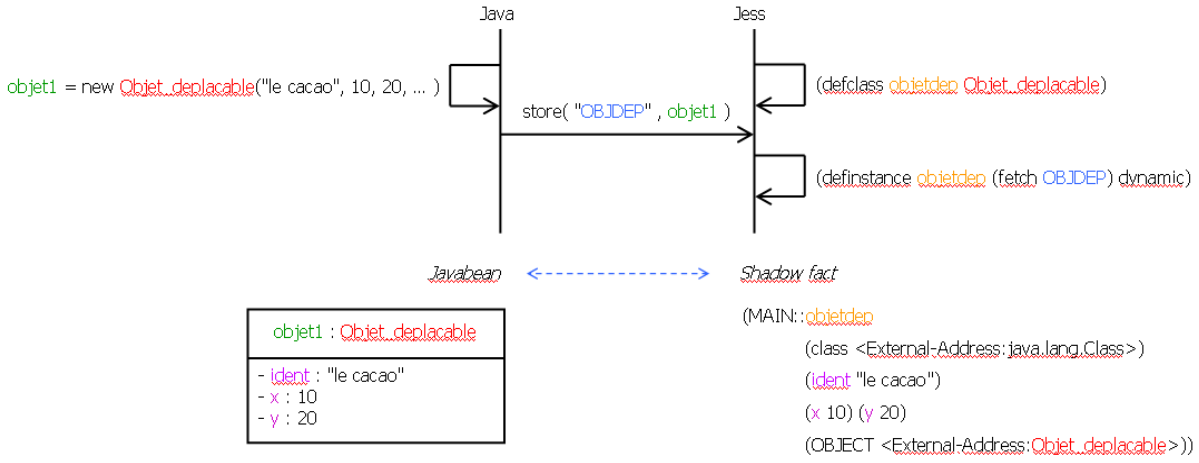


Figure 72 : Principe de création d'un « *shadow facts* »

3.2.2.1 Représenter le monde

Ainsi, le système maintient une cohérence entre ce qui est présenté à l'interface et la base de faits par l'intermédiaire des *shadow facts*. Lorsque le monde virtuel est initialisé à partir des données lues dans du fichier XML, la base de faits décrivant les objets et leurs attributs est automatiquement instanciée.

```
(MAIN::objetzone
      (class <External-Address:java.lang.Class>)
      (ident "le bouton du four")
      (image nil)
      (position <External-Address:java.awt.Point>)
      (OBJECT <External-Address:microworld.Objet_zone>))
```

```
(MAIN::objetdep
      (class <External-Address:java.lang.Class>)
      (ident "le cacao")
      (image <External-Address:sun.awt.windows.WImage>)
      (position <External-Address:java.awt.Point>)
      (OBJECT <External-Address:microworld.Objet_deplacable>))
```

Figure 73 : Exemple de *shadow-facts* créés à partir du fichier xml décrit Figure 69

3.2.2.2 Gérer les interventions du partenaire et de l’apprenant

Les possibilités d’intervention du partenaire sont multiples. Dans ce qui suit, nous présentons le code de quelques règles, le contexte dans lequel elles sont déclenchées, ainsi que leurs effets sur l’interface graphique.

- **Afficher une bulle textuelle et une image du partenaire** : lorsqu’un énoncé est créé, une bulle contenant le texte est produite. Il est également possible de modifier l’image du partenaire, cette image pouvant varier selon des humeurs prêtées au partenaire en fonction des énoncés de l’apprenant auquel il réagit (dans le cadre de l’activité de production). Cette possibilité n’est pas exploitée à l’heure actuelle.

```
(defrule INTERFACE::ecrire
      (declare (salience 80))
      ?p <- (MAIN::a-ecrire ?ch)
      =>
      (list-focus-stack)
      (retract ?p)
      (afficher (format nil "%s%n" ?ch) "partenaire.gif"))
```

Figure 74 : Règle d’affichage des énoncés et de l’apparence du partenaire

- **Manipuler un objet du monde** : ouvrir / déplacer / changer un objet. Pour l’activité de production, ces manipulations proviennent des méthodes opérationnelles des tâches de l’arbre des tâches. Elles sont déclenchées après la reconnaissance d’un énoncé de l’apprenant comme étant une instruction exécutable. S’appuyant sur les *shadow facts*, la fonction call permet d’exercer sur le ou les objets identifiés une méthode décrite dans la partie Java. Ces manipulations recouvrent l’ouverture, le déplacement et la transformation (c’est à dire le changement d’état) d’objets. Les figures suivantes

présentent les trois cas de manipulation, donnant d'abord la règle de manipulation puis ses conséquences graphiques. Ces trois cas correspondent à une séquence interactive qui commencerait par l'ouverture du frigo, la prise des oeufs et l'ouverture de la boîte provoquant sa transformation. Les ouvertures se font grâce à la règle suivante (Figure 75).

```
(defrule  INTERFACE::ouvrir
    (declare (salience 10))
    ?p <- (declenche "ouvrir" ?identobjet)
    ?r <- (objetouvrable (OBJECT ?objet) (ident ?identobjet)
          (position ?pos)(image ?im) (ouvert FALSE))
    =>
    (retract ?p ?r)
    (call ?objet ouvrir_jess)
    (assert (objetouvrable (OBJECT ?objet) (ident ?identobjet)
            (position ?pos) (image ?im)(ouvert TRUE))))
```

Figure 75 : Exemple de règle de manipulation, ouverture d'un rangement

Nous décrivons cette règle. La première ligne définit l'intitulé de la règle (INTERFACE:ouvrir), elle appartient donc au module INTERFACE et se nomme ouvrir. La seconde ligne définit la « salience », c'est à dire la priorité de la règle. Ce concept permet de donner des ordres de priorité dans les exécutions de plusieurs règles d'un même module. Ici la priorité de la règle est de 10, priorité que l'on peut comparer avec celle de la règle d'affichage des énoncés (80) présentée Figure 75. Les affichages des énoncés ont lieu avant les manipulations à l'interface (le partenaire dit ce qu'il fait avant de l'effectuer). Les deux lignes suivantes sont les prémisses de la règle. Deux faits doivent exister pour que la règle soit exécutée : le premier stipule le déclenchement de l'ouverture de l'objet qui a comme identifiant ?identobjet ; le second représente l'objet d'identifiant ?identobjet dont l'attribut ouvert est positionné à faux. Le premier fait est affecté à la variable ?p, le second à ?r. Les lignes suivantes présentent les conséquences de l'exécution de la règle. La première ligne supprime les faits, prémisses de la règle, de la base de faits. La seconde ligne appelle la méthode ouvrir_jess de l'objet ?objet, ceci déclenche l'ouverture de l'objet sur l'interface graphique. La dernière ligne ajoute un nouveau fait dans la base de faits. Ce fait est identique à celui qui est la seconde prémisse de la règle mais l'attribut ouvert a pour valeur vrai. Dans le cas de l'énoncé, « ouvrir le frigo », nous obtenons la configuration de l'interface suivante lors de cette ouverture (Figure 76).

Figure 76 : Interface après ouverture du frigo

L’interaction se poursuit, les oeufs sont sortis du frigo et déplacés vers le plan de travail (Figure 77).

```
(defrule INTERFACE::dep-sortir
     ?p <- (declenche "sortir" ?identobjet1 ?x ?y)
     ?q <- (INTERFACE::dans ?identobjet1 ?identobjet2)
     ?l <- (position ?identobjet1 ?x1 ?y1)
     ?objet <- (objetdep (OBJECT ?objet1) (ident ?identobjet1))
     =>
     (retract ?p ?q ?l)
     (bind ?dest (new java.awt.Point ?x ?y))
     (assert (INTERFACE::deplacement ?identobjet1))
     (assert (position ?identobjet1 ?x ?y))
     (call ?objet1 bouge ?dest))
```

Figure 77 : Exemple de règle de manipulation, déplacement d’un objet

L’interface évolue ainsi (Figure 78).

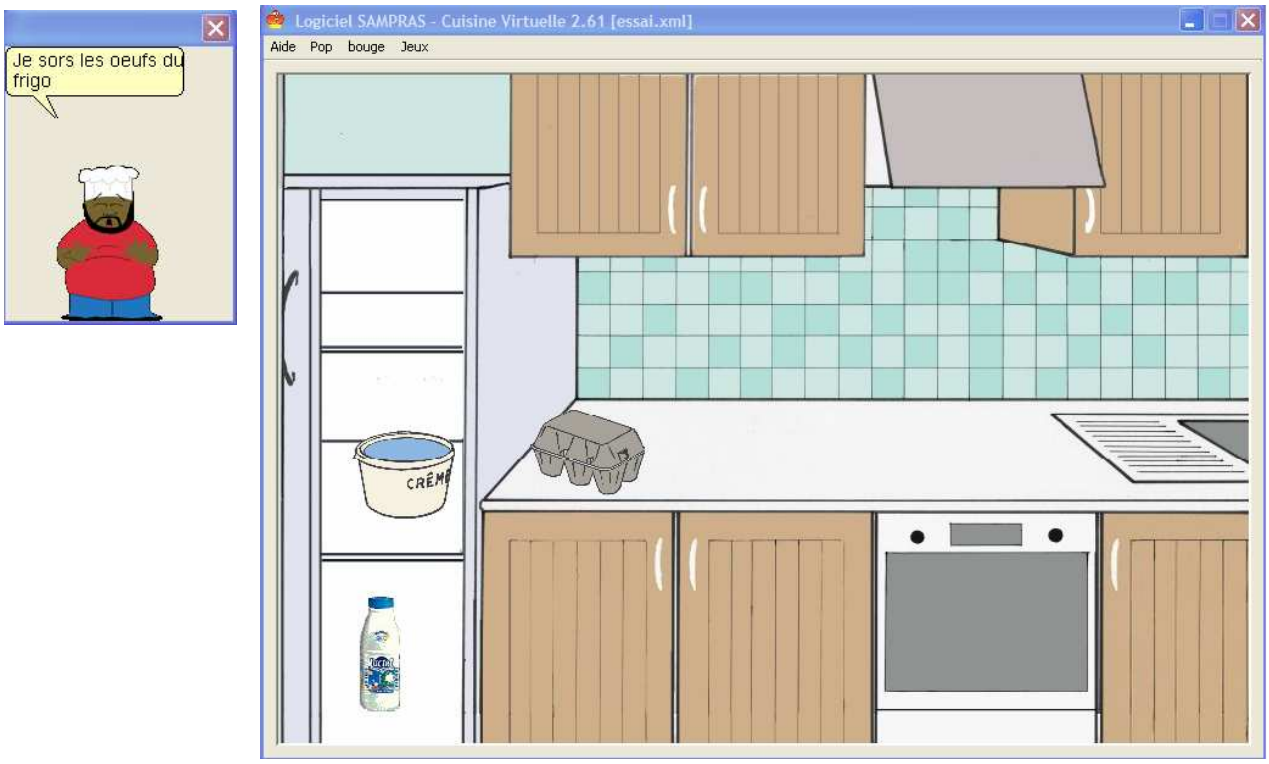


Figure 78 : Interface homme-machine après prise des œufs

Le pas suivant est d'ouvrir la boîte, provoquant le changement de l'apparence de la boîte à œufs (Figure 79).

```
(defrule INTERFACE::changement
      ?p <- (declenche "changer" ?identobjet ?etatf)
      ?f <- (INTERFACE::BON-ETAT)
      ?g <- (INTERFACE::ACTION-CHGT)
      (objetmod (OBJECT ?objet) (ident ?identobjet) (etat ?etatc&~?etatf))
      =>
      (retract ?p)
      (set ?objet etat ?etatf))
```

Figure 79 : Exemple de règle de manipulation, déplacement d'un objet

Nous obtenons alors l'interface suivante (Figure 80).

Figure 80 : Interface homme-machine après ouverture de la boîte à oeufs

L’autre fonctionnalité du module INTERFACE est de recueillir les interventions de l’apprenant. L’apprenant écrit un énoncé dans son champ de texte. Cet énoncé est transmis au module INTERFACE sous la forme d’un fait. Le module INTERFACE dirige alors l’énoncé vers le module d’analyse.

3.3 Analyse

L’analyse a pour but de faire émerger du sens à partir des énoncés de l’apprenant. L’analyse repose sur un lexique décrivant, sous la forme d’une grammaire sémantique, les différentes formes possibles des ordres attendus de la part de l’apprenant lors de la réalisation de l’activité de production.

3.3.1 Les connaissances lexicales

Le lexique contient les termes et les structures que l’utilisateur va être amené à employer au cours de l’activité de production. Y sont donc présents les objets du monde, les actions possibles avec ceux-ci et les questions que l’apprenant peut poser à son partenaire.

Le lexique est composé de **lexies** (unités lexicales) réunies par une même **catégorie lexicale**. Treize catégories lexicales ont été définies : recette, ingrédient, récipient, électroménager, contenant, ustensile, rangement, localisateur, rangement-plat, attribut, action, ordre, question. Ces catégories ont été identifiées à partir de l'étude des différentes recettes de gâteau et à partir des différents types d'interaction.

Une lexie est dénotée par un vocable et est associée à des motifs syntaxiques. Un **vocable** est un mot ou une expression caractérisable par une des catégories lexicales. Un **motif syntaxique** est une liste ordonnée de vocables et de catégories lexicales décrivant la structure attendue d'un énoncé reprenant le vocable. Par exemple, voici un fragment de la définition de la lexie "prendre" qui appartient à la catégorie lexicale "action". Le vocable de la lexie est la chaîne de caractère "prendre". Les motifs syntaxiques associés reprennent le vocable et lui associe des expressions. Celles-ci sont de catégories lexicales attendues à la suite de ce vocable. Dans le cas de "prendre", il peut être suivi d'une expression dont la catégorie est ingrédient, récipient, ustensile, *etc*. Le lexique est décrit sous la forme d'un fichier XML dont la DTD est présentée ci-dessous (Figure 81).

```
<!ELEMENT lexique (lexies*)>            Le lexique est organisé en lexies,
<!ELEMENT lexies (lexie+)>              soit un ensemble d'unités lexicales
<!ATTLIST lexies cat ID #REQUIRED>      de même catégorie lexicale.
<!ELEMENT lexie (syntaxe*)>         Une lexie possède des motifs syntaxiques
<!ATTLIST lexie voc CDATA #REQUIRED>    et est dénotée par un vocable.
<!ELEMENT syntaxe (#PCDATA|expr)*>  Un motif syntaxique est composé de mots
<!ELEMENT expr EMPTY>                   et d'expressions
<!ATTLIST expr cat IDREF #REQUIRED>     identifiées par une catégorie
                                        lexicale.
```

Figure 81 : La DTD servant de modèle de lexique

Partant de cette DTD, nous décrivons notre lexique dans un fichier XML. Un fragment de ce fichier est présenté ci-dessous (Figure 82). Nous pouvons voir que la lexie « prendre » dépend de la catégorie lexicale « action » et possède six motifs syntaxiques.

```
<?xml version="1.0"?>
<!DOCTYPE lexique SYSTEM "lexique.dtd">
<lexique>
   <lexies cat="ingrédient">
     <lexie voc="les ingrédients"/>
     <lexie voc="le sucre"/>
     <lexie voc="la farine"/>
     <lexie voc="le chocolat"/>
     <lexie voc="les oeufs"/>
     <lexie voc="le lait"/>
  </lexies>
  <lexies cat="rangement">
     <lexie voc="le placard">
      <syntaxe>le placard</syntaxe>
      <syntaxe>le placard<expr cat="attribut"/></syntaxe>
  </lexie>
    <lexie voc="le frigo"/>
  </lexies>
  <lexies cat="action">
    <lexie voc="prendre">
      <syntaxe>prendre<expr cat="ingrédient"/></syntaxe>
      <syntaxe>prendre<expr cat="récipient"/></syntaxe>
      <syntaxe>prendre<expr cat="ustensile"/></syntaxe>
      <syntaxe>prendre<expr cat="ingrédient"/><expr cat="localisateur"/>
               <expr cat="rangement"/></syntaxe>
      <syntaxe>prendre<expr cat="récipient"/><expr cat="localisateur"/>
               <expr cat="rangement"/></syntaxe>
      <syntaxe>prendre<expr cat="ustensile"/><expr cat="localisateur"/>
               <expr cat="rangement"/></syntaxe>
    </lexie>
  </lexies>
  <lexies cat="ordre">
     <lexie voc="tu dois">
      <syntaxe>tu dois<expr cat="action"/></syntaxe>
    </lexie>
  </lexies>
</lexique>
```

Figure 82 : Un extrait du fichier XML représentant le lexique

Ce fichier XML correspond à la grammaire sémantique définie ci-dessous (Figure 83). Les raisons de cette modélisation, ainsi que l'usage qui en est fait dans le processus d'analyse sont explicités dans les paragraphes 2.3.3 et 2.3.4.

“les ingrédients” → ⟨ingrédient⟩

“le sucre” → ⟨ingrédient⟩

“la farine” → ⟨ingrédient⟩

“le chocolat” → ⟨ingrédient⟩

“les œufs” → ⟨ingrédient⟩

“le lait” → ⟨ingrédient⟩

“le placard” → ⟨rangement⟩

“le placard ⟨attribut⟩” → ⟨rangement⟩

“le frigo” → ⟨rangement⟩

“prendre ⟨ingrédient⟩” → ⟨action⟩

“prendre ⟨récipient⟩” → ⟨action⟩

“prendre ⟨ustensile⟩” → ⟨action⟩

“prendre ⟨ingrédient⟩ ⟨localisateur⟩ ⟨rangement⟩” → ⟨action⟩

“prendre ⟨récipient⟩ ⟨localisateur⟩ ⟨rangement⟩” → ⟨action⟩

“prendre ⟨ustensile⟩ ⟨localisateur⟩ ⟨rangement⟩” → ⟨action⟩

“tu dois ⟨action⟩” → ⟨ordre⟩

Figure 83 : Grammaire sémantique équivalente

Le fichier XML est ensuite transformé via un script XSLT en une base de fait Jess (lexique.clp) qui va contenir les connaissances lexicales sur lesquelles l'analyse s'appuie.

3.3.2 Le module analyse

Le module analyse construit une interprétation de l'énoncé. Cette interprétation, s'il est possible d'en réaliser une, a pour but d'être comparée avec les instructions définies pour la tâche applicative. Nous avons présenté, dans le paragraphe 2.3.4 du chapitre 2, le déroulement de l'analyse d'un énoncé. Nous reprenons ici ce déroulement et explicitons les réalisations

informatiques qui les concrétisent. Le code complet de l'analyse est disponible dans l'annexe B.

3.3.2.1 Exploration de l'énoncé

Cette exploration repose sur la granularisation des énoncés. Les énoncés lorsqu'ils arrivent au module analyse se présentent sous la forme d'une simple chaîne de caractères. Celle ci est transformée en une liste de mots. Dans le cas où l'énoncé est "prendre les oeufs", nous obtenons le fait phrase suivant :

```
(ANALYSEUR::phrase (listemots prendre les oeufs))
```

Cette liste de mots est ensuite granularisée, c'est à dire que l'on essaye de catégoriser les éléments de l'énoncé par rapport aux éléments du lexique. Une granule est composée de différents champs : un texte (un mot ou une suite de mots), la catégorie lexicale dont le texte relève, la position de ce texte dans l'énoncé, la couverture (le nombre de mots) que ce texte recouvre, le nombre de mots sur lesquels sont posées des hypothèses et les éventuelles sous-granules qui le compose. Ces granules sont des faits possédant comme attribut chacun des champs défini (le champ couverture a été renommé taille).

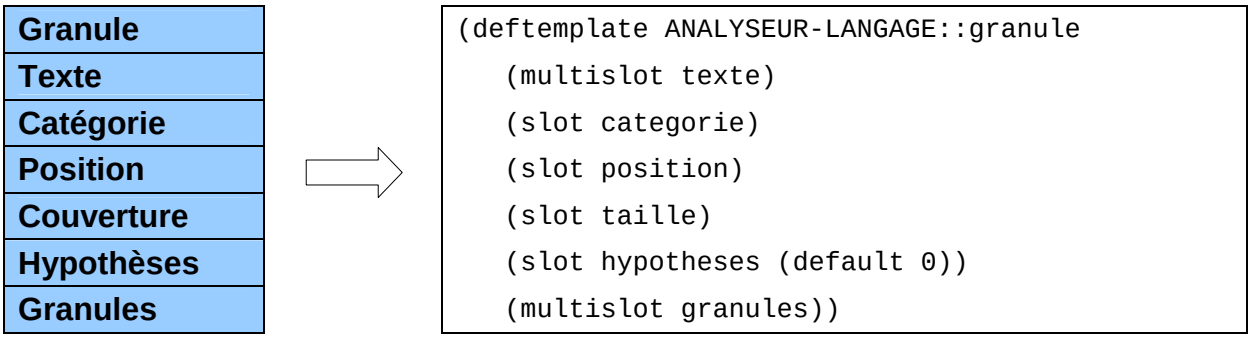


Figure 84 : Modèle d'une granule dans Jess (template)

En reprenant l'énoncé "prendre les oeufs", deux granules sont construites (les règles de construction sont comprises en annexe). L'énoncé est divisé en mots, puis ces mots, mis en relation avec les lexies du lexique, donnent les granules suivantes (Figure 85, Figure 86).

Granule	
Texte	Prendre
Catégorie	action
Position	0
Couverture	1
Hypothèses	0
Granules	

```
(ANALYSEUR-LANGAGE::granule
   (texte prendre)
   (categorie [action])
   (position 0)
   (taille 1)
   (hypotheses 0)
   (granules ))
```

Figure 85 : Granule pour le texte « prendre »

Granule	
Texte	les oeufs
Catégorie	ingrédient
Position	1
Couverture	2
Hypothèses	0
Granules	

```
(ANALYSEUR-LANGAGE::granule
   (texte les oeufs)
   (categorie [ingredient])
   (position 1)
   (taille 2)
   (hypotheses 0)
   (granules ))
```

Figure 86 : Granule pour le texte « les œufs »

Si des mots ne sont pas identifiés et ne donnent donc pas lieu à la création de granules, des hypothèses sur la nature de ces mots peuvent être posées. Ces mises d'hypothèses se déroulent en mettant en relation les motifs syntaxiques du lexique et les positions des éléments déjà reconnus. Si nous prenons l'énoncé "prendre les eggs dans le frigo", nous obtenons trois granules hypothétiques.

```
      ==> f-762 (ANALYSEUR-LANGAGE::granule (texte les eggs) (granules )
(categorie [ustensile]) (hypotheses 2) (position 1) (taille 2))
      ==> f-763 (ANALYSEUR-LANGAGE::granule (texte les eggs) (granules )
(categorie [recipient]) (hypotheses 2) (position 1) (taille 2))
      ==> f-764 (ANALYSEUR-LANGAGE::granule (texte les eggs) (granules )
(categorie [ingredient]) (hypotheses 2) (position 1) (taille 2))
```

L'assemblage des granules consiste à regrouper les granules contiguës en une seule. Les granules issues de la granularisation sont rassemblées une à une, créant des agglutinations de granules qui vont elle-même s'agglutiner entre elles. Les agglutinations qui correspondent à des motifs syntaxiques sont les premiers résultats de l'analyse. Dans le cas de l'énoncé

“prendre les oeufs”, trois résultats sont identifiés : les deux granules initiales et celle formée par leur assemblage.

```
==> f-704 (ANALYSEUR-LANGAGE::granule (texte prendre [ingredient]) (granules <Fact-696> <Fact-699>) (categorie [action]) (hypotheses 0) (position 0) (taille 3) (lex FALSE))
```

Il est à noter qu’il existe un grand nombre de granules créées lors des assemblages, puisque ces assemblages se réalisent seulement en fonction des positions relatives des granules, sans faire entrer en ligne de compte une quelconque sémantique.

Ces résultats sont ensuite comparés selon des heuristiques cherchant à déterminer quel résultat représente l’énoncé de la façon la plus performante, par comparaison des couvertures et des marqueurs d’hypothèse. Dans le cas de l’énoncé “prendre les oeufs”, le résultat est la granule dernièrement créée. Elle couvre tout l’énoncé avec un marqueur d’hypothèse à 0, contrairement aux deux autres qui, avec le même marqueur d’hypothèse, couvrent des portions plus réduites de l’énoncé. Les résultats sont des faits ayant comme attributs les adresses des faits granules le composant, sa position, la couverture du résultat, un marqueur d’hypothèse. Le résultat obtenu pour “prendre les oeufs” est le suivant.

```
==> f-710 (ANALYSEUR-LANGAGE::result (granule <Fact-704>) (granules <Fact-704> <Fact-699>) (position 0) (couverture 3) (hypotheses 0))
```

La figure suivante représente la structure produite à partir de l’énoncé « prendre les œufs ». Les étiquettes des nœuds font référence aux faits détaillés ci-dessus.

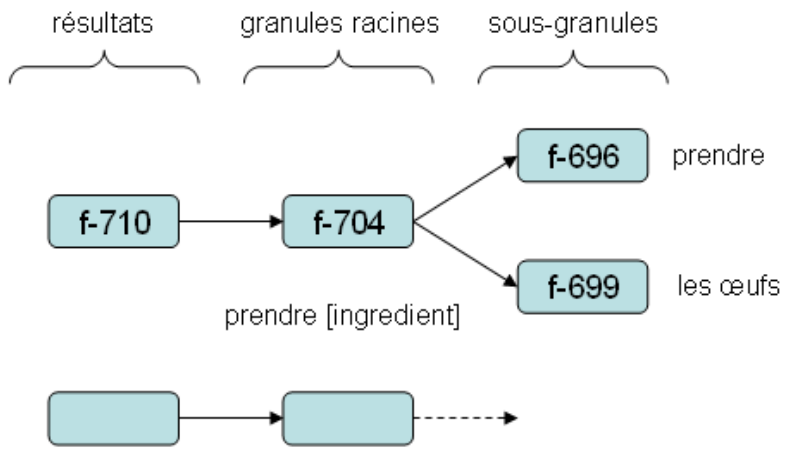


Figure 87 : Résultat produit par l’analyse

Ajoutons que l'analyse (non-déterministe) est susceptible de produire plusieurs résultats concurrents. Des heuristiques viennent ensuite choisir l'analyse la plus couvrante et la moins hypothétique.

3.3.2.2 Vérifications et manipulations

Les vérifications lexicales et syntaxiques (cf. § 2.3.4) permettent de déterminer si tous les mots de l'énoncé sont connus du lexique et si un motif syntaxique peut être appliqué à cet énoncé. Du résultat des vérifications va dépendre l'usage de composants annexes qui détermineront les analyses à envoyer au module génération (cf. § 2.3.4.1). Dans le cas n°1 d'une analyse ayant une **bonne couverture lexicale** et une **bonne couverture syntaxique**, le résultat est comparé avec les éléments de l'arbre des tâches. Dans le cas n°2 d'une analyse ayant une **bonne couverture lexicale** et une **mauvaise couverture syntaxique**, un composant va tenter de réorganiser les granules afin de faire correspondre l'énoncé à un motif syntaxique existant. Prenons l'énoncé "les oeufs prendre", deux granules sont identifiées :

```
==> f-720 (ANALYSEUR-LANGAGE::granule (texte les oeufs) (granules ) (categorie [ingredient]) (hypotheses 0) (position 0) (taille 2))

==> f-721 (ANALYSEUR-LANGAGE::granule (texte prendre) (granules ) (categorie [action]) (hypotheses 0) (position 2) (taille 1))
```

A partir de ces deux granules, une liste de nouveaux résultats possibles est créée. Cette liste comprend les motifs syntaxiques constructibles avec ces deux granules. Elle est ensuite fournie au module génération qui construit une réponse adéquate.

Dans le cas n°3 d'une analyse ayant une **mauvaise couverture lexicale** et une **bonne couverture syntaxique** (grâce aux hypothèses, un motif syntaxique est applicable), le résultat est envoyé au module génération pour confrontation avec l'état de la tâche. Dans le cas n°4 d'une analyse ayant une **mauvaise couverture lexicale** et une **mauvaise couverture syntaxique**, la suite des opérations est identique au cas n°2.

3.4 La tâche applicative

Les connaissances sur la tâche (cf. Figure 67) permettent la gestion de l'avancement de la tâche et la création d'une partie de la réaction du partenaire à un énoncé de l'apprenant.

3.4.1 Représenter la tâche

La tâche applicative, décrite dans le deuxième chapitre (cf. § 2.1.2.3), repose sur la mise en œuvre du paradigme « tâche/méthode ». Nous évoquons dans ce paragraphe son implémentation sous la forme de faits et de règles Jess. La figure suivante (cf. Figure 88) représente les différents concepts mis en œuvre : tâche, méthode et opération. Ces concepts étant réifiés sous la forme de templates Jess (cf. Figure 89).

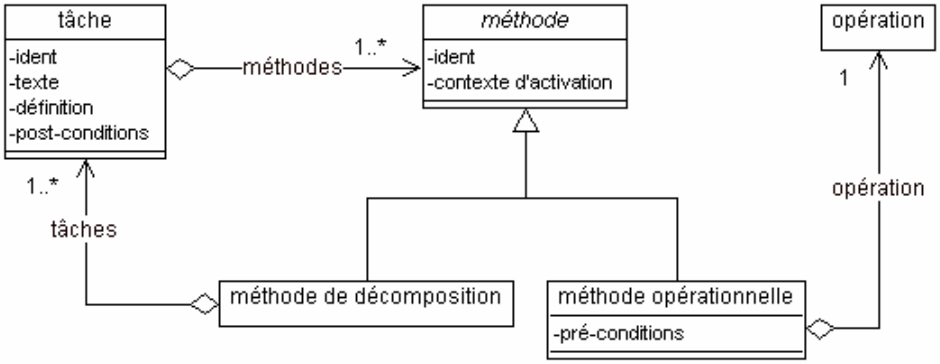


Figure 88 : Diagramme UML du modèle « Tâche/Méthode/Opération »

```
(deftemplate MAIN::tache
      (slot      ident)
      (slot      texte)
      (slot      definition)
      (multislot post-conditions)
      (multislot methodes))

(deftemplate MAIN::methode-de-decomposition
      (slot      ident)
      (multislot contexte-d-activation)
      (multislot taches))

(deftemplate MAIN::methode-operationnelle
      (slot      ident)
      (multislot contexte-d-activation)
      (multislot preconditions)
      (slot      operation))
```

Figure 89 : Modèles des tâches et des méthodes dans Jess (templates)

Pour comprendre le fonctionnement des tâches et des méthodes, nous en allons en étudier quelques unes. Le premier exemple concerne la tâche principale : la réalisation de la recette (Figure 90). Son identificateur est **t-faire-flan**. Le texte « faire un flan » est une chaîne de caractères désignant le texte correspondant à cette tâche. La définition ([action] (faire [recette])) ([recette] (un flan)) est un motif permettant de faire le lien entre la tâche et un résultat d'analyse. Les post-conditions (fait flan-cuit) et (fait flan-dispo) sont les résultats de l'exécution de cette tâche. L'unique méthode associée est **m-faire-flan** : une méthode de décomposition, sans contexte d'activation, qui énumère les tâches à accomplir pour que la tâche t-faire-flan soit effectivement réalisée : t-prendre-ingredients, t-prendre-ustensiles, t-prechauffer-four, t-preparer-flan, t-faire-cuire.

```
(tache
     (ident              t-faire-flan)
     (texte              "faire un flan")
     (definition         "([action] (faire [recette]))
                         ([recette] (un flan))")
     (post-conditions    "(fait flan-dispo)"
                         "(fait flan-cuit)")
     (methodes           m-faire-flan))

(methode-de-decomposition
     (ident              m-faire-flan)
     (taches             t-prendre-ingredients
                         t-prendre-ustensiles
                         t-prechauffer-four
                         t-preparer-flan
                         t-faire-cuire)))
```

Figure 90 : Tâche t-faire-flan et son unique méthode

Un autre exemple est celui d’une tâche qui possède des méthodes opérationnelles. La tâche **t-prendre-lait-dans-frigo** (Figure 91) possède la méthode opérationnelle **m-prendre-lait-dans-frigo** qui a pour préconditions (not (fait lait-dispo)), (fait frigo-ouvert), (fait frigo-contient-lait). Ces trois préconditions doivent être vérifiées pour que la méthode soit exécutable. A cette méthode est associée la fonction Jess **f-prendre-lait-dans-frigo**. Cette fonction illustre comment les ordres de manipulation des objets du monde sont passés au module interface. Le premier fait déclenche à l’interface la sortie du lait du frigo et son déplacement vers une nouvelle position dont sont données les coordonnées. Le second donne une phrase à écrire dans la bulle du partenaire: « Je sors le lait du frigo ». Le troisième signale que désormais le lait est disponible. Le fait que le lait est dans le frigo est supprimé.

```
(tache
     (ident              t-prendre-lait-dans-frigo)
     (texte              "prendre le lait")
     (definition         "([action] (prendre [ingredient]))
                         ([ingredient] (le lait))")
     (post-conditions    "(fait lait-dispo)"
                         "(not (fait frigo-contient-lait))")
     (methodes           m-prendre-lait-dans-frigo))

(methode-operationnelle
     (ident              m-prendre-lait-dans-frigo)
     (preconditions      "(not (fait lait-dispo))"
                         "(fait frigo-ouvert)"
                         "(fait frigo-contient-lait)")
     (fonction           f-prendre-lait-dans-frigo)))

(deffunction f-prendre-lait-dans-frigo ()
     (assert             (INTERFACE::declenche "sortir" "le lait" 480 230))
     (assert             (a-ecrire "Je sors le lait du frigo"))
     (assert             (fait lait-dispo))
     (retract-string     "(fait frigo-contient-lait)"))
```

Figure 91 : Tâche prendre-le-lait

Il est à noter que l'écriture de la tâche applicative est longue et fastidieuse : la description minutieuse des sous-tâches, la création de fonctions agissant sur l'interface obligent à faire constamment des allers-retours entre gestion de la base de faits, disposition de l'interface et écriture des tâches. Ceci ne rend pas aisé la transformation ou réécriture de la tâche applicative par un non-spécialiste de l'informatique.

3.4.2 Gérer la tâche

La gestion de la tâche, donc de l'état de son avancement, passe par la connaissance des actions réalisables à chaque moment de l'activité. Deux listes sont ainsi tenues à jour : celle des méthodes activables et celles des méthodes opérationnelles exécutables. Les méthodes activables sont celles dont le contexte d'activation est vérifié. Dans l'exemple suivant, la tâche **t-prendre-saladier** est réalisable au travers de deux méthodes (Figure 92) :

celle correspondant au placard ouvert et celle correspondant au placard fermé. Le contexte d'activation permet de les différencier.

```
(tache
    (ident              t-prendre-saladier)
    (texte              "prendre le saladier")
    (definition         "([action] (prendre [recipient]))
                         ([recipient] (le saladier))")
    (post-conditions    "(fait saladier-dispo)")
    (methodes           m-prendre-saladier-dans-placard5-ouvert
                        m-prendre-saladier-dans-placard5-ferme))

(methode-de-decomposition
    (ident              m-prendre-saladier-dans-placard5-ouvert)
    (contexte-d-activation     "(not (fait saladier-dispo))"
                               "(fait placard5-ouvert)"
                               "(fait placard5-contient-saladier)")
    (taches             t-prendre-saladier-dans-placard5))

(methode-de-decomposition
    (ident              m-prendre-saladier-dans-placard5-ferme)
    (contexte-d-activation     "(not (fait saladier-dispo))"
                               "(fait placard5-ferme)"
                               "(fait placard5-contient-saladier)")
    (taches             t-ouvrir-placard5))
```

Figure 92 : Tâche « prendre le saladier »

Les méthodes opérationnelles exécutables le sont lorsque toutes leurs préconditions sont vérifiées. Si nous reprenons l'exemple de la Figure 91, lorsque le lait n'est pas disponible (il est sorti dans le frigo), que le frigo est ouvert et contient le lait alors la méthode **m-prendre-lait-dans-frigo** devient exécutable.

Nous ne décrivons pas plus loin la gestion de la tâche, elle est explicitée dans [Delorme, 2002]. Fabien Delorme a imaginé et opérationnalisé la gestion de la tâche. Il est juste important de retenir pour la suite que l'exécution des méthodes opérationnelles modifie la liste des méthodes activables et opérationnelles. Cette liste sert au module d'interaction pour déterminer quelles tâches sont à exécuter suivant les éléments délivrés par les modules analyse et génération.

Le paragraphe suivant explicite le mécanisme de comparaison entre les résultats de l'analyse et les éléments de la tâche applicative.

3.5 Module Génération

A la suite de l'analyse, les résultats produits en termes de couverture (lexicale et syntaxique) permettent de guider l'interprétation qui peut être fait de l'énoncé de l'apprenant. Certains résultats de l'analyse sont inexploitables et donc l'étape de mise en contexte n'a pas lieu pour ceux-ci. Le module génération est divisé en sous-modules : comparaison-tâche, question, mémoire, incidence et incidence-mémoire.

Le **module comparaison-tâche**, cherche à découvrir la tâche de l'arbre des tâches à laquelle l'apprenant fait (éventuellement) référence dans son énoncé. Nous distinguons deux cas principaux : les cas où aucun élément provenant des interactions précédentes ne rentre en jeu et les cas où des éléments provenant de ces interactions sont pris en compte. Le **module question** est activé lorsqu'une question est détectée. Le **module incidence**, qui est utilisé si un énoncé n'est pas complètement interprété, entraîne une demande de reformulation auprès de l'apprenant. Le **module mémoire** cherche à reconstruire des interprétations en se fondant sur plusieurs échanges. Le **module incidence-mémoire** prend le relais du module mémoire lorsque le résultat produit par ce dernier ne correspond pas à une tâche attendue. Il se fonde alors sur des éléments des interactions précédentes pour construire une question.

La recherche d'une tâche correspondante est la première étape de la contextualisation de l'intervention de l'apprenant, le résultat de cette recherche étant ensuite transmis au module interaction.

3.5.1 Recherche de la tâche sans influence des interactions précédentes

Ce paragraphe explicite comment le résultat d'une analyse permet d'identifier et de sélectionner une tâche décrite. La première étape est la création d'un fait qui va correspondre aux champs définition des tâches. Les comparaisons peuvent ensuite avoir lieu, qui peuvent aboutir à l'identification d'une tâche résultat. La figure suivante illustre le processus à partir de l'énoncé « ouvrir le frigo » (cf. Figure 93). Le code complet du module de comparaison

avec la tâche est présenté dans l'annexe B. Les quatre règles qui supportent ce processus sont explicitées dans ce qui suit.

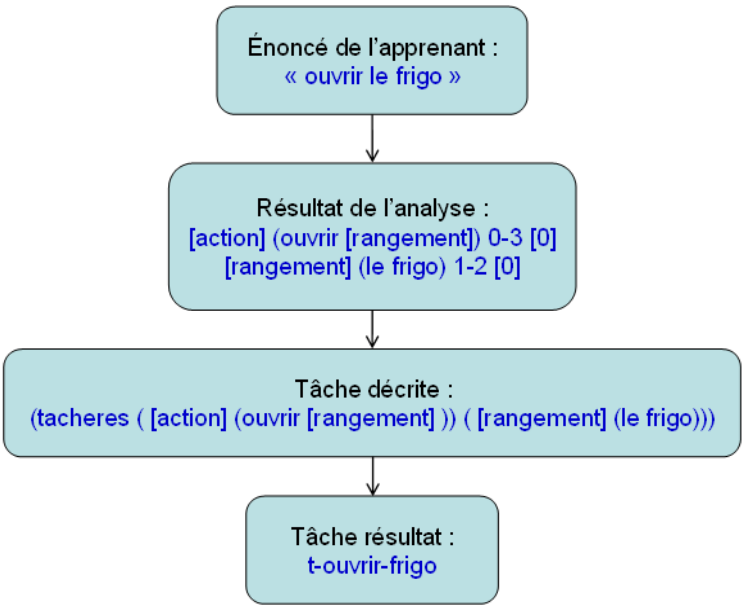


Figure 93 : Identification d'une tâche à partir d'un énoncé

Une première règle permet la détection des questions (Figure 94). Les questions implémentées concernent la place des éléments dans la cuisine. La suite du traitement a lieu dans le **module question**, dédié à la construction de réponses aux questions de l'apprenant.

```
(defrule COMPARAISONTACHE::question
      (declare (salience 11))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (test (COMPARAISONTACHE::member-string ?des "[question]"))
      =>
      (retract ?r ?p)
      (assert (analyse (resultat question-ou ?des)))
      (focus QUESTION))
```

Figure 94 : Règle permettant de détecter une question

Une seconde règle identifie si la tâche résultat est une tâche de l'arbre des tâches (Figure 95). Elle délivre alors le résultat final de l'analyse (la tâche identifiée) et demande le nettoyage des éléments qui ont participé à l'analyse de l'énoncé (les diverses granules et

agglutinations) et les éléments de la mémoire s'il y en a. L'analyse finale sert alors au module interaction.

```
(defrule COMPARAISONTACHE::correspondancearbretache
      (declare (salience 10))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (tache (ident ?ident) (definition ?def))
      (test (COMPARAISONTACHE::member-string ?des ?def))
      =>
      (retract ?p ?r)
      (assert (analyse (resultat tache ?ident)))
      (focus NETTOYAGE)
      (focus NETTOYAGE-MEMOIRE))
```

Figure 95 : Règle permettant de détecter une tâche

Nous décrivons cette règle. Les prémisses de cette règle sont l'existence de trois faits et la vérification d'un test. La première prémisse est un fait qui indique l'existence d'une comparaison entre le résultat d'une analyse et l'arbre des tâches. La seconde prémisse est un fait représentant le résultat de l'analyse avec un attribut qui représente le résultat sous la même forme que celle du champ définition des tâches de l'arbre des tâches. La troisième prémisse est une tâche de l'arbre des tâches avec sa définition. La quatrième est un test qui compare le champ définition des tâches de l'arbre des tâches (?def) et l'attribut représentant le résultat (?des). Ce test renvoie vrai si les éléments de ?def sont des éléments de ?des. Il y a alors concordance entre le résultat de l'analyse et une tâche de l'arbre des tâches. Les conséquences de l'exécution de cette règle sont la suppression des deux premiers faits prémisses de la règle (première ligne), l'assertion du résultat final, c'est-à-dire l'identité de la tâche détectée. Les deux dernières commandent l'activation de deux modules annexes qui nettoient les éléments créés au cours de l'analyse et vident la mémoire. Dans le cas de l'énoncé « ouvrir le frigo », La tâche identifiée est la suivante (Figure 96) :

```
(tache
     (ident              t-ouvrir-frigo)
     (texte              "ouvrir le frigo")
     (definition         "([action] (ouvrir [rangement]))
                          ([rangement] (le frigo))")
     (post-conditions    "(fait frigo-ouvert)"
                         "(not (fait frigo-ferme))")
     (methodes           m-ouvrir-frigo))

(methode-operationnelle
     (ident              m-ouvrir-frigo)
     (preconditions      "(fait frigo-ferme)")
     (fonction           f-ouvrir-frigo)))
```

Figure 96 : Tâche identifiée à partir de la tâche résultat

La troisième règle (Figure 97) identifie les cas où (condition n°1) il n’y a pas de correspondance entre la tâche résultat et des tâches de l’arbre des tâches, et où (condition n°2) il n’y a pas d’éléments de la tâche résultat dans les définitions de ces mêmes tâches (sachant qu’il est possible que certains éléments du lexique ne correspondent pas à une tâche précise). Le résultat est une « action hors-tâche ». Ce cas concerne par exemple, l’action « ranger » qui existe dans le lexique mais pas dans la tâche.

```
(defrule COMPARAISONTACHE::noncorrespondancearbretache-1
      (declare (salience 5))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      (test (not (COMPARAISONTACHE::member-string ?des ?def))) ; cond n°1
      (test (not (COMPARAISONTACHE::member-string ?def ?des))) ; cond n°2
      =>
      (retract ?p ?r)
      (assert (analyse (resultat pastache)))
      (focus NETTOYAGE)
      (focus NETTOYAGE-MEMOIRE))
```

Figure 97 : Règle de détection d’énoncés hors-tâche

La quatrième règle identifie les cas où (condition n°1) il n’y a pas de correspondance entre la tâche résultat et les tâches de l’arbre des tâches et où (condition n°2) il existe des

éléments communs à la tâche résultat et à certaines tâches de l'arbre (Figure 98). Le résultat de l'analyse est envoyé au module incidence pour une analyse plus complète.

```
(defrule COMPARAISONTACHE::noncorrespondancearbretache-2-ordreincomplet
      (declare (salience 7))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      (test (not (COMPARAISONTACHE::member-string ?des ?def))) ; cond n°1
      (test (COMPARAISONTACHE::member-string ?def ?des))       ; cond n°2
      =>
      (retract ?p ?r)
      (assert (ANALYSEUR-LANGAGE::analysepblincomplet
                 (resultat $?granules)))
      (focus INCIDENCE))
```

Figure 98 : Règle de détection d'énoncés représentant de façon incomplète une tâche

Les résultats qui ne correspondent pas à la description d'une tâche sont transmis au **module incidence**. Celui-ci récupère les résultats et en extrait les granules. Ces granules sont utilisées d'une part pour la création des réactions du partenaire (granule-seule) et d'autre part sous la forme de faits mémoire (granule-mem) qui pourront être réutilisés pour des analyses futures comme **indices mémoire**. La figure suivante représente les granules extrait du résultat de l'analyse de « prendre les oufs dans le frigo » : deux « granules seules » qui seront utilisées comme indices d'incompréhension, et deux « granules mémoires » qui peuvent être utiles dans des identifications de tâches futures (Figure 99).

```
==> f-706 (INCIDENCE::granule-seule (categorie [action]) (texte prendre)
(granules ))
==>  f-707  (MEMOIRE::granule-mem  (granules  )  (texte  prendre)  (categorie
[action]))
==> f-1026 (INCIDENCE::granule-seule (categorie [rangement]) (texte le
frigo) (granules ))
 ==> f-1027 (MEMOIRE::granule-mem (granules ) (texte le frigo) (categorie
[rangement]))
```

Figure 99 : Granules créées lors d'une incidence

Les indices permettent de créer une partie de la réaction du partenaire. En effet, en fonction de ces granules, nous sommes à même de poser des questions ou engager des reformulations. Si nous reprenons l’exemple où l’on identifie une action (exemple : prendre) et un rangement (exemple: le frigo) qui sont donc conservées sous la forme de granule-seule, nous créons un **indice de compréhension** (Figure 100). Cet indice est destiné à être un élément de la réaction du partenaire. Cette création d’indices d’incompréhension se fait à partir de règles telles que celle présentée Figure 101. Ces différentes règles sont construites à partir des différents motifs syntaxiques du lexique.

```
==> f-708 (INCIDENCE::indice-incomprehension (chaine " Que veux-tu prendre
dans le frigo?"))
```

Figure 100 : Indice de compréhension

```
(defrule INCIDENCE::reac-une-action-un-rangement
      (declare (salience 25))
      ?p <- (INCIDENCE::granule-seule (categorie [action])
               (texte $? ?voc $?)(granules $?grs1))
      (ANALYSEUR-LANGAGE::lexie (vocable ?voc) (categorie [action]))
      ?q <- (INCIDENCE::granule-seule (categorie [rangement])
               (texte $?tex2)(granules $?grs2))
      =>
      (retract ?p ?q)
      (bind ?chaine "")
      (bind ?chaine2 "")
      (foreach ?t $?tex2
         (bind ?chaine2 (str-cat ?chaine2 (str-cat " " ?t))))
      (bind ?chaine (str-cat "Que veux-tu " ?voc))
      (bind ?chaine (str-cat ?chaine (str-cat " dans " ?chaine2 )))
      (bind ?chaine (str-cat ?chaine " ?"))
      (assert (INCIDENCE::indice-incomprehension (chaine ?chaine)))
      (assert (MEMOIRE::granule-mem (categorie [action])
                 (texte ?voc [rangement])(granules $?grs1)))
      (assert (MEMOIRE::granule-mem (categorie [rangement])
                 (texte $?tex2)(granules $?grs2)))
      (assert (INCIDENCE::fin)))
```

Figure 101 : Règle de création d’indice d’incompréhension

3.5.2 Recherche de la tâche avec influence d'interactions précédentes

Il existe une règle qui s'exécute si aucune tâche n'est identifiée lors d'une recherche alors qu'il existe des indices mémoire qui permettent peut-être de trouver le sens de l'énoncé (Figure 102). Cette règle est déclenchée lorsqu'il existe des indices-mémoire (condition n°1) ET lorsqu'il n'y a pas de correspondance de la tâche résultat avec une tâche de l'arbre des tâches (condition n°2) ET lorsqu'il existe des éléments communs à cette tâche résultat et à une ou plusieurs définitions de tâches de l'arbre des tâches (condition n°3). Le résultat de l'analyse est envoyé au **module mémoire** pour une analyse plus complète.

```
(defrule COMPARAISONTACHE::non-corresp-arbretache-ordreincomplet-mem
      (declare (salience 8))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      ?m <- (MEMOIRE::granule-mem)                            ; cond n°1
      (tache (ident ?ident) (definition ?def))
      (test (not (COMPARAISONTACHE::member-string ?des ?def))) ; cond n°2
      (test (COMPARAISONTACHE::member-string ?def ?des))       ; cond n°3
      =>
      (retract ?p ?r)
      (assert(ANALYSEUR-LANGAGE::analysepblincomplet
             (resultat $?granules)))
      (focus MEMOIRE))
```

Figure 102 : Règle lançant la consultation de la mémoire

La mémoire contient les éléments des échanges précédents sous forme de granules qu'il est possible de recombiner avec les granules du dernier résultat pour trouver de nouveaux résultats qui retourneront vers le module de mise en comparaison avec l'arbre de tâche. Le module mémoire renvoie un nouveau résultat d'analyse qui subit la même transformation que celle d'un résultat provenant d'une analyse première, comme sur la Figure 93. De nouveaux cas de comparaison avec l'arbre des tâches prennent place.

Le premier cas est celui où à partir d'une analyse où aucun élément n'était reconnu et après passage dans la mémoire, il y a correspondance avec l'arbre des tâches (condition n°1). La tâche trouvée est envoyée au module interaction et il est procédé au nettoyage des éléments qui ont servis à l'analyse et à ceux de la mémoire (Figure 103).

```
(defrule COMPARAISONTACHE::correspondancearbretache-mem
      (declare (salience 11))
      ?r <- (ANALYSEUR-LANGAGE::compartachemem (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      ?f <- (ANALYSEUR-LANGAGE::analysepblincomplet)
      (test(COMPARAISONTACHE::member-string ?des ?def)) ; cond n°1
      =>
      (retract ?p ?r ?f)
      (assert (analyse (resultat tache ?ident)))
      (focus NETTOYAGE)
      (focus NETTOYAGE-MEMOIRE))
```
:

Figure 103 : Règle de création d’indice d’incompréhension

Un exemple de ce cas est le suivant: l’apprenant rentre « Ouvrir », puis l’énoncé suivant, « Le frigo ». « Ouvrir » est tout d’abord conservé comme indice mémoire, puis l’énoncé « le frigo » ne correspond que partiellement à des définitions de tâches de l’arbre des tâches. L’invocation du module mémoire permet de recomposer entièrement la tâche « ouvrir le frigo ». Cette tâche est utilisée comme résultat final de l’analyse pour le module interaction.

Le second cas est celui où, après passage dans la mémoire, aucune tâche de l’arbre des tâches n’est identifiée comme correspondante avec la tâche résultat de l’analyse (condition n°1) et qu’il existe des éléments communs à cette tâche résultat et à une ou plusieurs définitions de tâches de l’arbre des tâches (condition n°2) (Figure 104). La focalisation sur le module actif se déplace alors sur le module incidence-memoire.

```
(defrule COMPARAISONTACHE::noncorrespondancearbretache-2-ordreincomplet-mem
      (declare (salience 7))
      ?r <- (ANALYSEUR-LANGAGE::compartachemem (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      ?m <- (MEMOIRE::granule-mem)
      ?l <- (MEMOIRE::listeresultats-mem (resultats $?resultats))
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(not(COMPARAISONTACHE::member-string ?des ?def)))     ; cond n°1
      (test(COMPARAISONTACHE::member-string ?def ?des))          ; cond n°2
      =>
      (retract ?p ?r)
      (assert     (ANALYSEUR-LANGAGE::analysepblincomplet          (resultat
$?granules)))
      (focus INCIDENCE-MEMOIRE))
```

Figure 104 : Règle invoquant le module incidence-mémoire

Le module incidence-mémoire joue le même rôle que le module incidence mais cette fois ci utilise les granules mémoires qui ont été créées, le but est toujours la création des indices de compréhension (Figure 106). Nous reprenons le même exemple, l'utilisateur rentre « Ouvrir », puis « Le placard », la différence se fait sur la présence de granules mémoire à la place des granules seules. Nous avons alors l'indice de compréhension suivant (Figure 105).

```
==> f-834 (INCIDENCE::indice-incomprehension (chaine "Quel placard veux-tu
ouvrir ?"))
```

Figure 105 : Règle invoquant le module incidence-mémoire

Il existe une ensemble de règles comparables à celle du module incidence permettant de créer les indices de compréhension (voir annexe C).

```
(defrule INCIDENCE-MEMOIRE::reac-une-action-un-rangement
      (declare (salience 25))
         ?p <- (MEMOIRE::granule-mem (categorie [action])
               (texte $? ?voc $?)(granules $?grs1))
       (ANALYSEUR-LANGAGE::lexie (vocable ?voc) (categorie [action]))
       ?q <- (MEMOIRE::granule-mem (categorie [rangement])
               (texte $?tex2)(granules $?grs2))
       ?l <- (MEMOIRE::listeresultats-mem (resultats $?resultats))
```

```
=>
(retract ?l)
(bind ?chaine "")
(bind ?chaine2 "")
(foreach ?t $?tex2
(bind ?chaine2 (str-cat ?chaine2 (str-cat " " ?t))))
(bind ?chaine (str-cat "Quel" ?chaine2))
(bind ?chaine (str-cat ?chaine (str-cat " veux-tu " ?voc)))
(bind ?chaine (str-cat ?chaine " ?"))
(assert (INCIDENCE::indice-incomprehension (chaine ?chaine))))
```

Figure 106 : Règle créant un indice de compréhension

La suite de la création de la réaction du partenaire se fait dans le module interaction. Il est à noter que ce composant mémoire voit ses éléments supprimés lorsque qu'un énoncé correct est identifié. Il ne constitue pas une mémoire de toute l'interaction, mémoire qui pourrait être utile pour une analyse approfondie des compétences linguistiques des apprenants.

3.6 Module Interaction

La phase précédente nous permet d'identifier à quelle tâche l'apprenant fait référence ou nous permet de disposer d'indices quant à ce que celui-ci exprime. Nous nous intéressons dans cette section au module interaction et son fonctionnement, puis spécifiquement à la génération des réponses du partenaire.

3.6.1 Description

Ce module a en charge la gestion de l'interaction. Si nous reprenons le modèle de l'activité de production, celle-ci organise l'interaction comme une boucle alternant énoncé de l'apprenant et réaction du partenaire.

Le rôle de ce module d'interaction est de gérer les tours de parole, d'adapter la stratégie de dialogue à la situation [Bilange, 1992]. La décision quant à la stratégie à employer se fait en fonction des résultats de l'analyse et des éléments de la génération. Le

module interaction est décrit sous formes de tâches et méthodes [Trichet *et al.*, 1999]. Ceci permet la structuration de l'interaction :

```
(deffacts MAIN::structurer-dialogue
      (meta-tache
            (ident         t-structurer-dialogue)
            (methodes      m-structurer-dialogue))
      (meta-methode-de-decomposition-puis
            (ident          m-structurer-dialogue)
            (taches         t-ouvrir-dialogue
                            t-dialoguer
                            t-fermer-dialogue)))
```

Cette structuration générale de l'interaction permet de représenter l'interaction comme un ensemble de tâches qui disposent de méthodes pour être exécutées. La tâche t-structurer-dialogue possède une méthode m-structurer-dialogue qui permet cette structuration effective sous la forme de l'exécution consécutive de trois tâches (t-ouvrir-dialogue, t-dialoguer, t-fermer-dialogue). La première tâche t-ouvrir-dialogue lance le dialogue en faisant demander par le partenaire à l'apprenant de lui faire faire une recette. La tâche suivante est celle de dialoguer et est décomposée en deux sous-tâches qui sont celle de récupérer un énoncé de l'apprenant et celle d'y réagir tant que la tâche n'est pas accomplie. La dernière tâche est la fermeture de l'interaction.

```
(deffacts MAIN::dialoguer
     (meta-tache
            (ident        t-dialoguer)
            (methodes m-dialoguer))
     (meta-methode-de-decomposition-tant-que
            (ident                        m-dialoguer)
            (condition-d-arret            "(stop)")
            (taches                       t-saisir
                                    t-reagir)))
```

L'interaction se déroule donc jusqu'à son terme comme la succession de la saisie d'un énoncé et de la réaction qui suit. La saisie de l'énoncé est présentée paragraphe 3.2.2.2. Nous concentrons donc la suite de notre propos sur la réaction.

La tâche t-reagir est celle qui concerne les réactions du partenaire aux énoncés de l’apprenant. Elle est exécutable au travers de différentes méthodes. La méthode de réaction choisie entre toutes, l’est à partir des résultats de l’analyse et des éventuelles comparaisons avec la tâche.

```
(deffacts MAIN::reagir
     (meta-tache
       (ident    t-reagir)
       (methodes m-traiter-arret
                 m-traiter-incomprehension
                 m-traiter-question-ou
                 m-traiter-horstache
                 m-traiter-ordre-executable-bonne-couverture-bonne-syntaxe
                 m-traiter-ordre-executable-bonne-couverture-mauvaise-syntaxe
                 m-traiter-ordre-non-executable-bonne-couverture-bonne-
syntaxe
                 m-traiter-ordre-non-executable-bonne-couverture-mauvaise-
syntaxe)))
```

Toutes ces méthodes vont produire une ou des réactions qui sera ensuite envoyée au module IHM. Ces réactions sont autant des actes verbaux du partenaire que des manipulations d’objets de l’interface.

- m-traiter-arret : permet de traiter la fin de l’interaction. La réaction du partenaire est de clore l’activité interactive par un « Au revoir ».
- m-traiter-incomprehension : permet de traiter les cas où l’énoncé n’est pas compris. Cette méthode peut s’appuyer sur les indices qui lui ont été renvoyés de la phase analyse s’ils existent. Cette situation correspond aux cas (2) (5) (8) et (9) du tableau des analyses (§2.3.4.1, Tableau 6). Dans les cas (2) (4), la réaction du système dépend de ce que l’analyseur a reconnu : s’il reconnaît un objet (comme « le frigo »), le partenaire répond : *« Que veux-tu faire avec le frigo ?»* ; s’il reconnaît une action (comme « ouvrir »), il répond : *« Que veux-tu ouvrir ?»*. Dans le cas (5), l’analyseur trouve une bonne couverture syntaxique à l’aide d’hypothèses. Dans l’exemple, « les eggs » peut être un ingrédient, un contenant, ou un ustensile. Le partenaire réponde alors : *« Que veux-tu prendre dans le frigo ? »*. Dans le cas (8) l’analyseur ne reconnaît pas assez de mots pour arriver à une couverture syntaxique, même en générant des hypothèses. Le partenaire

répond alors comme dans le cas (4). Dans le cas (9), l'analyseur ne reconnaît rien. Le partenaire répond alors : *« Je ne comprends pas ce que tu écris »*.

- m-traiter-question-ou : permet de répondre aux questions de l'apprenant quant aux places des objets dans le monde. Inclus les cas où l'on ne peut pas répondre à la question.
- m-traiter-hors-tâche : permet de réagir aux énoncés corrects de l'apprenant qui ne sont pas à faire dans la tâche.
- m-traiter-ordre-executable-bonne-couverture-bonne-syntaxe : permet de réagir aux instructions de l'apprenant que l'on peut accomplir. Ces énoncés étant parfaitement formés.
- m-traiter-ordre-executable-bonne-couverture-mauvaise-syntaxe : permet de réagir aux instructions de l'apprenant que l'on peut accomplir. Ces énoncés n'étant pas parfaitement formés.
- m-traiter-ordre-non-executable-bonne-couverture-bonne-syntaxe : permet de réagir aux instructions de l'apprenant que l'on ne peut pas accomplir. Ces énoncés étant parfaitement formés.
- m-traiter-ordre-non-executable-bonne-couverture-mauvaise-syntaxe : permet de réagir aux instructions de l'apprenant que l'on ne peut pas accomplir. Ces énoncés n'étant pas parfaitement formés.

Les quatre dernières méthodes s'illustrent dans les cas (1) (3) (6) (7), la réaction du système dépend de l'état du monde : soit l'action est possible (le frigo est ouvert), alors l'action est effectuée à l'interface (les méthodes ordre-exécutable), soit l'action est impossible (le frigo est fermé), dans ce cas, le partenaire répond : *« Je ne peux pas exécuter ta demande. »* (les méthodes ordre-non-exécutable).

3.6.2 Fonctionnement

Le fonctionnement global du module d'interaction consiste en l'empilement et le dépilement des tâches et méthodes.

Lorsque que le système attend un énoncé de la part de l'apprenant, la tâche en haut de la pile est celle de réagir (Figure 107). Une méthode de la tâche t-réagir est exécutable quand les pré-conditions de son exécution sont remplies par les éléments produits par

l'analyse et la génération. Ceci déclenche l'exécution de la tâche possédant cette méthode, cette dernière la remplaçant dans la pile d'exécution (flèche 1). Le dépilement de cette dernière provoque sa propre exécution, c'est à dire envoyant au module IHM les instructions quant aux interventions du partenaire dans le cas de la Figure 107 (flèche 2). La méthode m-dialoguer est ensuite exécutée, remettant en tête de pile les méthodes t-saisir et t-réagir tant que l'activité n'est pas terminée (flèche 3).

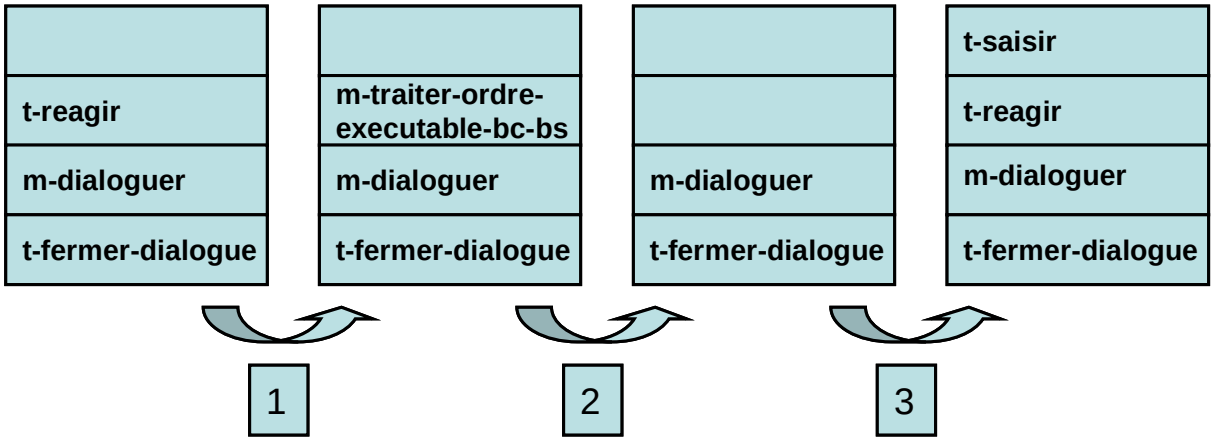


Figure 107 : Evolution de la pile d'exécution

Dans le cas des méthodes du type m-traiter-ordre-exécutable de la tâche t-réagir, elles sont déclenchées par l'identification de la tâche. Ce sont des méthodes opérationnelles, l'opération qui leur est associée provoque l'activation d'un module où l'on retrouve les règles qui vont entraîner l'exécution ou non des tâches identifiées. Nous expliquons ce fonctionnement au travers de l'exemple suivant (Figure 108). La méthode m-traiter-ordre-executable-bonne-couverture-bonne-syntaxe a parmi ses préconditions le fait que la tâche trouvée après la comparaison avec l'arbre des tâches appartienne à la liste des méthodes activables et des méthodes opérationnelles exécutable. C'est au travers de ces pré-conditions que l'on voit comment le module interaction consulte les informations relatives à l'avancée de la tâche applicative. Si les pré-conditions sont remplies, la fonction (l'opération) f-traiter-ordre-executable-bcbs est alors exécutée.

```
(deffacts MAIN::traiter-ordre-executable-bonne-couverture-bonne-syntaxe
     (meta-methode-operationnelle
        (ident   m-traiter-ordre-executable-bonne-couverture-bonne-syntaxe)
        (preconditions "(MAIN::analyse (resultat tache ?ident))"
            "(not(ANALYSEUR-LANGAGE::analysemsbc (resultat $?granules)))"
            "(MAIN::tache (ident ?ident) (methodes $?methodes))"
            "(MAIN::methodes-activables $?methodes-activables)"
            "(MAIN::methodes-operationnelles-executables $?methodes-
operationnelles-executables)"
            "(test (not (MAIN::vide$ (MAIN::intersection3$ $?methodes
$?methodes-activables $?methodes-operationnelles-executables))))")
        (fonction      MAIN::f-traiter-ordre-executable-bcbs)))
```

Figure 108 : méthode opérationnelle traiter-ordre-executable-bonne-couverture-bonne-syntaxe

L'exécution de la fonction f-traiter-ordre-executable-bcbs déclenche l'activation du module TRAITER-ORDRE-EXECUTABLE-BCBS (Figure 109). Dans ce module, une seule règle : elle entraîne l'exécution de la méthode opérationnelle associée à la tâche résultat de la comparaison avec l'arbre des tâches. La tâche réalisée est ajoutée à la liste des tâches exécutées. Le niveau interaction contrôle le niveau tâche en commandant l'exécution des tâches de l'arbre des tâches. Le module INTERFACE est alors activé car un fait INTERFACE va être produit par l'exécution de la méthode opérationnelle de la tâche résultat. Ce fait déclenche une action dans le micromonde.

```
(defmodule TRAITER-ORDRE-EXECUTABLE-BCBS);;
(deffunction MAIN::f-traiter-ordre-executable-bcbs ()
      (focus TRAITER-ORDRE-EXECUTABLE-BCBS))

(defrule TRAITER-ORDRE-EXECUTABLE-BCBS::r-traiter-ordre-executable-bcbs
      ?p <- (MAIN::analyse (resultat tache ?ident))
      (MAIN::tache (ident ?ident) (texte ?texte) (methodes $? ?m $?))
      (MAIN::methodes-operationnelles-executables $? ?m $?)
      (methodes-activables $? ?m $?)
      (MAIN::methode-operationnelle (ident ?m) (fonction ?f))
      ?q <- (taches-executees $?taches)
      ?r <- (enonce (texte ?texte1))
      =>
      (retract ?p ?q ?r)
      (apply ?f)
      (assert (a-ecrire "J'ai bien compris ton ordre, et je l'exécute !"))
    (focus INTERFACE)
    (assert (taches-executees $?taches ?ident))
    (printout t "assertion tâches exécutées" $?taches crlf)
    (pop-focus))
```

Figure 109 : module TRAITER-ORDRE-EXECUTABLE-BCBS

Le module interaction est celui qui recèle encore aujourd’hui le plus de développement potentiels : l’usage des UMIs permettrait comme suggéré dans les modèles d’interaction de débloquer des situations de trop grande complexité. De même, la possibilité d’ajouter de nouvelles structures de dialogue dans le module est à l’étude. L’architecture devrait être revue : le module interaction devrait récupérer le contrôle de tous les autres modules, en particulier celui de commander l’analyse et la tâche. Ceci est possible en incluant dans la définition de la tâche interagir des sous-tâches pour l’analyse et pour la génération.

3.7 Mises à l’essai

Le logiciel a été testé au cours de deux périodes auprès d’étudiants thaïlandais et chinois en année d’adaptation à l’Université du Maine. Ces apprenants sont débutants en français et passent cette année d’adaptation à l’apprendre pour ensuite intégrer les cursus de l’Université. La première séance de mises à l’essai a concerné la recherche de bugs, l’observation des premières utilisations et les réactions des apprenants devant l’interface

graphique et les possibilités d'interaction. La seconde mise à l'essai s'est attachée à la validation de la réactivité du système, à savoir la capacité de réagir aux énoncés problématiques et à l'observation des nouveaux comportements dans l'usage de l'environnement. Dans cette section, nous présentons le déroulement de ces tests et les observations que nous en avons retirées.

3.7.1 Premières mises à l'essai, hiver 2004-2005

Cette première séquence d'essais a visé à tester l'utilisabilité de l'environnement informatique. Les questions qui motivaient ces tests étaient : que va-t-il se passer lorsque l'environnement sera placé entre les mains de réels utilisateurs ? L'environnement est-il assez robuste ? Quelles observations pouvons-nous retirer des premières traces obtenues ? Quelles sont les remarques des apprenants au niveau de l'ergonomie du logiciel ? Les tests ont concerné trois binômes composés d'étudiants thaïlandais. Ces tests se sont déroulés au LIUM. Les étudiants étaient seuls en binôme avec un observateur.

3.7.1.1 Déroulement des mises à l'essai

L'observateur a présenté aux étudiants le logiciel : le monde de la cuisine, l'activité à accomplir, ainsi que les modes d'interaction avec le partenaire. L'activité a été sommairement expliquée ainsi : « vous devez dire à votre partenaire comment réaliser la recette en lui donnant des instructions ». Pour pallier au fait qu'il n'y avait pas eu d'activité de compréhension qui aurait permis aux apprenants d'évaluer le champ des expressions linguistiques utilisées par le système, les apprenants disposaient d'une liste de mots désignant les objets du monde, et d'actions qu'ils pouvaient demander au partenaire (cette liste est présentée en annexe D). Lors de l'activité, l'observateur ne s'est pas mis dans une position faussement passive [Chanier, 2000] : il répondait aux questions que posaient les étudiants en particulier sur des objets qu'ils n'arrivaient pas à relier aux mots de la liste. D'un point de vue systémique, ces interactions font partie de l'activité pédagogique. Les apprenants collaboraient à l'écriture des énoncés à écrire à l'intention du partenaire virtuel. Des problèmes techniques sont survenus. Le logiciel s'est arrêté plusieurs fois et des problèmes au niveau de l'analyse et de la génération ont été constatés : phrases parfaites non comprises, énoncés incorrects de la part du partenaire. Ces problèmes ont déstabilisé les apprenants les obligeant à discuter longuement pour trouver les bons énoncés qui ne déclencheraient pas de

problèmes. Ces discussions ont eu lieu en français. De même l’implication dans l’activité était difficile à susciter du fait de ces problèmes techniques. Aucun des groupes n’a eu recours à aucun moment de l’activité à la possibilité de poser des questions au partenaire alors que cette possibilité leur avait été expliquée. Même lorsqu’ils ne pouvaient plus avancer dans la réalisation de la tâche applicative, nous mettons cela sur le fait de la présence de l’observateur qu’ils considéraient comme un interlocuteur privilégié pour leurs questions et également sur l’apparente faillibilité du système.

Les interactions écrites ont été enregistrées sous la forme de fichiers textes (un des textes est présenté en annexe A). Les énoncés des apprenants et les énoncés du partenaire virtuel). Pour tous les groupes, l’activité a duré quarante cinq minutes.

3.7.1.2 Observations

De nombreux problèmes techniques ont été constatés. Ces premiers tests ont mis en lumière des problèmes de programmation, en particulier au niveau des modules analyse et génération. Les problèmes découverts ne remettaient pas en cause les modèles mais demandaient à repenser certaines parties du code. Le module d’analyse présentait de nombreuses instabilités. La gestion de la mémoire de l’interaction était aléatoire. Les granules produites se chevauchaient, rendant très longue la production de l’analyse. Nous avons même perdu les traces des interactions de ce binôme. Par contre, en ce qui concerne la gestion de la tâche applicative, de l’interaction, aucun problème n’a été constaté. De même, les déplacements des objets sur l’interface graphique se sont effectués sans problème.

Nous avons pu par contre conserver la trace d’un des binômes. Dans les traces, nous avons retrouvé les différents cas d’interaction attendus. Sur 46 échanges entre apprenants et partenaire virtuel, nous avons comptabilisé trente-deux échanges liés à la tâche sans problèmes de communication (Figure 110), quatorze échanges avec des problèmes linguistiques (Figure 111), et un échange avec un problème concernant la réalisation de la tâche (Figure 112). Dans les traces, « Partenaire » désigne le partenaire virtuel. Lorsque le partenaire déclare faire une action (comme Figure 110), l’action a aussi lieu sur l’interface.

```
Apprenants>  prendre le lait
Partenaire> Je sors le lait du frigo.
Apprenants>  prendre la crème
Partenaire> Je prends la crème dans le frigo.
```

Figure 110 : Échanges sans problèmes

```
Apprenants>  ouvrir le placard du haut a droit
Partenaire> Je ne te comprends pas lorsque tu écris ouvrir le
            placard du haut a droit.
Apprenants>  ouvrir le placard du haut a droite
Partenaire> J'ouvre le placard.
```

Figure 111 : Échanges avec problèmes linguistiques

Il est à noter ici que les réponses du partenaire sont pauvres et ne pointent pas forcément sur le problème de façon très claire. Ceci constitua un des chantiers de correction du logiciel à partir de ces tests. Ici il aurait été intéressant que le système réponde : « Je ne te comprends pas lorsque tu écris : « du haut a droit », quel placard veux-tu ouvrir ? ».

```
Apprenants>  prendre le lait
Partenaire> J'ai bien compris ton ordre, mais je ne peux pas
            l'exécuter !
```

Figure 112 : Échanges avec problèmes liés à la tâche

D'un point de vue « production », les énoncés des apprenants ont été relativement cours (de 1 à 7 mots) mais ceci est fortement conditionné par la tâche. Ce constat nous incite à imaginer des situations amenant à des productions plus complexes, même si cette simplicité a parfaitement convenu à notre public. Les énoncés à structure syntaxique « verbe + groupe nominal (GN) » et « verbe + GN + dans + GN » ont été majoritaires. La figure ci-dessous illustre ce phénomène pour un binôme (Figure 113).

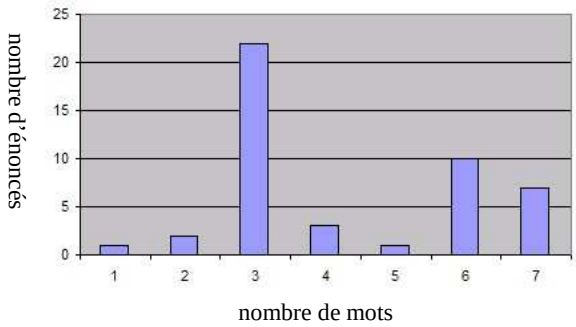


Figure 113 : Répartition de la taille des énoncés pour un binôme

A l'issue des tests, les apprenants ont eu un court entretien (dix minutes) avec l'observateur où ils pouvaient critiquer l'environnement. De ces entretiens est ressorti le manque d'une animation sonore : les apprenants auraient souhaité que le partenaire leur « parle » et que les actions à l'interface déclenchent des sons dans le micromonde. Un autre manque qu'ils ont souligné est celui de l'animation du partenaire : ils auraient souhaité que celui-ci ait des expressions significatives quant à leur réussite dans les interactions. Ils auraient aussi désiré que le partenaire fasse avec eux le point sur l'avancement de la tâche plus souvent et marque les étapes de celle-ci. Ils ont aussi pointé les faiblesses du logiciel en matière de réponse et le manque d'explication quand certains énoncés n'étaient pas reconnus.

3.7.2 Deuxièmes mises à l'essai, hiver 2005-2006

La seconde période de mises à l'essai affine les questionnements de la première. La question centrale est, après les corrections faites sur le prototype, si le support à l'interaction fonctionne, à savoir **si les réactions aux énoncés se font selon les modèles interactifs construits et sont compréhensibles par les apprenants**. L'autre questionnement concerne les comportements des apprenants dans une situation où le système est stable, c'est-à-dire où sa faillibilité n'est pas en cause.

3.7.2.1 Déroulement des mises à l'essai

Deux binômes issus de l'année d'intégration de l'Université du Maine ont participé aux expérimentations. Ils étaient composés d'étudiants thaïlandais et chinois. Le dispositif des tests précédents est conservé. Une liste des termes utiles à la réalisation de la recette était fournie aux apprenants. L'observateur présentait de la même façon l'activité à réaliser.

Les mises à l'essai avec un système bien plus stable ont vu de les apprenants adopter de nouveaux comportements. La possibilité de poser des questions a été employée par les deux groupes. L'implication dans la réalisation de la tâche a été immédiate et sans retenue de la part des apprenants, contrairement aux tests précédents où les problèmes techniques freinaient les apprenants. Les apprenants ont beaucoup pratiqué de travail réflexif, s'auto-corrigeant entre eux lors de l'écriture de leurs énoncés. Les questions à l'adresse de l'observateur ont été plus rares et portaient sur des éléments de la tâche pédagogique (par exemple : « Quel mode utiliser : infinitif ou impératif ? »). Des formes de langue nouvelles sont aussi apparues, incluant des formules de politesse, que nous n'avions pas incluses dans la définition des motifs syntaxiques. Ceci n'est pas problématique au regard du système d'interaction car ces formules sont ignorées et le sens de l'énoncé saisi, par contre dans une perspective d'apprentissage, passer à côté de ces informations qui témoignent de la richesse des structures employées est dommageable. Ces questions et ces formulations dénotent les limites du prototype dans sa forme actuelle, les apprenants auraient voulu varier les formes linguistiques employées.

Les traces des énoncés des apprenants ainsi que ceux du partenaire virtuel ont été conservées et sont disponibles dans l'annexe A. De même que lors des premiers tests, l'accomplissement de l'activité de production par les apprenants a duré quarante cinq minutes.

3.7.2.2 Observations

Les problèmes techniques ont presque disparu, les dernières hésitations de l'analyse ont été détectées et corrigées. Quelques malformations de réaction apparaissent de temps à autre dans les traces. Elles amènent des pertes d'informations qui pénalisent les échanges qui suivent immédiatement mais ces problèmes disparaissent grâce à l'effacement de la mémoire de l'interaction lorsque un nouvel énoncé correct est entré. Mais l'environnement est désormais stabilisé.

Les traces des interactions révèlent l'apparition de phénomènes nouveaux et la confirmation des cas d'interactions prévus. Ainsi nous pouvons par exemple, relever cette séquence (Figure 114) qui correspond aux situations de négociation que nous avons définies dans la partie interaction (§ 2.2.3.2). Dans cet exemple, les apprenants commettent une erreur « prend ». Le terme « le lait » est bien compris et le partenaire cherche à savoir ce que les

apprenants veulent faire avec le lait. Les apprenants entrent l’énoncé « le prendre ». Grâce à la mémoire de l’interaction, l’action « prendre » est associée avec l’ingrédient « le lait », formant le motif syntaxique connu « prendre le lait ». L’instruction est reconnue et l’interaction peut avancer.

```
Apprenants > prend le lait
Partenaire > Je ne te comprends pas lorsque tu écris "prend le lait". Que
veux tu faire avec le lait ?
Apprenants > le prendre
Partenaire > J'ai bien compris ton ordre, et je l'exécute !
Partenaire > Je sors le lait du frigo
```

Figure 114 : séquence de négociation

Les mécanismes de réaction aux instructions incomplètes ont été améliorés, affinant les énoncés du partenaire en fonction de critère tel que le genre des éléments reconnus. La Figure 115 illustre ce phénomène : l’énoncé « prendre le pot » est reconnu mais le pot en question n’est pas désigné. Les apprenants ont décidé de ne pas répondre à la question et ont préféré se consacrer à un autre énoncé.

```
Apprenants > prendre le pot
Partenaire > Je ne te comprends pas lorsque tu écris "prendre le pot".
Quel pot veux-tu prendre ?
Apprenants > prendre le saladier
Partenaire > J'ai bien compris ton ordre, et je l'exécute !
Partenaire > Je sors le saladier du placard
```

Figure 115 : exemple de réaction

Reprenant des critiques émises lors de la période d’essais précédentes (absence de son, animation du partenaire, point sur l’avancement de la tâche), les entretiens de la seconde période d’essais ont aussi permis de mettre d’autres points en avant. Certains apprenants ont regretté de ne pas pouvoir cliquer sur des éléments du micromonde pour demander au partenaire ce qu’ils représentaient. Cette fonctionnalité existe par contre dans l’environnement MEPA. Ce manque est lié au problème qu’il n’y a pas eu d’activité de compréhension préalable à l’activité de production. En terme de scénarisation, il serait judicieux de faire faire l’activité de compréhension avant celle de production. Ils ont aussi regretté l’absence de fausses pistes qui leur feraient rater la recette !

Le bilan de ces dernières mises à l'essai met en évidence deux points :

- le mécanisme de détection et réaction aux non attendus dans le dialogue est fonctionnel.
- Par contre, prétendre que nous avons là une réelle situation d'apprentissage est prématurée, trop de phénomènes linguistiques ne peuvent être détectés et corrigés. De même, les énoncés du partenaire restent stéréotypés.

CONCLUSION ET PERSPECTIVES

Bilan des travaux

Nous nous sommes intéressés dans cette thèse à la mise en place d'un type d'activités pédagogiques particulier pour l'apprentissage d'une langue étrangère et au support à l'interaction dans le cadre de ces activités.

Nous avons identifié trois dimensions d'analyse et de conception que nous présentons sous la forme de trois pôles complémentaires et interdépendants : la tâche, la langue et l'interaction, chaque pôle se définissant à partir des deux autres. Cette méthode nous a permis de construire des modélisations interdépendantes. Ceci dit, nous ne sommes pas focalisés sur ces trois dimensions avec la même insistance, et les solutions apportées ne sont pas de même ordre. Ainsi, le pôle « langue » est abordé plus rapidement et certains aspects des solutions informatiques sont plus ad hoc. Les modèles fondent la conception de l'environnement sur des éléments issus de l'apprentissage des langues, du dialogue homme-machine et des EIAH. Les modèles du pôle tâche décrivent les tâches pédagogiques (et leur dimension interactive) qui permettent d'impliquer un apprenant dans une interaction avec un partenaire au sein d'un micromonde, et la tâche applicative prétexte à l'interaction. Les modèles du pôle interaction décrivent les représentations et les processus qui permettent au système de gérer les échanges (langagiers et non langagiers) entre l'apprenant et son partenaire. Les modèles du pôle langue décrivent la structure du lexique ainsi que les processus d'analyse des énoncés de l'apprenants et de génération des énoncés du partenaire. Ces derniers modèles sont conçus pour permettre au système de réagir aux énoncés incorrects des apprenants grâce à des stratégies de feed-back, en fonction des éléments reconnus dans les énoncés de l'apprenant. Ces stratégies sont pensées pour continuer l'interaction autour de la tâche applicative.

Sur la base de nos modèles, nous avons conçu et développé un prototype logiciel appelé Sampras. L'environnement propose pour l'instant aux apprenants une activité de production langagière. Certains des composants de l'environnement, comme le micromonde, peuvent être réutilisés pour d'autres applications s'inspirant de la même approche ou nécessitant des interactions de même type. En effet, le micromonde et son interface graphique sont génériques : la scène, les objets manipulables et leurs comportements sont décrits dans des fichiers XML et CLP (règles Jess). De même, l'analyseur d'énoncés, même s'il est encore

très améliorable (absence de traitement morphosyntaxique), peut être réutilisé tel quel, ses connaissances (lexique au format XML) étant totalement séparées des traitements. L'analyseur dans sa forme actuelle pourrait devenir un composant qui serait situé entre un analyseur qui couvrirait vraiment le champ linguistique délimité par le micromonde et le module de génération.

Si nous comparons nos travaux à l'environnement FLUENT, le plus proche de Sampras, il existe un nombre de différences intéressantes à signaler. Pour Fluent, Reeder recherchait des techniques pour caractériser l'interaction, nous proposons de nous fonder sur les travaux en acquisition en langue et de les déployer dans les activités interactives en s'inspirant de techniques du dialogue homme-machine. Nous sommes alors partis dans une optique de détection et gestion des non attendus dans les énoncés des apprenants pour permettre la continuation de l'interaction entre les apprenants et le partenaire. Par contre, FLUENT est un environnement bien plus complet que le notre. Il dispose en effet d'un analyseur permettant un diagnostic assez fin des compétences linguistiques des apprenants. La tâche est paramétrable : à chaque plan et même action de la tâche applicative peut être associé un type d'interaction lié à une compétence à exercer et des contraintes au niveau de la langue à employer. Ceci permet aux professeurs de programmer la tâche donnée. Nos activités sont plus figées ne mélangeant pas les types d'interactions liés aux compétences et le paramétrage des actions de la tâche et même son écriture en fonction d'objectifs linguistiques n'est pas possible à l'heure actuelle.

Nous avons pu expérimenter l'usage de Sampras auprès d'apprenants débutants en français langue étrangère. Les premières « mises à l'essai » ont permis de corriger des défauts des modules d'analyse et de génération ainsi que de l'interface. Elles n'ont pas remis en cause les modèles que nous avons conçu. Les expériences suivantes ont confirmé la stabilisation de l'environnement. Ces tests soulèvent plus de question au niveau de l'ergonomie et de l'organisation de la tâche interactive. Pour l'organisation de la tâche interactive, il s'agit de problème d'aide à l'apprenant quand il est perdu dans la tâche et de problèmes de réponse aux énoncés qui sortent du cadre de la tâche applicative. De même, revoir l'analyse nous permettrait d'accéder à de réelles situations d'apprentissage.

Perspectives

Nous sommes conscients que ce travail souffre d'un manque d'expérimentation. Les premiers tests sont limités en nombre et circonscrits à un même public, celui des étudiants asiatiques en année d'intégration à l'université du Maine. Mener de nouvelles campagnes d'essai auprès d'autres milieux d'apprenants nous semble essentiel.

Avant ces nouveaux tests, un nouveau travail au niveau de l'analyse des énoncés est à effectuer. La variété des énoncés à appréhender demande plus de souplesse au niveau morphosyntaxique. Si le problème des structures syntaxiques nouvelles (comme les formules de politesse) est simple à intégrer en ajoutant de nouveaux motifs syntaxiques, celui des conjugaisons, pluriels et genre demande de poser une nouvelle dimension dans l'analyse. A partir d'une évaluation complète du champ linguistique déterminé par le micromonde, nous pourrions constituer une nouvelle analyse qui permettrait de s'attacher à des problèmes linguistiques déterminés et donc de construire des réactions du partenaire plus pertinentes dans une optique d'apprentissage, en se servant du travail déjà fait au niveau de la gestion de l'interaction.

Au niveau du prototype, de nombreuses évolutions sont souhaitables. Du côté de l'interface Java, une évolution concerne un des problèmes soulignés par l'ensemble des apprenants ayant testé Sampras. Ce problème est le manque d'animation du partenaire. Ils nous paraît important, comme il l'a paru aux apprenants, que le partenaire puisse exprimer des émotions en fonction des énoncés de l'apprenant. Par ailleurs, beaucoup ont regretté l'absence totale de sons (interventions du partenaire, bruits dans la cuisine), certaines améliorations d'ordre « cosmétique » (comme les bruits de cuisine) sont assez facilement envisageables, d'autres, comme les expressions du partenaire, exigent certainement de nouvelles études et recherches.

Du côté de la partie Jess, des évolutions sont aussi à prévoir. Une évolution à considérer concerne l'écriture de la tâche applicative. Cette écriture est longue, rébarbative, nécessitant de nombreux tests jusqu'à arriver à une stabilisation. La création d'un outil permettant de créer de façon systématique la tâche applicative permettrait de résoudre ce problème majeur d'utilisation du logiciel. De même, la description des procédures dialogiques

n'est pas accessible et modifiable pour un utilisateur. Ce point fait l'objet d'un stage de recherche au LIUM.

Nous terminerons en précisant qu'il nous semblerait intéressant d'appliquer les modèles et de réutiliser les modules créés sur des tâches plus simples et moins longues. Les tâches applicatives se trouveraient utilisées dans les scènes de simulations globales telles que celles proposés dans l'environnement Mepa[4].

[4] http://lucke.univ-lemans.fr/mepa

ANNEXES

ANNEXE A – TRACES D'UTILISATION DE SAMPRAS..... 173

ANNEXE B – IMPLÉMENTATION DE SAMPRAS 192

ANNEXE C – LEXIQUE DE SAMPRAS..... 215

ANNEXE D – DOCUMENT POUR LES APPRENANTS..... 221

Annexe A – traces d'utilisation de Sampras

Cette première annexe présente quelques traces d'utilisation de Sampras. Ces traces incluent les dialogues en langue naturelle ayant eu lieu entre le partenaire et les apprenants. Nos ne présentons pas toutes les traces mais simplement trois d'entre elles. Les erreurs liées à des problèmes de programmation (problèmes de génération corrigés après leurs découvertes) ont été laissées.

Trace d'utilisation de Sampras (janvier 2005)

Partenaire > Bonjour, tu vas me dire comment réaliser un flan au chocolat
Partenaire > Je t'écoute...

Apprenants > ouvrir le frigo
Partenaire >J'ouvre le frigo

Apprenants > prendre le lait
Partenaire > Je sors le lait du frigo

Apprenants > ouvrir le placard
Partenaire > Je ne te comprends pas lorsque tu écris "ouvrir le placard". Quel le placard veux-tu ouvrir ?

Apprenants > ouvrir du bas
Partenaire > je ne te comprends pas lorsque tu écris ouvrir du bas.

Apprenants > ouvrir le placard du bas
Partenaire > je ne te comprends pas lorsque tu écris ouvrir le placard du bas.

Apprenants > prendre la creme
Partenaire > Je prends la crème dans le frigo

Apprenants > ouvrir le placard du bas a gauche

Partenaire > J'ouvre le placard

Apprenants > prendre la cuillere

Partenaire > Je sors la cuillère du placard

Apprenants > ouvrir le placard du bas a droite

Partenaire > J'ouvre le placard

Apprenants > ouvrir le placard du bas au milieu

Partenaire > J'ouvre le placard

Apprenants > prendre le doseur

Partenaire > Je sors le doseur du placard

Apprenants > ouvrir le placard du haut a droit

Partenaire > je ne te comprends pas lorsque tu écris ouvrir le placard du haut a droit.

Apprenants > ouvrir le placard du haut a droite

Partenaire > J'ouvre le placard

Apprenants > ouvrir le placard du haut a gauche

Partenaire >J'ouvre le placard

Apprenants > ouvrir le placard du haut au milieu

Partenaire > J'ouvre le placard

Apprenants > prendre le saladier

Partenaire > Je sors le saladier du placard

Apprenants > prendre les oeufs

Partenaire > Je sors les oeufs du frigo

Apprenants > ouvrir la boite a les oeufs

Partenaire > J'ouvre la boite a oeufs

Apprenants > extirper les oeufs

Partenaire > Je prends les oeufs dans la boîte à oeufs

Apprenants > mettre les oeufs dans le saladier

Partenaire > Je mets les oeufs dans le saladier

Apprenants > poser la creme dans le saladier

Partenaire > je ne te comprends pas lorsque tu écris poser la creme dans le saladier.

Apprenants > melanger

Partenaire > Je ne te comprends pas lorsque tu écris "melanger ". Que veux-tu melanger ?

Apprenants > melanger les oeufs

Partenaire > je ne te comprends pas lorsque tu écris melanger les oeufs .

Apprenants > poser la creme dans le saladier

Partenaire > je ne te comprends pas lorsque tu écris poser la creme dans le saladier.

Apprenants > prendre le lait

Partenaire > J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > prendre la farine

Partenaire > Je prends la farine dans le frigo

Apprenants > prendre le sucre

Partenaire >Je prends le sucre dans le placard

Apprenants > preparer la creme dans le saladier

Partenaire > je ne te comprends pas lorsque tu écris preparer la creme dans le saladier.

Apprenants > mettre le sucre dans le saladier

Partenaire > Je ne te comprends pas lorsque tu écris "mettre le sucre dans le saladier". Que veux tu faire avec le sucre ?

Apprenants > incorporer le sucre

Partenaire > J'incorpore le sucre

Apprenants > incorporer la farine

Partenaire > J'incorpore la farine

Apprenants > mettre le lait

Apprenants > verser le lait

Partenaire > Je verse le lait

Apprenants > melanger le flan

Partenaire > Je mélange le tout

Apprenants > ouvrir le four

Partenaire > J'ouvre le four

Apprenants > prendre le moule

Partenaire > Je sors le moule du placard

Apprenants > mettre le flan dans le moule

Partenaire > je ne te comprends pas lorsque tu écris mettre le flan dans le moule.

Apprenants > verser le flan dans le moule

Partenaire > Je verse le flan dans le moule

Apprenants > mettre le moule dans le four

Partenaire > Je mets le moule dans le four

Apprenants > fermer le four

Partenaire > Je ferme le four

Apprenants > faire cuire

Partenaire > je ne te comprends pas lorsque tu écris faire cuire.

Apprenants > faire cuire le flan

Partenaire > Je fais cuire le flan

Apprenants > ouvrir le four

Partenaire > J'ouvre le four

Apprenants > sortir le flan

Partenaire > je ne te comprends pas lorsque tu écris sortir le flan.

Apprenants > sortir le moule de le four

Partenaire > je ne te comprends pas lorsque tu écris sortir le moule de le four.

Apprenants > Au revoir

Partenaire > D'accord, arrêtons-nous là...

Partenaire > Au revoir...

Trace d'utilisation de Sampras (mars 2006)

Partenaire >Bonjour, tu vas me dire comment réaliser un flan au chocolat

Partenaire >Je t'écoute...

Apprenants > la cusine

Partenaire >je ne te comprends pas lorsque tu écris la cusine.

Apprenants > qu'est ce qu'on fiat?

Partenaire >je ne te comprends pas lorsque tu écris qu est ce qu on fiat .

Apprenants > bonjour

Partenaire >je ne te comprends pas lorsque tu écris bonjour.

Apprenants > bonjour

Partenaire >je ne te comprends pas lorsque tu écris bonjour.

Apprenants > bonjour

Partenaire >je ne te comprends pas lorsque tu écris bonjour.

Apprenants > comment ferme le foyer?

Partenaire >je ne te comprends pas lorsque tu écris comment ferme le foyer .

Apprenants > ouvrir?

Partenaire >Je ne te comprends pas lorsque tu écris "ouvrir ". Que veux-tu ouvrir ?

Apprenants > faire quoi?

Partenaire >Je ne te comprends pas lorsque tu écris "faire quoi ". Que veux-tu faire ?

Apprenants > cuisine

Partenaire >je ne te comprends pas lorsque tu écris cuisine.

Apprenants > ouvrir le frigo

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le frigo

Apprenants > prend le lait

Partenaire >Je ne te comprends pas lorsque tu écris "prend le lait". Que veux tu faire avec le lait ?

Apprenants > le prendre

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je sors le lait du frigo

Apprenants > ouvriv le placard

Partenaire >Je ne te comprends pas lorsque tu écris "ouvriv le placard". Que veux tu faire avec le placard ?

Apprenants > ouvrir le placard

Partenaire >Je ne te comprends pas lorsque tu écris "ouvrir le placard". Quel placard veux-tu ouvrir ?

Apprenants > tous

Partenaire >je ne te comprends pas lorsque tu écris tous.

Apprenants > ovrir le placard à gauche

Partenaire >Je ne te comprends pas lorsque tu écris "ovrir le placard a gauche". Que veux-tu ranger dans le placard [attribut] ?

Apprenants > ouvrir le placard à gauche au milieu

Partenaire >je ne te comprends pas lorsque tu écris ouvrir le placard a gauche au milieu.

Apprenants > ouvrir le placard du haut à droite

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvrir le placard du haut à gauche

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvrir le placard du haut au milieu

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvrir le placard du bas à gauche

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvir le placard du bas au milieu

Partenaire >Je ne te comprends pas lorsque tu écris "ouvir le placard du bas au milieu". Que veux tu faire avec au milieu ?

Apprenants > ouvrir le placard du bas à droite

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvrir le placard du bas au milieu

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants >prendre le pot

Partenaire >Je ne te comprends pas lorsque tu écris "prendre le pot". Que veux le pot [contenu] veux-tu prendre ?

Apprenants > prendre la creme

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je prends la crème dans le frigo

Apprenants > prendre les oeufs

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je sors les oeufs du frigo

Apprenants > prendre le sucre

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je prends le sucre dans le placard

Apprenants > ouvrir les oeufs

Partenaire >Je ne te comprends pas lorsque tu écris "ouvrir les oeufs". Que veux-tu ouvrir [contenant] ?

Apprenants > ouvrir la boite à les oeufs

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre la boite a oeufs

Apprenants > prendre le beurre

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Apprenants > prendre la farine

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je prends la farine dans le frigo

Apprenants > prendre le chocolat

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je prends le chocolat dans le placard

Apprenants > prendre le saladier

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je sors le saladier du placard

Apprenants > prendre le moule

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je sors le moule du placard

Apprenants > prendre la bouteille

Partenaire >Je ne te comprends pas lorsque tu écris "prendre la bouteille". Que veux la bouteille veux-tu prendre ?

Apprenants > extiper les oeufs

Partenaire >J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > mettre les oeufs dans le saladier

Partenaire >J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > ouvrir la farine

Partenaire >Je ne te comprends pas lorsque tu écris "ouvrir la farine". Que veux-tu ouvrir [contenant] ?

Apprenants > extiper les oeufs

Partenaire >Je ne te comprends pas lorsque tu écris "extiper les oeufs". Que veux tu faire avec la farine ?

Apprenants > extirper les oeufs

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je prends les oeufs dans la boîte à oeufs

Apprenants > mettre la farine

Partenaire >Je ne te comprends pas lorsque tu écris "mettre la farine". Que veux-tu mettre [ingredient] sur le plan ?

Apprenants > mettre la farine dans le saladier

Partenaire >Je ne te comprends pas lorsque tu écris "mettre la farine dans le saladier". Que veux-tu mettre dans le saladier ?

Apprenants > mettre le sucre dans le saladier

Partenaire >Je ne te comprends pas lorsque tu écris "mettre le sucre dans le saladier". Que veux-tu mettre dans le saladier ?

Apprenants > incorporer la farine

Partenaire >J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > mettre les oeufs dans le saladier

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je mets les oeufs dans le saladier

Apprenants > incorporer la farine

Partenaire >J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > incorporer le sucre

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'incorpore le sucre

Apprenants > incorporer la farine

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'incorpore la farine

Apprenants > verser le lait

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je verse le lait

Apprenants > melanger le flan

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je mélange le tout

Apprenants > verser le flan dans le moule

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je verse le flan dans le moule

Apprenants > ouvrir le four

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le four

Apprenants > mettre le moule dans le four

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je mets le moule dans le four

Apprenants > fermer le four

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je ferme le four

Apprenants > cuire le flan

Partenaire >Je ne te comprends pas lorsque tu écris "cuire le flan". Que veux tu faire avec le flan ?

Apprenants > faire cuire le flan

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je fais cuire le flan

Apprenants > ouvrir le four

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le four

Apprenants > sortir le flan

Partenaire >Je ne te comprends pas lorsque tu écris "sortir le flan". Que veux-tu sortir [ingredient] ?

Apprenants > sortir le moule de le four

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je sors le moule du four et voilà le flan est prêt!!

Apprenants > fermer le four

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je ferme le four

Apprenants > fermer le placard du haut à gauche

Partenaire >J'ai bien compris ton ordre, et je l'exécute !

Partenaire >Je ferme le placard

Trace d'utilisation de Sampras (avril 2006)

Partenaire > Bonjour, tu vas me dire comment réaliser un flan

Partenaire > Je t'écoute...

Apprenants > où est le sucre?

Partenaire > le sucre est dans le placard n°1

Apprenants > le ouvrir

Partenaire > Je ne te comprends pas lorsque tu écris "le ouvrir". Que veux-tu ouvrir ?

Apprenants > pouvez vous ouvrir le placard à gauche du bas?

Partenaire > je ne te comprends pas lorsque tu écris pouvez vous ouvrir le placard a gauche du bas .

Apprenants > ouvrir le placard du bas à gauche

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvrir le placard du haut à gauche

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > où est le sucre vanille

Partenaire > le sucre est dans le placard n°1

Apprenants > où est la farine

Partenaire > la farine est dans le placard n°3

Apprenants > ouvrir le placard du bas à droite

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvrir le placard du haut à droite

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > où est le chocolat

Partenaire > je ne comprends pas ta question

Apprenants > où est le chocolat

Partenaire > je ne comprends pas ta question

Apprenants > ouvrir le placard du bas au milieu

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

J'ouvre le placard

Apprenants > ouvrir le placard du haut au milieu

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire >J'ouvre le placard

Apprenants > ouvrir le frigo

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > J'ouvre le frigo

Apprenants > ouvrir le tiroir

Partenaire > Je ne te comprends pas lorsque tu écris "ouvrir le tiroir". Quel placard veux-tu ouvrir ?

Apprenants > où est le beurre

Partenaire > le beurre est dans le frigo

Apprenants > prendre la farine

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je prends la farine dans le frigo

Apprenants > prendre le lait

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je sors le lait du frigo

Apprenants > prendre le sucre

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je prends le sucre dans le placard

Apprenants > prendre la crême

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je prends la crème dans le frigo

Apprenants > prendre les oeufs

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je sors les oeufs du frigo

Apprenants > prendre le chocolat

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je prends le chocolat dans le placard

Apprenants > prendre le poelle

Partenaire > Je ne te comprends pas lorsque tu écris "prendre le poelle". Que veux-tu prendre ?

Apprenants > prendre le pot

Partenaire > Je ne te comprends pas lorsque tu écris "prendre le pot". Quel pot veux-tu prendre ?

Apprenants > prendre le saladier

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je sors le saladier du placard

Apprenants > prendre le moule

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je sors le moule du placard

Apprenants > mettre de la farine dans le saladier

Partenaire > J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > ouvrir la boite a les oeufs

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > J'ouvre la boite a oeufs

Apprenants > extirper les oeufs

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je prends les oeufs dans la boîte à oeufs

Apprenants > mettre les oeufs dans le saladier

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je mets les oeufs dans le saladier

Apprenants > mettre de la farine dans le saladier

Partenaire > J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > mettre du sucre dans le saladier

Partenaire > Je ne te comprends pas lorsque tu écris "mettre du sucre dans le saladier". Que veux-tu mettre dans le saladier ?

Apprenants > le sucre

Partenaire > Je ne te comprends pas lorsque tu écris "le sucre ". Que veux-tu mettre dans le saladier ?

Apprenants > mettre le sucre dans le saladier

Partenaire > Je ne te comprends pas lorsque tu écris "mettre le sucre dans le saladier". Que veux-tu mettre dans le saladier ?

Apprenants > incorporer le sucre

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > J'incorpore le sucre

Apprenants > incorporer la farine

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > J'incorpore la farine

Apprenants > incorporer le lait

Partenaire > Je ne te comprends pas lorsque tu écris "incorporer le lait". Que veux-tu incorporer ?

Apprenants > verser le lait

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je verse le lait

Apprenants > melanger

Partenaire > Je ne te comprends pas lorsque tu écris "melanger". Que veux-tu melanger ?

Apprenants > melanger le flan

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je mélange le tout

Apprenants > verser le flan dans le moule

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je verse le flan dans le moule

Apprenants > verser le flan dans le moule

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je verse le flan dans le moule

Apprenants > mettre le moule dans le four

Partenaire > J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > ouvrir le four

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > J'ouvre le four

Apprenants > mettre le moule dans le four

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je mets le moule dans le four

Apprenants > fermer le four

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je ferme le four

Apprenants > cuire

Partenaire > Je ne te comprends pas lorsque tu écris "cuire". Que veux-tu cuire ?

Apprenants > cuire le flan

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je fais cuire le flan

Apprenants > sortir le flan

Partenaire > J'ai bien compris ton ordre, mais je ne peux pas l'exécuter !

Apprenants > ouvrir le four

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > J'ouvre le four

Apprenants > sortir le flan

Partenaire > J'ai bien compris ton ordre, et je l'exécute !

Partenaire > Je sors le moule du four et voilà le flan est prêt!!

Annexe B – Implémentation de Sampras

Cette annexe présente quelques éléments représentatifs du code de Sampras. Nous présentons les modules d'analyse et de comparaison des énoncés avec la tâche.

Module d'analyse

```
;; ============================================================
;; Analyseur sémantique non-déterministe hypothético-déductif
;; Jérôme Lehuen, Johan Michel
;; ============================================================

(defmodule ANALYSEUR )
(defmodule ANALYSEUR-LANGAGE )
(defmodule ANALYSEUR-PHASE-1 )
(defmodule ANALYSEUR-PHASE-11)
(defmodule ANALYSEUR-PHASE-2 )
(defmodule ANALYSEUR-PHASE-3 )


(deftemplate ANALYSEUR::phrase "Les unités lexicales"

      (multislot listemots))                ; Une liste de mots

(deffunction ANALYSEUR::membre-liste (?res $?listeres)
      (if (member$ ?res $?listeres)
            then (return TRUE)
            else (return FALSE)))

(defrule ANALYSEUR::init
      (declare (salience 999))
      (enonce (texte ?texte))

      =>
      (list-focus-stack)
      (focus ANALYSEUR-PHASE-3)
      (list-focus-stack)
```

```
      (focus ANALYSEUR-PHASE-2)
      (list-focus-stack)
      (focus ANALYSEUR-PHASE-11)
      (focus ANALYSEUR-PHASE-1)
      (list-focus-stack))

;;  ---------------------------------------------------------------------------
---------
;; Quelques fonctions utilitaires
;;  ---------------------------------------------------------------------------
---------

(deffunction ANALYSEUR::vide$ ($?liste) (eq 0 (length$ $?liste)))
(deffunction ANALYSEUR::car$ ($?liste) (nth$ 1 $?liste))
(deffunction ANALYSEUR::cdr$ ($?liste) (rest$ $?liste))

;;  ---------------------------------------------------------------------------
---------
;; Le modèle de langage
;;  ---------------------------------------------------------------------------
---------

(deftemplate ANALYSEUR-LANGAGE::lexie "Les unités lexicales"
      (slot      categorie)         ; Une catégorie lexicale
      (multislot vocable)           ; Une liste de mots
      (multislot syntaxe))          ; Une liste de mots et/ou de catégories

(deftemplate ANALYSEUR-LANGAGE::granule "Les granules de compréhension"

      (multislot texte)                ; Une liste de vocables
      (multislot granules)             ; Les FACT-ADDRESS des sous-granules
      (slot      categorie)            ; Une categorie conceptuelle
      (slot      hypotheses (default 0))  ; Un nombre d'hypothèses (une par
mot)
      (slot      position)             ; La position dans la phrase
      (slot      taille)               ; Le nombre de mots dans la phrase
      (slot      lex (default FALSE))) ; TRUE si le granule est lexical
```

```
(deftemplate ANALYSEUR-LANGAGE::agglu "Les groupes de granules syntaxiques"

      (multislot granules)            ; Liste des FACT-ADDRESS des granules
      (multislot syntaxe)
      (slot      hypotheses)
      (slot      position)
      (slot      taille))

(deftemplate ANALYSEUR-LANGAGE::result "Les résultats de l'analyse"

      (slot             granule)    ; la granule qui lui donne naissance
      (multislot granules)         ; Liste des FACT-ADDRESS des granules
      (slot      position)         ; La position dans la phrase
      (slot      couverture)       ; Le nombre de mots dans la phrase
      (slot      hypotheses))      ; Le nombre d'hypothèses

(deftemplate ANALYSEUR-LANGAGE::listeresultats "la liste des résultats"

      (multislot resultats))

(deftemplate ANALYSEUR-LANGAGE::listegranulestexte   "la liste des granules
texte"

      (multislot granules))

(deftemplate ANALYSEUR-LANGAGE::analyse1
      (multislot resultat))

(deftemplate ANALYSEUR-LANGAGE::compartache
      (multislot resultat))

(deftemplate ANALYSEUR-LANGAGE::compartachemem
      (multislot resultat))

(deftemplate ANALYSEUR-LANGAGE::analysepbl
      (multislot resultat))

(deftemplate ANALYSEUR-LANGAGE::analysepblincomplet
      (multislot resultat))
```

```
(deftemplate ANALYSEUR-LANGAGE::analysemsbc
      (multislot resultat))

;; ---------------------------------------------------------------------------
;; Quelques fonctions sur le modèle de langage
;; ---------------------------------------------------------------------------

(deffunction ANALYSEUR-LANGAGE::affiche-granule (?granule)

      (printout t
            ?granule " "
            (fact-slot-value ?granule categorie) " "
            (fact-slot-value ?granule texte) " "
            (fact-slot-value ?granule position) "-"
            (fact-slot-value ?granule taille) " ["
            (fact-slot-value ?granule hypotheses) "]" crlf))

(deffunction ANALYSEUR-LANGAGE::affiche-resultat (?result)

      (bind $?granules (fact-slot-value ?result granules))
      (printout t
            "-------------------------------------------------------" crlf
            "Résultat: " $?granules " "
            (fact-slot-value ?result position) "-"
            (fact-slot-value ?result couverture) " ["
            (fact-slot-value ?result hypotheses) "] " ?result crlf)
      (foreach ?gr $?granules (ANALYSEUR-LANGAGE::affiche-granule ?gr)))

(deffunction ANALYSEUR-LANGAGE::conflictuels (?resultat1 ?resultat2)
      (bind ?pos1 (fact-slot-value ?resultat1 position))
      (bind ?pos2 (fact-slot-value ?resultat2 position))
      (bind ?len1 (fact-slot-value ?resultat1 couverture))
      (bind ?len2 (fact-slot-value ?resultat2 couverture))

      (or (and (>= ?pos1 ?pos2) (< ?pos1 (+ ?pos2 ?len2)))
          (and (>= ?pos2 ?pos1) (< ?pos2 (+ ?pos1 ?len1)))))

;; ---------------------------------------------------------------------------
```

```
---------
;; PHASE-1 / Générer des granules de compréhension
;; ---------------------------------------------------------------------------

(defrule ANALYSEUR-PHASE-1::vide
      (declare (salience 40))
      ?p <- (enonce (texte ""))
      =>
      (retract ?p)
      (assert (MAIN::analyse (resultat vide))))

(defrule ANALYSEUR-PHASE-1::fermer-dialogue
      (declare (salience 40))
      ?p <- (enonce (texte "au revoir"))
      =>
      (retract ?p)
      (assert (MAIN::analyse (resultat clore-dialogue))))

(defrule ANALYSEUR-PHASE-1::init

      (declare (salience 999))
      (MAIN::enonce (texte ?texte))
      =>
      (bind $?liste (explode$ ?texte))
      (assert (ANALYSEUR::phrase (listemots $?liste))))

(defrule ANALYSEUR-PHASE-1::cree-liste-des-granules-texte

      (declare (salience 35))
      (not (ANALYSEUR-LANGAGE::listegranulestexte))
      =>
      (assert (ANALYSEUR-LANGAGE::listegranulestexte (granules))))
```

```
(defrule ANALYSEUR-PHASE-1::cree-granule-lexical

      (declare (salience 30))
      (ANALYSEUR::phrase (listemots $?avant ?mot $?arriere))
      =>
      (assert (ANALYSEUR-LANGAGE::granule (texte ?mot)
                        (categorie ?mot)
                        (position (length$ $?avant))
                        (taille 1)
                        (lex TRUE))))

(defrule ANALYSEUR-PHASE-1::cree-granule

      (declare (salience 20))
      (ANALYSEUR-LANGAGE::lexie (vocable $?texte) (categorie ?cat&~nil))
      (ANALYSEUR::phrase (listemots $?avant $?texte $?))
      =>
      (assert (ANALYSEUR-LANGAGE::granule (texte $?texte)
                        (categorie ?cat)
                        (position (length$ $?avant))
                        (taille (length$ $?texte)))))

(defrule ANALYSEUR-PHASE-1::ajouter-liste-granulestexte

      (declare (salience 15))
      ?liste     <-     (ANALYSEUR-LANGAGE::listegranulestexte     (granules
$?listegranulestexte))
      ?granule <- (ANALYSEUR-LANGAGE::granule (lex FALSE))
      (test(not(ANALYSEUR::membre-liste ?granule $?listegranulestexte)))
      =>
      (retract ?liste)
      (bind $?granules (create$ $?listegranulestexte ?granule))
      (assert        (ANALYSEUR-LANGAGE::listegranulestexte        (granules
$?granules)))
)
```

```
;;  ----------------------------------------------------------------------
---------
;; PHASE-1 / Générer des granules de compréhension hypothétiques
;;  ----------------------------------------------------------------------
---------


(defrule ANALYSEUR-PHASE-11::cree-granule-hypothetique-gauche


      (declare (salience 10))
      (ANALYSEUR-LANGAGE::granule  (texte  $?apres)  (categorie  ?cat-apres)
(hypotheses 0))
      (ANALYSEUR-LANGAGE::lexie (syntaxe ?cat ?cat-apres $?))
      (ANALYSEUR::phrase (listemots $?texte $?apres $?))
      (categories $? ?cat $?)
      ;; Vérifier que ?texte n'est pas vide
      (test (> (length$ ?texte) 0))
      ;; Vérifier que le granule n'a pas déjà été généré
      (not (ANALYSEUR-LANGAGE::granule (texte $?texte) (categorie ?cat)))
      =>
      (assert (ANALYSEUR-LANGAGE::granule (texte ?texte)
                       (categorie ?cat)
                       (hypotheses (length$ ?texte))  ; 1
                       (position 0)
                       (taille (length$ ?texte)))))


(defrule ANALYSEUR-PHASE-11::cree-granule-hypothetique-droite


      (declare (salience 10))
      (ANALYSEUR-LANGAGE::granule  (texte  $?avant)  (categorie  ?cat-avant)
(hypotheses 0))
      (ANALYSEUR-LANGAGE::lexie (syntaxe $? ?cat-avant ?cat))
      (ANALYSEUR::phrase (listemots $?toto $?avant $?texte))
      (categories $? ?cat $?)
      ;; Vérifier que ?texte n'est pas vide
      (test (> (length$ ?texte) 0))
      ;; Vérifier que le granule n'a pas déjà été généré
      (not (ANALYSEUR-LANGAGE::granule (texte $?texte) (categorie ?cat)))
      =>
      (assert (ANALYSEUR-LANGAGE::granule (texte ?texte)
                       (categorie ?cat)
```

```
                        (hypotheses (length$ ?texte))
                        (position (+ (length$ ?toto) (length$ ?avant)))
                        (taille (length$ ?texte)))))


(defrule ANALYSEUR-PHASE-11::cree-granule-hypothetique-milieu


      (declare (salience 10))
      (ANALYSEUR-LANGAGE::granule  (texte  $?avant)  (categorie  ?cat-avant)
(hypotheses 0))
      (ANALYSEUR-LANGAGE::granule  (texte  $?apres)  (categorie  ?cat-apres)
(hypotheses 0))
      (ANALYSEUR-LANGAGE::lexie (syntaxe $? ?cat-avant ?cat ?cat-apres $?))
      (ANALYSEUR::phrase (listemots $?toto $?avant $?texte $?apres $?))
      ;; Vérifier que ?texte n'est pas vide
      (test (> (length$ ?texte) 0))
      ;; Vérifier que le granule n'a pas déjà été généré
      (not (ANALYSEUR-LANGAGE::granule (texte $?texte) (categorie ?cat)))


      =>
      (assert (ANALYSEUR-LANGAGE::granule (texte ?texte)
                        (categorie ?cat)
                        (hypotheses (length$ ?texte))
                        (position (+ (length$ ?toto) (length$ ?avant)))
                        (taille (length$ ?texte)))))
```

```
;; ---------------------------------------------------------------------
---------
;; PHASE-1 / Agglutiner les granules de compréhension
;; ---------------------------------------------------------------------
---------


(defrule ANALYSEUR-PHASE-11::cree-agglu-initier


      (declare (salience 6))
      ?gr1  <-  (ANALYSEUR-LANGAGE::granule  (categorie  ?cat1)  (position
?pos1) (taille ?len1) (hypotheses ?hyp1))
      ?gr2  <-  (ANALYSEUR-LANGAGE::granule  (categorie  ?cat2)  (position
?pos2) (taille ?len2) (hypotheses ?hyp2))


      (test (= ?pos2 (+ ?pos1 ?len1)))


      =>
      ;(printout t "on init une agglu" ?gr1 ?gr2 crlf)
      (assert (ANALYSEUR-LANGAGE::agglu (granules ?gr1 ?gr2)
                     (syntaxe ?cat1 ?cat2)
                     (hypotheses (+ ?hyp1 ?hyp2))
                     (position ?pos1)
                     (taille (+ ?len1 ?len2)))))


(defrule ANALYSEUR-PHASE-11::cree-agglu-continuer


      (declare (salience 6))
      (ANALYSEUR-LANGAGE::agglu (granules $?gra) (syntaxe $?syn) (position
?pos1) (taille ?len1) (hypotheses ?hyp1))
      ?gr2  <-  (ANALYSEUR-LANGAGE::granule  (categorie  ?cat2)  (position
?pos2) (taille ?len2) (hypotheses ?hyp2))


      (test (= ?pos2 (+ ?pos1 ?len1)))
      =>
      (assert (ANALYSEUR-LANGAGE::agglu (granules $?gra ?gr2)
                     (syntaxe $?syn ?cat2)
                     (hypotheses (+ ?hyp1 ?hyp2))
                     (position ?pos1)
                     (taille (+ ?len1 ?len2)))))
```

```
;; ---------------------------------------------------------------------------
---------
;; PHASE-1 / Générer des granules de haut niveau
;; ---------------------------------------------------------------------------
---------


(defrule ANALYSEUR-PHASE-11::cree-super-granule


      (declare (salience 40))
      ?agglu <- (ANALYSEUR-LANGAGE::agglu (granules $?granules) (syntaxe
$?texte) (hypotheses ?hyp) (position ?pos) (taille ?len))
      (ANALYSEUR-LANGAGE::lexie (syntaxe $?texte) (categorie ?cat))
      =>
      (assert (ANALYSEUR-LANGAGE::granule (texte $?texte)
                        (granules $?granules)
                        (categorie ?cat)
                        (hypotheses ?hyp)
                        (position ?pos)
                  (taille ?len))))
```

```
;; ---------------------------------------------------------------------------
;; PHASE-2 / Construire les faits résultat
;; ---------------------------------------------------------------------------


(deffunction ANALYSEUR-PHASE-2::mapcar$ (?liste ?fonc)
(if (ANALYSEUR::vide$ $?liste)
      then return (create$)
      else return (create$ (apply ?fonc (nth$ 1 $?liste)) (ANALYSEUR-PHASE-2::mapcar$ (rest$ $?liste) ?fonc))))


(deffunction ANALYSEUR-PHASE-2::cree-liste-resultat (?granule)
      (bind $?granules (fact-slot-value ?granule granules))
      (if (ANALYSEUR::vide$ $?granules)
            then (if (not (fact-slot-value ?granule lex)) then return (create$ $?granule) else return (create$))
            else return (create$ $?granule (ANALYSEUR-PHASE-2::mapcar$ $?granules ANALYSEUR-PHASE-2::cree-liste-resultat))))


(defrule ANALYSEUR-PHASE-2::cree-resultat


      (declare (salience 35))
      ?granule <- (ANALYSEUR-LANGAGE::granule (lex FALSE) (position ?pos) (taille ?len) (hypotheses ?hyp))
      =>
      (assert (ANALYSEUR-LANGAGE::result (granule ?granule) (granules (ANALYSEUR-PHASE-2::cree-liste-resultat ?granule))
                        (position ?pos)
                        (couverture ?len)
                        (hypotheses ?hyp)))
)
```

```
;; ---------------------------------------------------------------------------
;; PHASE-2 / Heuristiques de résolution de conflits entre résultats
;; ---------------------------------------------------------------------------

(defrule ANALYSEUR-PHASE-2::heuristique-conflit-1

      "A couvertures égales, on garde les résultats de degré de certitude maximal"

      (declare (salience 25))
      ?res1 <- (ANALYSEUR-LANGAGE::result (couverture ?len) (hypotheses ?hyp1))
      ?res2 <- (ANALYSEUR-LANGAGE::result (couverture ?len) (hypotheses ?hyp2&:(> ?hyp1 ?hyp2)))
      (test (ANALYSEUR-LANGAGE::conflictuels ?res1 ?res2))

      =>
      (retract ?res1))

(defrule ANALYSEUR-PHASE-2::heuristique-conflit-2

      "A degrés de certitude égaux, on garde les résultats de couverture maximale"

      (declare (salience 25))
      ?res1 <- (ANALYSEUR-LANGAGE::result (hypotheses ?hyp) (couverture ?len1))
      ?res2 <- (ANALYSEUR-LANGAGE::result (hypotheses ?hyp) (couverture ?len2&:(> ?len1 ?len2)))
      (test (ANALYSEUR-LANGAGE::conflictuels ?res1 ?res2))

      =>
      (retract ?res2))
```

```
(defrule ANALYSEUR-PHASE-2::heuristique-meilleure-proba
      "le gain en couverture est supérieur au risque ajouté"

      (declare (salience 25))
      ?res1 <- (ANALYSEUR-LANGAGE::result (hypotheses ?hyp1) (couverture
?len1))
      ?res2 <- (ANALYSEUR-LANGAGE::result (hypotheses ?hyp2) (couverture
?len2&:(> ?len1 ?len2)))
      (test (and (> ( - ?len1 ?len2) (- ?hyp1 ?hyp2)) (ANALYSEUR-
LANGAGE::conflictuels ?res1 ?res2)))

      =>
      ;(printout t "Heuristique n°3: suppression de " ?res2 " au profit de
" ?res1 crlf)
      (retract ?res2))

(defrule ANALYSEUR-PHASE-2::heuristique-result-idem

      "Deux résultats sont identiques"

      (declare (salience 25))
      ?res1 <- (ANALYSEUR-LANGAGE::result (granules $?granules1)
(hypotheses ?hyp1) (position ?pos1) (couverture ?len1))
      ?res2 <- (ANALYSEUR-LANGAGE::result (granules $?granules2)
(hypotheses ?hyp2) (position ?pos2) (couverture ?len2))

      (test (and (neq ?res1 ?res2) (= ?hyp1 ?hyp2) (= ?len1 ?len2)(= ?pos1
?pos2)))

      =>
      (retract ?res2))
```

```
;;  ---------------------------------------------------------------------
---------
;; PHASE-2 / Affichage et nettoyage
;;  ---------------------------------------------------------------------
---------

(defrule ANALYSEUR-PHASE-2::cree-liste-des-resultats

      (declare (salience 10))
      (not (ANALYSEUR-LANGAGE::listeresultats))
      =>
      (assert (ANALYSEUR-LANGAGE::listeresultats (resultats))))

(defrule ANALYSEUR-PHASE-2::ajouter-liste-resultats

      (declare (salience 15))
      ?res <- (ANALYSEUR-LANGAGE::result)
      ?liste <- (ANALYSEUR-LANGAGE::listeresultats (resultats $?listeres))

      (test(not(ANALYSEUR::membre-liste ?res $?listeres)))
      =>
      (retract ?liste)
      (bind $?resultats (create$ $?listeres ?res))
      (assert (ANALYSEUR-LANGAGE::listeresultats (resultats $?resultats)))
)

(defrule ANALYSEUR-PHASE-2::affiche-resultat-avant

      (declare (salience 20))

      ?res <- (ANALYSEUR-LANGAGE::result)
      =>
      (ANALYSEUR-LANGAGE::affiche-resultat ?res))

(defrule ANALYSEUR-PHASE-2::affiche-resultat-apres

      (declare (salience 1))
      ?res <- (ANALYSEUR-LANGAGE::result)
```

```
      =>
      (ANALYSEUR-LANGAGE::affiche-resultat ?res))


;;  ---------------------------------------------------------------------------
---------
;; PHASE-3 / Ce qui ressort de l'analyse et nettoyage
;;  ---------------------------------------------------------------------------
---------


(deffunction ANALYSEUR-PHASE-3::couverturegtexte (?l $?listegranules)
      (bind ?couv 0)
      (printout t "Qui m'appelle ?")
      (foreach ?granule $?listegranules
            (bind ?couv (+ ?couv (fact-slot-value ?granule taille))))
      (printout t "la longueur" ?l "la couv" ?couv)
      (return (eq ?l ?couv)))

(defrule ANALYSEUR-PHASE-3::bonne-couverture

      (declare (salience 20))
      (ANALYSEUR-LANGAGE::listegranulestexte (granules $?granules))
      ?p <- (ANALYSEUR::phrase (listemots $?mots))
      (test    (ANALYSEUR-PHASE-3::couverturegtexte    (length$    $?mots)
$?granules))

      =>
      (assert (ANALYSEUR-PHASE-3::couvbonne))
)

(defrule ANALYSEUR-PHASE-3::mauvaise-couverture

      (declare (salience 20))
      (ANALYSEUR-LANGAGE::listegranulestexte (granules $?granules))
      ?p <- (ANALYSEUR::phrase (listemots $?mots))

      (not(test   (ANALYSEUR-PHASE-3::couverturegtexte   (length$   $?mots)
$?granules)))

      =>    (printout t (length$ $?mots))
      (assert (ANALYSEUR-PHASE-3::couvmauvaise))
      (focus MAUVAISECOUVERTURE)
```

```
      )


(deffunction ANALYSEUR-PHASE-3::bonling (?res $?mots)
      (bind ?p (fact-slot-value ?res position))
      (bind ?c (fact-slot-value ?res couverture))
      (bind ?h (fact-slot-value ?res hypotheses))
      (bind ?l (length$ $?mots))
      (and (eq 0 ?h) (and (eq 0 ?p) (eq ?l ?c))))

(defrule ANALYSEUR-PHASE-3::bonne-syntaxe-bonne-couverture

      (declare (salience 20))

      ?r <- (ANALYSEUR-PHASE-3::couvbonne)
      ?p <- (ANALYSEUR::phrase (listemots $?liste))
      ?res <- (ANALYSEUR-LANGAGE::result)
      (test (ANALYSEUR-PHASE-3::bonling ?res $?liste))
      =>
      (printout t "on trouve une bonne couverture et une bonne syntaxe"
crlf)
      (retract ?r)
      (bind $?granules (fact-slot-value ?res granules))
      (assert (ANALYSEUR-LANGAGE::compartache (resultat $?granules)))
      (printout t "on trouve le bon res" crlf)
      (focus COMPARAISONTACHE)
      )

(defrule ANALYSEUR-PHASE-3::mauvaise-syntaxe-bonne-couverture

      (declare (salience 20))
      ?r <- (ANALYSEUR-PHASE-3::couvbonne)
      ?p <- (ANALYSEUR::phrase (listemots $?liste))
      ?res <- (ANALYSEUR-LANGAGE::result (granules $?granules))
      (not(test (ANALYSEUR-PHASE-3::bonling ?res $?liste)))
      =>
      (printout t "on trouve une bonne couverture mais une mauvaise
syntaxe" crlf);
      (retract ?r)
      (assert (ANALYSEUR-LANGAGE::analysemsbc (resultat $?granules)))
      (focus SEMANTIQUE)
```

```
      )



(defrule ANALYSEUR-PHASE-3::question-quoi
      (declare (salience 20))
      ?p <- (enonce (texte "Que faire ?"))

      =>
      (retract ?p)
      (assert (ANALYSEUR-LANGAGE::analyse1 (resultat que-faire))))
;;  ----------------------------------------------------------------------
---------
```

Module de comparaison entre l'énoncé et l'arbre des tâches

```
;; ============================================================
;; Comparaison des résultats de l'analyse avec l'arbre des tâches
;; Johan Michel -
;; ============================================================

(defmodule COMPARAISONTACHE)

;; compare deux chaines de caractères et renvoie vrai si les éléments de
l'une sont dans l'autre (sans notion d'ordre)
;; ex: (member-string "action" "action lapin") renvoie FALSE)
;; ex: (member-string "action lapin" "action") renvoie TRUE)

(deffunction COMPARAISONTACHE::member-string (?ch1 ?ch2)
      (bind $?liste1 (explode$ ?ch1))
      (bind $?liste2 (explode$ ?ch2))
      (return (subsetp $?liste2 $?liste1)))

;; transforme les granules en une tâche résultat qui sera à comparer avec
celles définies

(deffunction COMPARAISONTACHE::transformation ($?granules)
      (bind ?chaine "")
      (foreach ?gr $?granules
            (bind ?cat (fact-slot-value ?gr categorie))
            (bind ?chaine (str-cat ?chaine (str-cat "(" ?cat)))
            (bind ?tex (fact-slot-value ?gr texte))
            (bind ?chaine (str-cat ?chaine (str-cat " (" (str-cat ?tex
"))")))))
      (assert (ANALYSEUR-LANGAGE::tacheres ?chaine)))

;;    préparation à la transformation précédente

(defrule COMPARAISONTACHE::transfobonres

      (declare (salience 20))
      ?p <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      =>
```

```
      (COMPARAISONTACHE::transformation $?granules))


(defrule COMPARAISONTACHE::transfobonres-mem


      (declare (salience 20))
      ?p <- (ANALYSEUR-LANGAGE::compartachemem (resultat $?granules))
      =>
      (COMPARAISONTACHE::transformation $?granules))



(defrule COMPARAISONTACHE::question


      (declare (salience 11))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p  <-  (ANALYSEUR-LANGAGE::tacheres  ?des)    ;;([question]  (ou  est
[ingredient]))([ingredient] (le sucre))
      (test(COMPARAISONTACHE::member-string ?des "[question]"))


      =>
      (printout t "on a une question")
      (retract ?r ?p)
      (assert (analyse (resultat question-ou ?des)))
      (focus QUESTION))


;;    cas où il y a correspondance entre la tâche identifiée et les tâches
existantes


(defrule COMPARAISONTACHE::correspondancearbretache


      (declare (salience 10))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(COMPARAISONTACHE::member-string ?des ?def))
      =>
      (retract ?p ?r)
      (assert (analyse (resultat tache ?ident)))
      (focus NETTOYAGE)
      (focus NETTOYAGE-MEMOIRE)
      )
```

```
;     cas où après une balade dans la mémoire on revient avec une analyse qui nous rend une tâche


(defrule COMPARAISONTACHE::correspondancearbretache-mem1


      (declare (salience 11))
      ?r <- (ANALYSEUR-LANGAGE::compartachemem (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      ?f <- (ANALYSEUR-LANGAGE::analysepblincomplet)
      (test(COMPARAISONTACHE::member-string ?des ?def))
      =>
      (retract ?p ?r ?f)
      (assert (analyse (resultat tache ?ident)))
      (focus NETTOYAGE)
      (focus NETTOYAGE-MEMOIRE)
      )


(defrule COMPARAISONTACHE::correspondancearbretache-mem2


      (declare (salience 11))
      ?r <- (ANALYSEUR-LANGAGE::compartachemem (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      ?f <- (ANALYSEUR-LANGAGE::analysepbl)
      (test(COMPARAISONTACHE::member-string ?des ?def))
      =>
      (retract ?p ?r ?f)
      (assert (analyse (resultat tache ?ident)))
      (focus NETTOYAGE)
      (focus NETTOYAGE-MEMOIRE)
      )


; on a une granule en mémoire que l'on va essayer de matcher avec le résultat qui sera transformé en granule mémoire ex: "ouvrir " puis "le frigo"
(defrule    COMPARAISONTACHE::noncorrespondancearbretache-2-ordreincomplet-mem-mem
      (declare (salience 8))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
```

```
      ?m <- (MEMOIRE::granule-mem)
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(not(COMPARAISONTACHE::member-string ?des ?def)))
      (test(COMPARAISONTACHE::member-string ?def ?des))
      =>
      (retract ?p ?r)
      (assert     (ANALYSEUR-LANGAGE::analysepblincomplet         (resultat
$?granules)))
      (focus MEMOIRE))

;;    cas où il n'y a pas correspondance entre la tâche identifiée et les
tâches existantes

; "ouvrir", correspondance partielle.
(defrule COMPARAISONTACHE::noncorrespondancearbretache-2-ordreincomplet

      (declare (salience 7))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(not(COMPARAISONTACHE::member-string ?des ?def)))
      (test(COMPARAISONTACHE::member-string ?def ?des))

      =>
      (retract ?p ?r)
      (assert     (ANALYSEUR-LANGAGE::analysepblincomplet         (resultat
$?granules)))
      (focus INCIDENCE))

; cas où à partie d'une liste mémoire donnant une tâche incomplète, on doit
trouver la question à poser

(defrule COMPARAISONTACHE::noncorrespondancearbretache-2-ordreincomplet-mem
      (declare (salience 7))
      ?r <- (ANALYSEUR-LANGAGE::compartachemem (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      ?m <- (MEMOIRE::granule-mem)
      ?l <- (MEMOIRE::listeresultats-mem (resultats $?resultats))
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(not(COMPARAISONTACHE::member-string ?des ?def)))
      (test(COMPARAISONTACHE::member-string ?def ?des))
```

```
      =>
      (retract ?p ?r)
      (assert     (ANALYSEUR-LANGAGE::analysepblincomplet         (resultat
$?granules)))
      (focus INCIDENCE-MEMOIRE))

(defrule COMPARAISONTACHE::noncorrespondancearbretache-2-mem
      (declare (salience 7))
      ?r <- (ANALYSEUR-LANGAGE::compartachemem (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      ?m <- (MEMOIRE::granule-mem)
      ?l <- (MEMOIRE::listeresultats-mem (resultats $?resultats))
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(not(COMPARAISONTACHE::member-string ?des ?def)))
      (test(not(COMPARAISONTACHE::member-string ?def ?des)))
      =>
      (retract ?p ?r)
      (assert     (ANALYSEUR-LANGAGE::analysepblincomplet         (resultat
$?granules)))
      (focus INCIDENCE))

(defrule COMPARAISONTACHE::noncorrespondancearbretache-1-mem

      (declare (salience 6))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      ?m <- (MEMOIRE::granule-mem)
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(not(COMPARAISONTACHE::member-string ?des ?def)))
      (test(not(COMPARAISONTACHE::member-string ?def ?des)))
      =>
      (assert (ANALYSEUR-LANGAGE::analysepbl  (resultat $?granules)))
      (focus INCIDENCE))
```

```
(defrule COMPARAISONTACHE::noncorrespondancearbretache-1

      (declare (salience 5))
      ?r <- (ANALYSEUR-LANGAGE::compartache (resultat $?granules))
      ?p <- (ANALYSEUR-LANGAGE::tacheres ?des)
      (MAIN::tache (ident ?ident) (definition ?def))
      (test(not(COMPARAISONTACHE::member-string ?des ?def)))
      (test(not(COMPARAISONTACHE::member-string ?def ?des)))

      =>
      (retract ?p ?r)
      ;(printout t "noncorrespondancearbretache-1")
      (assert (ANALYSEUR-LANGAGE::analysepbl  (resultat $?granules)))
      (focus INCIDENCE))
      (focus INCIDENCE-MEMOIRE))
```

Annexe C – Lexique de Sampras

Cette annexe présente le lexique de Sampras. Il comprend une soixantaine de mots répartis selon quatorze catégories lexicales. Ce lexique est un fichier XML.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lexique SYSTEM "lexique.dtd">

<lexique>

<!--les endroits-->

<lexies cat="endroit">
	<lexie voc="la cuisine"/>
</lexies>

<!-- les plats -->

<lexies cat="recette">
	<lexie voc="le flan"/>
	<lexie voc="le flan au chocolat"/>
	<lexie voc="la preparation"/>
</lexies>

<!-- les ingredients -->

<lexies cat="ingredient">
	<lexie voc="les ingredients"/>
	<lexie voc="le sucre"/>
	<lexie voc="le sucre vanille"/>
	<lexie voc="la farine"/>
	<lexie voc="le chocolat"/>
	<lexie voc="les oeufs"/>
	<lexie voc="le beurre"/>
	<lexie voc="le lait"/>
	<lexie voc="la creme"/>
</lexies>

<!-- les recipients -->

<lexies cat="recipient">
	<lexie voc="le saladier"/>
	<lexie voc="le moule"/>
	<lexie voc="le plat"/>
	<lexie voc="la casserole"/>
</lexies>
```

```
<!-- les appareils electro-menagers -->

<lexies cat="electromen">
	<lexie voc="le four"/>
</lexies>

<!-- les contenants -->

<lexies cat="contenant">
	<lexie voc="la boite">
		<syntaxe>la boite</syntaxe>
		<syntaxe>la boite<expr cat="contenu"/></syntaxe>
	</lexie>
	<lexie voc="la bouteille"/>
	<lexie voc="le pot">
		<syntaxe>le pot</syntaxe>
		<syntaxe>le pot<expr cat="contenu"/></syntaxe>
	</lexie>
</lexies>

<!-- les contenus -->

<lexies cat="contenu">
	<lexie voc="a">
		<syntaxe>a</syntaxe>
		<syntaxe>a<expr cat="ingredient"/></syntaxe>
	</lexie>
	<lexie voc="de">
		<syntaxe>de</syntaxe>
		<syntaxe>de<expr cat="ingredient"/></syntaxe>
	</lexie>
</lexies>

<!-- les ustensiles -->

<lexies cat="ustensile">
	<lexie voc="le fouet"/>
	<lexie voc="la cuillere"/>
	<lexie voc="la balance"/>
	<lexie voc="le doseur"/>
	<lexie voc="les ustensiles"/>
</lexies>

<!-- les rangements -->

<lexies cat="rangement">
	<lexie voc="le placard">
		<syntaxe>le placard</syntaxe>
		<syntaxe>le placard<expr cat="attribut"/></syntaxe>
	</lexie>
```

```
		<lexie voc="le tiroir"/>
		<lexie voc="le frigo"/>
</lexies>

<lexies cat="rangement-plat">
		<lexie voc="le plan"/>
		<lexie voc="le plan de travail"/>
</lexies>

<!-- les attributs -->

<lexies cat="attribut">
		<lexie voc="du bas">
			<syntaxe>du bas</syntaxe>
			<syntaxe>du bas<expr cat="attribut"/></syntaxe>
		</lexie>
		<lexie voc="du haut">
			<syntaxe>du haut</syntaxe>
			<syntaxe>du haut<expr cat="attribut"/></syntaxe>
		</lexie>
		<lexie voc="a gauche"/>
		<lexie voc="a droite"/>
		<lexie voc="au milieu"/>
</lexies>

<!-- les localisateurs -->

<lexies cat="localisateur">
		<lexie voc="dans"/>
		<lexie voc="sur"/>
</lexies>

<!-- les actions -->

<lexies cat="action">

		<lexie voc="bouillir">
			<syntaxe>bouillir<expr cat="ingredient"/></syntaxe>
			<syntaxe>bouillir<expr cat="ingredient"/><expr cat="localisateur"/><expr
cat="recipient"/></syntaxe>
		</lexie>

		<lexie voc="cuire">
			<syntaxe>cuire<expr cat="recette"/></syntaxe>
			<syntaxe>cuire<expr cat="ingredient"/></syntaxe>
			<syntaxe>cuire<expr cat="ingredient"/><expr cat="recipient"/><expr
cat="electromen"/></syntaxe>
			<syntaxe>cuire<expr cat="ingredient"/><expr cat="recipient"/><expr
cat="recipient"/></syntaxe>
		</lexie>
```

```
<lexie voc="deposer">
	<syntaxe>deposer<expr cat="ingredient"/></syntaxe>
	<syntaxe>deposer<expr cat="recipient"/></syntaxe>
	<syntaxe>deposer<expr cat="ustensile"/></syntaxe>
	<syntaxe>deposer<expr cat="ingredient"/><expr cat="recipient"/><expr
cat="rangement"/></syntaxe>
	<syntaxe>deposer<expr cat="ingredient"/><expr cat="recipient"/><expr
cat="rangement-plat"/></syntaxe>
</lexie>

<lexie voc="extirper">
	<syntaxe>extirper<expr cat="ingredient"/></syntaxe>
</lexie>

<lexie voc="faire">
	<syntaxe>faire<expr cat="recette"/></syntaxe>
	<syntaxe>faire<expr cat="action"/></syntaxe>
</lexie>

<lexie voc="fermer">
	<syntaxe>fermer<expr cat="rangement"/></syntaxe>
	<syntaxe>fermer<expr cat="contenant"/></syntaxe>
	<syntaxe>fermer<expr cat="electromen"/></syntaxe>
</lexie>

<lexie voc="fondre">
	<syntaxe>fondre<expr cat="ingredient"/></syntaxe>
	<syntaxe>fondre<expr cat="ingredient"/><expr cat="localisateur"/><expr
cat="recipient"/></syntaxe>
</lexie>

<lexie voc="incorporer">
	<syntaxe>incorporer<expr cat="ingredient"/></syntaxe>
	<syntaxe>incorporer<expr cat="ingredient"/>et<expr
cat="ingredient"/></syntaxe>
</lexie>

<lexie voc="melanger">
	<syntaxe>melanger<expr cat="ingredient"/></syntaxe>
	<syntaxe>melanger<expr cat="ingredient"/>et<expr
cat="ingredient"/></syntaxe>
	<syntaxe>melanger<expr cat="ingredient"/>et<expr cat="ingredient"/><expr
cat="localisateur"/><expr cat="recipient"/></syntaxe>
	<syntaxe>melanger<expr cat="recette"/></syntaxe>
</lexie>
```

```
<lexie voc="mettre">
	<syntaxe>mettre<expr cat="ingredient"/><expr cat="localisateur"/><expr
cat="recipient"/></syntaxe>
	<syntaxe>mettre<expr cat="ingredient"/><expr cat="localisateur"/><expr
cat="rangement"/></syntaxe>
	<syntaxe>mettre<expr cat="ustensile"/><expr cat="localisateur"/><expr
cat="recipient"/></syntaxe>
	<syntaxe>mettre<expr cat="ustensile"/><expr cat="localisateur"/><expr
cat="rangement"/></syntaxe>
	<syntaxe>mettre<expr cat="recipient"/><expr cat="localisateur"/><expr
cat="electromen"/></syntaxe>
	<syntaxe>mettre<expr cat="ingredient"/>sur le plan</syntaxe>
	<syntaxe>mettre<expr cat="ingredient"/>sur le plan</syntaxe>
</lexie>

<lexie voc="ouvrir">
	<syntaxe>ouvrir<expr cat="rangement"/></syntaxe>
	<syntaxe>ouvrir<expr cat="electromen"/></syntaxe>
	<syntaxe>ouvrir<expr cat="contenant"/></syntaxe>
</lexie>

<lexie voc="prechauffer">
	<syntaxe>prechauffer<expr cat="electromen"/></syntaxe>
</lexie>

<lexie voc="prendre">
	<syntaxe>prendre<expr cat="ingredient"/></syntaxe>
	<syntaxe>prendre<expr cat="recipient"/></syntaxe>
	<syntaxe>prendre<expr cat="ustensile"/></syntaxe>
	<syntaxe>prendre<expr cat="ingredient"/><expr cat="localisateur"/><expr
cat="rangement"/></syntaxe>
	<syntaxe>prendre<expr cat="recipient"/><expr cat="localisateur"/><expr
cat="rangement"/></syntaxe>
	<syntaxe>prendre<expr cat="ustensile"/><expr cat="localisateur"/><expr
cat="rangement"/></syntaxe>
	<syntaxe>prendre<expr cat="contenant"/>a<expr
cat="ingredient"/></syntaxe>
	<syntaxe>prendre<expr cat="contenant"/>de<expr
cat="ingredient"/></syntaxe>
</lexie>

<lexie voc="preparer">
	<syntaxe>preparer<expr cat="recette"/></syntaxe>
</lexie>

<lexie voc="poser">
	<syntaxe>poser<expr cat="ingredient"/></syntaxe>
	<syntaxe>poser<expr cat="recipient"/></syntaxe>
	<syntaxe>poser<expr cat="ustensile"/></syntaxe>
```

```
		<syntaxe>poser<expr cat="ingredient"/>dans<expr
cat="rangement"/></syntaxe>
	</lexie>

	<lexie voc="verser">
		<syntaxe>verser<expr cat="ingredient"/></syntaxe>
		<syntaxe>verser<expr cat="recipient"/></syntaxe>
		<syntaxe>verser<expr cat="ingredient"/><expr cat="localisateur"/><expr
cat="recipient"/></syntaxe>
		<syntaxe>verser<expr cat="ingredient"/><expr cat="localisateur"/><expr
cat="rangement"/></syntaxe>
		<syntaxe>verser<expr cat="recette"/></syntaxe>
		<syntaxe>verser<expr cat="recette"/><expr cat="localisateur"/><expr
cat="recipient"/></syntaxe>
	</lexie>

	<lexie voc="sortir">
		<syntaxe>sortir<expr cat="recipient"/>de<expr cat="electromen"/></syntaxe>
		<syntaxe>sortir<expr cat="ingredient"/></syntaxe>
	</lexie>

	<lexie voc="ranger">
		<syntaxe>ranger<expr cat="endroit"/></syntaxe>
	</lexie>

</lexies>

<!-- les ordres -->

<lexies cat="ordre">
	<lexie voc="il faut">
		<syntaxe>il faut<expr cat="action"/></syntaxe>
	</lexie>
</lexies>

<!-- les questions -->

<lexies cat="question">
	<lexie voc="ou est">
		<syntaxe>ou est<expr cat="ingredient"/></syntaxe>
		<syntaxe>ou est<expr cat="recipient"/></syntaxe>
		<syntaxe>ou est<expr cat="ustensile"/></syntaxe>
	</lexie>
</lexies>

</lexique>
```

Annexe D – Document pour les apprenants

Cette annexe présente le document remis aux apprenants lors des mises à l'essai. Ce document présente les éléments du lexique que les apprenants peuvent utiliser pour communiquer avec le compagnon.

---

endroit

---

la cuisine

---

recette

---

le flan

la préparation

---

ingrédient

---

les ingrédients

le sucre

la farine

le chocolat

les oeufs

le beurre

le lait

la crème

---

récipient

---

le saladier

le moule

le plat

la casserole

Electro-ménager

---

le four

---

contenant

---

la bouteille

la boîte

le pot

---

contenu

à

de

---

ustensile

---

le fouet

la cuillère

la balance

le doseur

les ustensiles

---

rangement

---

le tiroir

le frigo

le placard

---

rangement-plat

---

le plan

le plan de travail

---

attribut

à gauche

à droite

au milieu

du bas

du haut

---

localisateur

---

dans

sur

---

action

---

bouillir

cuire

déposer

extirper

faire

fermer

fondre

incorporer

mélanger

mettre

ouvrir

préchauffer

prendre

préparer

poser

verser

sortir

ranger

---

ordre

---

il faut

---

question

---

où est

Table des illustrations

Tableau 1 : Tâche de l'arbre des tâches 50
Tableau 2 : Méthode de décomposition 51
Tableau 3 : Méthode opérationnelle 51
Tableau 4 : Aperçu de la structure d'unité minimale d'interaction, extrait de [Lehuen, 1997]72
Tableau 5 : Une UMI dans l'activité de production 73
Tableau 6 : Exemples de cas à traiter par l'analyseur 105
Tableau 7 : Création des indices de compréhension 109

Figure 1 : Le modèle interactionniste 22
Figure 2 : Disciplines connexes de l'alao, extrait de [Chanier, 1998] 26
Figure 3 : L'environnement FLUENT, extrait de [Schoelles *et al.*, 1997] 32
Figure 4 : L'environnement TLTS, extrait de [Johnson *et al.*, 2004] 34
Figure 5 : Tâche, interaction et langue 41
Figure 6 : Lien entre activité et langage 47
Figure 7 : Exemple d'arbre de tâches 52
Figure 8 : Lien tâche et interaction 53
Figure 9 : Interface du micromonde 55
Figure 10 : Exemples d'échange au cours de l'activité de compréhension 56
Figure 11 : Exemple de feedback au cours de l'activité de compréhension 56
Figure 12 : Exemple de question réponse dans l'activité de compréhension 56
Figure 13: Modèle de l'activité de compréhension 57
Figure 14 : Exemple d'échange au cours de l'activité de production 58
Figure 15 : Exemple d'échange où l'instruction est irréalisable 58
Figure 16 : Exemple de *feedback* au cours de l'activité de production 58
Figure 17 : Exemple de question/réponse au cours de l'activité de production 58
Figure 18 : Modèle de l'activité de production 59
Figure 19 : Lien interaction et activité 64
Figure 20 : Lecture simplifiée des types d'interaction 65
Figure 21 : Type d'interaction attendu au cours de l'activité de compréhension 66
Figure 22 : Type d'interaction quand une action de l'apprenant pose problème 67
Figure 23 : Type d'interaction de question-réponse 67
Figure 24 : Type d'interaction attendu au cours de l'activité de production 68
Figure 25 : Type d'interaction lorsqu'une instruction de l'apprenant pose problème 69
Figure 26 : Type d'interaction lorsqu'un énoncé contient des problèmes linguistiques 69
Figure 27 : Type d'interaction de question-réponse 70
Figure 28 : Enoncés de l'apprenant et réactions du partenaire 72
Figure 29 : Echange au cours de l'activité de production 74
Figure 30 : Représentation en UMI de l'échange Figure 29 74
Figure 31 : Correspondance entre le type d'interaction principal de l'activité de production et la représentation en UMIs 74
Figure 32 : Un échange avec un problème dans l'énoncé 75
Figure 33 : Représentation en UMI de l'échange Figure 32 75

Figure 34 : Correspondance entre le type d'interaction feedback de l'activité de production et la représentation en UMIs 76
Figure 35 : Exemple d'échange de question-réponse 76
Figure 36 : Représentation en UMI de l'échange de la Figure 35 76
Figure 37 : Correspondance entre le type d'interaction question-réponse et la représentation en UMIs 77
Figure 38 : Diagramme d'activité vu selon les UMIs 78
Figure 39 : Exemple d'énoncé donnant lieu à une négociation de sens 80
Figure 40 : Représentation en UMI de l'échange Figure 39 81
Figure 41 : Echange où deux incidences se succèdent 82
Figure 42 : Représentation en UMI de l'échange Figure 41 83
Figure 43 : Echange où deux incidences s'imbriquent 84
Figure 44 : Représentation en UMI de l'échange Figure 43 85
Figure 45 : Exemple d'analyse 92
Figure 46 : Lien entre langue et tâche 93
Figure 47 : Lexique 95
Figure 48 : Lien entre langage et interaction 95
Figure 49 : Granule 96
Figure 50 : Granularisation de l'énoncé « prendre les oeufs dans le frigo » 97
Figure 51 : Granules de l'énoncé « prendre les oeufs dans le frigo » 97
Figure 52 : Mise d'hypothèses sur « les eggs » 98
Figure 53 : Granules de l'énoncé « prendre les oeufs dans le frigo » 98
Figure 54 : Assemblage des granules de l'énoncé « prendre les oeufs dans le frigo » 99
Figure 55 : Assemblage des granules de l'énoncé « prendre les eggs dans le frigo » 99
Figure 56 : Résultats à couvertures égales mais marqueurs d'hypothèse différents 100
Figure 57 : Résultats avec marqueurs d'hypothèse égaux mais couvertures différentes 101
Figure 58 : Résultats mettant en concurrence couverture et hypothèse 102
Figure 59 : Deux résultats identiques en couverture et hypothèse 102
Figure 60 : Première phase de l'analyse, traitement des éléments de l'énoncé 103
Figure 61 : Exemple de réorganisation dans le but de retrouver un patron syntaxique 104
Figure 62 : Création d'un indice de compréhension à partir d'un élément reconnu 108
Figure 63 : Création d'un indice de compréhension à partir d'une action et d'un rangement reconnus 108
Figure 64 : Interface après la génération de la réaction du partenaire à l'énoncé « ouvrir le frigo » 110
Figure 65 : Architecture en deux parties de SAMPRAS 117
Figure 66 : Architecture modulaire, extrait de [Baker *et al.*, 2006] 118
Figure 67 : Architecture de SAMPRAS 119
Figure 68 : Interface homme-machine 121
Figure 69 : Extrait du fichier XML de description du monde 122
Figure 70 : Diagramme de classe simplifié de Sampras à partir de la classe MicroWorld 123
Figure 71 : Diagramme de classe de Sampras à partir de la classe MWLabel 123
Figure 72 : Principe de création d'un « *shadow facts* » 124
Figure 73 : Exemple de *shadow-facts* créés à partir du fichier xml décrit Figure 69 125
Figure 74 : Règle d'affichage des énoncés et de l'apparence du partenaire 125
Figure 75 : Exemple de règle de manipulation, ouverture d'un rangement 126
Figure 76 : Interface après ouverture du frigo 127
Figure 77 : Exemple de règle de manipulation, déplacement d'un objet 127
Figure 78 : Interface homme-machine après prise des œufs 128
Figure 79 : Exemple de règle de manipulation, déplacement d'un objet 128

Figure 80 : Interface homme-machine après ouverture de la boîte à oeufs 129
Figure 81 : La DTD servant de modèle de lexique 130
Figure 82 : Un extrait du fichier XML représentant le lexique..... 131
Figure 83 : Grammaire sémantique équivalente 132
Figure 84 : Modèle d'une granule dans Jess (template)..... 133
Figure 85 : Granule pour le texte « prendre » 134
Figure 86 : Granule pour le texte « les œufs » 134
Figure 87 : Résultat produit par l'analyse 135
Figure 88 : Diagramme UML du modèle « Tâche/Méthode/Opération »..... 137
Figure 89 : Modèles des tâches et des méthodes dans Jess (templates)..... 138
Figure 90 : Tâche t-faire-flan et son unique méthode 139
Figure 91 : Tâche prendre-le-lait..... 140
Figure 92 : Tâche « prendre le saladier » 141
Figure 93 : Identification d'une tâche à partir d'un énoncé 143
Figure 94 : Règle permettant de détecter une question 143
Figure 95 : Règle permettant de détecter une tâche 144
Figure 96 : Tâche identifiée à partir de la tâche résultat 145
Figure 97 : Règle de détection d'énoncés hors-tâche..... 145
Figure 98 : Règle de détection d'énoncés représentant de façon incomplète une tâche 146
Figure 99 : Granules créées lors d'une incidence 146
Figure 100 : Indice de compréhension 147
Figure 101 : Règle de création d'indice d'incompréhension 147
Figure 102 : Règle lançant la consultation de la mémoire 148
Figure 103 : Règle de création d'indice d'incompréhension 149
Figure 104 : Règle invoquant le module incidence-mémoire 150
Figure 105 : Règle invoquant le module incidence-mémoire 150
Figure 106 : Règle créant un indice de compréhension 151
Figure 107 : Evolution de la pile d'exécution..... 155
Figure 108 : méthode opérationnelle traiter-ordre-executable-bonne-couverture-bonne-syntaxe..... 156
Figure 109 : module TRAITER-ORDRE-EXECUTABLE-BCBS 157
Figure 110 : Échanges sans problèmes 160
Figure 111 : Échanges avec problèmes linguistiques..... 160
Figure 112 : Échanges avec problèmes liés à la tâche 160
Figure 113 : Répartition de la taille des énoncés pour un binôme 161
Figure 114 : séquence de négociation 163
Figure 115 : exemple de réaction 163

Références bibliographiques

Références bibliographiques par ordre alphabétique

[Antoniadis *et al.*, 2004]

Antoniadis Georges, Echinard Sandra, Kraif O., Lebarbé Thomas, et Ponton Claude (2004), *Modélisation de l'intégration de ressources TAL pour l'apprentissage des langues: la plateforme MIRTO*, Actes de la conférence TAL et apprentissage des langues, Grenoble, pp. 57-70.

[Baker *et al.*, 2006]

Baker M., Bronner A., Lehuen J., Quignard M. (2006), *Gestion de l'interaction et modèles de dialogues*, In: Environnements informatiques pour l'apprentissage humain, Monique Grandbastien et Jean-Marc Labat (ed.), Cognition et Traitement de I'information, Édité par Hermès, ISBN 2-7462-1171-8.

[Barré *et al.*, 2004]

Barré V., Choquet C., Corbiere A., et Iksal S. (2004), *Usage analysis in an e-Learning System: LD Representation Significance*, Actes de la conférence IEEE International Conference on Advanced Learning Technologies (ICALT'2004), Joensuu (Finlande), 30 août - 1er septembre 2004, pp. 570-574.

[Bellemain, 1992]

Bellemain F. (1992), *Conception, réalisation et utilisation d'un logiciel d'aide à l'enseignement de la géométrie: CabriGéomètre*, Thèse de l'Université Joseph Fourier, Grenoble (France).

[Betbeder M-L *et al.*, 2003]

Betbeder M-L et Tchounikine P. (2003), *Symba: a Framework to Support Collective Activities in an Educational Context*, Actes de la conférence International Conference on Computers in Education (ICCE 2003), Hong-Kong, China, pp. 188-196.

[Bilange, 1992]

Bilange Eric (1992), *Modélisation du dialogue oral finalisé par une approche structurelle*. Paris: Hermès

[Boylan *et al.*, 1998]

Boylan P. et Micarelli A. (1998), Learning language as "culture" with CALL, *CALL, Culture and the language curriculum*, L. G. Calivi, W., Ed. London, Springer, pp. 60-72.

[Bruillard, 1997]

Bruillard E. (1997), *Les machines à enseigner*. Paris: Hermes

[Bruillard *et al.*, 2000]

Bruillard E., Delozanne E., P. Leroux, et Delannoy P., Dubourg, X., Jacoboni, P.,Lehuen, J., Luzatti, D., Teutsch, P. (2000), Quinze ans de recherche informatique sur les sciences et techniques éducatives au LIUM, *Sciences et Techniques Educatives*, vol. 7, n° 1, pp. 87-185.

[Carbonell, 1970]

Carbonell J.R (1970), AI in CAI: an artificial approach to computer-assisted instruction, *IEEE Transactions on Man-Machine Systems*, vol. 11, n° 4, pp. 190-202.

[Chanier, 1998]

Chanier T. (1998), *Relations entre le TAL et l'ALAO ou l'ALAO un "simple" domaine d'application du TAL ?*, Actes de la conférence International conference on natura language processing and industrial application (NLP+IA'98), Moncton, Canada.

[Chanier *et al.*, 1996]

Chanier T. et Renié D. (1996), Collaboration and computer-assisted learning of a second language, *Computer-Assisted Language Learning*, vol. 8, n° 1, pp. 3-30.

[Chanier, 2000]

Chanier Thierry (2000), Hypermedia, interaction et apprentissage dans des systèmes d'information et de communication : résultats et agenda de recherche, *Apprendre une langue dans un environnement multimedia*, L. Duquette and M. Laurier, Eds. Québec, Les éditions LOGIQUES, pp. 53-89.

[Chapelle, 1998]

Chapelle Carol A. (1998), Multimedia CALL: lessons to be learned from research on instructed SLA, *Language Learning & Technology*, vol. 2, n° 1, pp. 22-34.

[Chapelle, 2000]

Chapelle Carol A. (2000), Interaction, communication et acquisition d'une langue seconde en ELAO, *Apprendre une langue dans un environnement multimédia*, L. Duquette and M. Laurier, Eds. Québec, Les éditions LOGIQUES, pp. 19-52.

[Chapelle, 2002]

Chapelle Carol A. (2002), La conception d'outils pour l'apprentissage des langues: la théorie peut-elle aider?, www.upmf-grenoble.fr/sciedu/actualite/#resumes.

[Chevallier, 1992]

Chevallier R. (1992), *Mise en oeuvre d'un modèle dynamique de dialogue dans un Tuteur Intelligent*, Thèse de l'Université du Maine, Le Mans.

[Chomsky, 1975]

Chomsky N. (1975), *Reflections on language* Temple Smith

[Coffinet-Rasseneur, 2004]

Coffinet-Rasseneur Dorothée (2004), *Saafir : un environnement support à l'appropriation d'une formation à distance par l'apprenant*, Thèse de l'Université du Maine, Le Mans, 230 pages.

[Collins *et al.*, 1982]

Collins A. et Stevens A.L (1982), Goals and strategies for inquiry teachers., *Advances in Instructional Psychology*, vol. 2.

[Coulthard, 1985]

Coulthard M. (1985), *An introduction to discourse analysis*

[Cuq, 2003]

Cuq Jean-Pierre (2003), *Dictionnaire de didactique du français langue étrangère et seconde*. Paris, asdifle

[De Pietro *et al.*, 1989]

De Pietro Jean-François, Matthey Marinette, et Py B. (1989), *Acquisition et contrat didactique: les séquences potentiellement acquisitionnelles dans la conversation exolingue*, Actes de la conférence troisième colloque régional de linguistique, Strasbourg, pp. 99-124.

[Delorme, 2002]

Delorme F. (2002), Modélisation d'activités interactives complexes pour l'apprntissage des langues assisté par ordinateur selon une approche communicative et actionnelle: les langages Lucrèce et Lautréamont, Université du Maine, Le Mans 117 p.

[Delozanne, 1992]

Delozanne Elisabeth (1992), Explications en EIAO : études à partir d'ELISE, un logiciel pour entraîner à une méthode de calcul de primitives, Université du Maine, Le Mans.

[Demaiziere, 1986]

Demaiziere F. (1986), *Enseignement assisté par ordinateur des langues*, Thèse de l'université Paris VII.

[Demaiziere *et al.*, 2005]

Demaiziere F. et Narcy-Combes J.P. (2005), *le défi de la dénativisation: apport des TIC*, Actes de la conférence Untele Compiègne, France.

[Despres, 2001]

Despres Christophe (2001), *Modélisation et Conception d'un Environnement de Suivi Pédagogique Synchrone d'Activités d'Apprentissage à Distance*, Thèse de l'Université du Maine, Le Mans, 286 pages.

[Dubourg, 1995]

Dubourg Xavier (1995), *Modélisation de l'interaction en EIAO, une approche évènementielle pour la réalisation du système REPERES*, Thèse de l'Université de caen, Caen.

[El Kechaï *et al.*, 2005]

El Kechaï N. et Despres C. (2005), *METISSE, modèle de description de tâches orienté assistance et suivi de l'apprenant*, Actes de la conférence 16e journées francophones d'Ingénierie des connaissances, Nice (France), 30 mai - 3 juin 2005.

[Furstenberg, 1997]

Furstenberg G. (1997), Scénarios d'exploitation pédagogique, *OUDART*, pp. 64-75.

[Gajo *et al.*, 2000]

Gajo L. et Mondada L. (2000), *Interactions et acquisitions en contexte*. Fribourg: Editions universitaires

[Graesser *et al.*, 2001]

Graesser A.C., Van Lehn K., Rose C., Jordan P., et Harter D. (2001), Intelligent tutoring systems with conversationnal dialogue *AI Magazine*, vol. 22, pp. 39-51.

[Gueye, 2005]

Gueye Omar (2005), *instrumentation des activités du tuteur en ligne: le cas de croisières, dispositif de formation ouverte et à distance en langues*, Thèse de l'Université du Maine, Le Mans, 367 pages.

[Hamburger, 1994]

Hamburger H. (1994), Foreign language immersion: Science, practice, and a System., *JAIED*, vol. 5(4), n° Special Issue on Language learning, pp. 429-453.

[Harrington, 1996]

Harrington M. (1996), Intelligent computer-assisted language learning, *ON-CALL*, vol. 10, n° 3.

[Harris *et al.*, 1989]

Harris Zellig, Gottfried Michael, Ryckman Thomas, Matick Paul, Daladier Anne, Harris T.N, et Harris S. (1989), *The form of information in science: analysis of an immunology sublanguage*: Kluwer Academic Publishers, 590 pages.

[Heift, 2003]

Heift T. (2003), Multiple Learners Error and Feedback: A Challenge for ICALL Systems, *CALICO*, vol. 20, n° 3, pp. 553-548.

[Holland *et al.*, 1999]

Holland V. Melissa, Kaplan Jonathan D., et A. Sabol Mark (1999), Preliminary tests of language learning in a speech-Intercative Graphics Microworld, *Calico*, vol. 16, n° 3, pp. 339-359.

[Holland *et al.*, 1993]

Holland V. Melissa, Maisano Richard, Alderks Cathie, et Martin Jeffrey (1993), Parsers in tutors: What are they good for ?, *Calico*, vol. 11, n° 1, pp. 28-46.

[Hymes, 1972]

Hymes D. (1972), On communicative competence, *Sociolinguistics*, pp. 269-293.

[Joab, 1990]

Joab M. (1990), *Modélisation d'un dialogue pédagogique en langage naturel*, Thèse de l'Paris 6, Paris.

[Johnson *et al.*, 2004]

Johnson W. Lewis, Marsella Stacy, Mote Nicolaus, Viljhamsson Hannes, Narayan Shrikanth, et Choi Sunhee (2004), *Tactical Language Training System: Supporting the rapid acquisition of foreign language and Cultural Skills*, Actes de la conférence InSTIL/ICALL, Venise, 17-19 june.

[Johnson *et al.*, 2000]

Johnson W. Lewis, Rickel J.W., et Lester J.C (2000), Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments, *International Journal of Artificial Intelligence in Education*, vol. 11, pp. 47-78.

[Karlström *et al.*, 2005]

Karlström P., Cerrato Pargman T., et Ramberg R. (2005), *Designing for collaboration in Intelligent Computer Assisted Language Learning*, Actes de la conférence 5th IEEE

International Conference on Advanced Learning Technologies, Kaohsiung, Taïwan, pp. 321-322.

[Krashen, 1982]

Krashen (1982), *Principles and Practice in Second Language Acquisition*. New York: Pergamon Press

[l'Europe, 2001]

l'Europe Conseil de (2001), *Cadre européen commun de référence pour les langues: apprendre, enseigner, évaluer.* Paris

[L'Haire *et al.*, 2003]

L'Haire S. et Vandeventer Faltin A (2003), Diagnostic d'erreurs dans le projet FreeText, *Alsic*, vol. 3, n° 2, pp. 21-37.

[Laperrousaz *et al.*, 2005]

Laperrousaz Christelle, Leroux Pascal, et Teutsch Philippe (2005), *Perception par le tuteur de l'activité d'un apprenant engagé dans une activité collective à distance*, Actes de la conférence EIAH'05
Montpellier (France), 25-27 mai, pp. 413-418.

[Larsen-freeman *et al.*, 1991]

Larsen-freeman et Long M.H. (1991), *An introduction to second language acquisition research*. London: Longman

[Lehuen *et al.*, 2006]

Lehuen J., Kitlinska S. (2006), *La Plateforme de Jeux de Rôles MEPA-2D - Vers la Dimension Globale de la Simulation Langagière pour l'Apprentissage d'une Langue Etrangère en Réseau*, In: Actes de TICE'06, 25-27 octobre 2006, Toulouse (France).

[Lehuen *et al.*, 2002]

Lehuen J, Lemeunier T, et Luzzati Daniel (2002), *Acquisition et études d'un corpus FLE, vers une analyse automatique des erreurs*, Actes de la conférence Quatrième colloque des Usages des Nouvelles Technologies dans l'Enseignement des Langues Etrangères, Compiègne (France), 28-30 mars 2002.

[Lehuen, 1997]

Lehuen Jérôme (1997), *Un modèle de dialogue dynamique et générique intégrant l'acquisition de sa compétence linguistique - Le système COALA*, Thèse de l'Université du Maine, Le Mans.

[Lemeunier, 2000]

Lemeunier Thierry (2000), *L'intentionnalité communicative dans le dialogue homme-machine en langue naturelle*, Thèse de l'Université du Maine, Le Mans, 235 pages.

[Long, 1985]

Long M.H. (1985), Input and second language acquisition theory, *Input in second language acquisition*. Rowley, MA, Newbury house publishers, pp. 377-393.

[Long, 1996]

Long M.H. (1996), The role of linguistic environment in second language acquisition, *Handbook of second language acquisition*, W. C. Ritchie and T. K. Bhatia, Eds. San Diego, Academic Press.

[Luzzati, 1995a]

Luzzati Daniel (1995a), *Le dialogue verbal homme-machine*. Paris: Masson

[Luzzati, 1995b]

Luzzati Daniel (1995b), *Le dialogue verbal homme machine*. Paris

[Mangenot, 2000]

Mangenot F. (2000), communication au 17ème colloque Triangle (Paris, Goethe Institut, 30/1/98), *Triangle 17, Multimédia et apprentissage des langues*, ENS Editions.

[Marcelli, 2004]

Marcelli Agnès (2004), *Temps et apprentissage d'une langue étrangère: vers un modèle bicontextuel d'enseignement/apprentissage, initié en présentiel et continué à distance à l'étranger (approche théorique et mise en oeuvre).* Thèse de l'Université de Franche-Comté, Besançon.

[Matthey, 1996]

Matthey Marinette (1996), *Apprentissage d'une langue et interaction verbale*. Berne: Peter Lang

[Michel, 2001]

Michel J (2001), conceptions d'activités interactives et pédagogiques au sein du projet Cordon Bleu, Université du Maine, Le Mans, rapport de stage de DEA.

[Michel *et al.*, 2004]

Michel J et Lehuen J (2004), *Un analyseur hypothético-déterministe pour l'apprentissage et la pratique d'une langue*, Actes de la conférence TAL et apprentissage des langues, Grenoble (France), 22 octobre 2004, pp. 13-22.

[Michel *et al.*, 2005]

Michel J et Lehuen J (2005), *Un environnement de simulation pour la pratique d'une langue étrangère*, Actes de la conférence EIAH'05, Montpellier (France), 25-27 mai.

[Moeschler, 1985]

Moeschler J. (1985), *Argumentation et conversation. Eléments pour une analyse pragmatique du discours*

[Papert, 1981]

Papert Seymour (1981), *Jaillissement de l'esprit: ordinateurs et apprentissage*. Paris: Flammarion

[Pica, 1994]

Pica T (1994), Research on negociation : What does it reveal about second-language learning conditions, processes and outcomes?, *Language Learning*, vol. 44, n° 3, pp. 493-527.

[Pica *et al.*, 1996]

Pica T et al (1996), Language learners' interaction : how does it adress the input, output and feedback needs of the L2 learners?, *TESOL quarterly*, vol. 30, n° 1, pp. 59-84.

[Pilkington, 1999]

Pilkington Rachel (1999), *Analysing Educationnal Dialogue Interaction: Towards Models that support learning*, Actes de la conférence Workshop at the 9th Internatioanl Conference on Artificial Intelligence in Education, Le Mans.

[Puren, 2001]

Puren Christian (2001), *Les nouvelles technologies face aux nouvelles options didactiques*, Actes de la conférence Colloque apprendre les langues européennes avec les nouvelles technologies, Paris, Goethe-Institut.

[Reeder, 2000]

Reeder F. (2000), *Could you repeat the question?*, Actes de la conférence AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications, Cape-Cod, USA, pp. 144-147.

[Rezeau, 2001]

Rezeau Joseph (2001), *Médiatisation et médiation pédagogique dans un environnmement multimédia*
Le cas de l'apprentissage de l'anglais en Histoire de l'Art à l'université, Thèse de l'Université Victor Ségalen Bordeaux 2, Bordeaux, 617 pages.

[Roulet, 1985]

Roulet E. (1985), *L'articulation du discours en français contemporain*. Berne: Peter Lang

[Schoelles *et al.*, 1997]

Schoelles Michael et Hamburger Henry (1997), *The NLP role in Animated Conversation for CALL*, Actes de la conférence ANLP: 5th Applied Natural Language Processing Conference, Washington, USA, pp. 127-134.

[Skehan, 2002]

Skehan P. (2002), *focus on form, tasks, and technology*, Actes de la conférence UNTELE, Compiègne.

[Slabbers *et al.*, 2005]

Slabbers N. et Knott A. (2005), *A system for generating teaching initiatives in a computer-aided language learning dialogue*, Actes de la conférence DIALOR, Nancy, France.

[Swain, 1985]

Swain (1985), Communicative competence: some roles of comprehensible input and comprehensible output in its development, *Input in second language acquisition*, S. M. Gass and C. G. Madden, Eds. Rowley, MA, Newbury House Publishers, pp. 235-253.

[Swain *et al.*, 1995]

Swain et Lapkin (1995), Problems in output and the cognitive processes they generate: a step towards second language learning, *Applied linguistics*, vol. 16, pp. 371-391.

[Taurisson *et al.*, 2005]

Taurisson Nel et Tchounikine Pierre (2005), *Une approche de l'apprentissage de l'organisation du travail collectif par la simulation*, Actes de la conférence EIAH'05, Montpellier (France), pp. 153-164.

[Tchounikine, 2002]

Tchounikine Pierre (2002), Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain *I3*, vol. 2, n° 1, pp. 59-95.

[Tchounikine *et al.*, 2004]

Tchounikine Pierre, Baker Mickael, Balacheff Nicolas, Baron Monique, Derycke Alain, Guin Dominique, Nicaud Jean-François, et Rabardel Pierre (2004), Platon-1: quelque dimensions pour l'analyse des travaux de recherche en conception d'EIAH STIC-CNRS 19 p.

[Trichet *et al.*, 1999]

Trichet F. et Tchounikine P. (1999), DSTM: a framework to operationalize and refine a Problem-Solving Method modeled in terms of Tasks and Methods, *International Journal of Experts Systems and Methods*, vol. 16, pp. 105-120.

[Underwood, 1984]

Underwood J (1984), *Linguistics, Computers, and the Language Teacher*: MA: Newbury House

[Vasseur *et al.*, 1992]

Vasseur Marie-Thérèse et Arditty J. (1992), *les activités réflexives en situation de communication exolingue : 10 ans de réflexion,*, Actes de la conférence colloque du réseau européen sur l'acquisition des langues, Lyon.

[Veronique, 1992]

Veronique D. (1992), Etat des lieux et perspectives, *Nouvelles perspectives dans l'étude de l'apprentissage d'une langue étrangère en milieu scolaire et en milieu social*, vol. 1, pp. 13.

[Voguel, 1995]

Voguel K. (1995), *L'interlangue. La langue de l'apprenant*. Toulouse: Presses Universitaires du Mirail

[Wenger, 1987]

Wenger E. (1987), *Artificial Intelligence and Tutoring Systems*: Morgn Kaufman

Références bibliographiques par ordre thématique

EIAH

[Baker *et al.*, 2006]

Baker M., Bronner A., Lehuen J., Quignard M. (2006), *Gestion de l'interaction et modèles de dialogues*, In: Environnements informatiques pour l'apprentissage humain, Monique Grandbastien et Jean-Marc Labat (ed.), Cognition et Traitement de I'information, Édité par Hermès, ISBN 2-7462-1171-8.

[Barré *et al.*, 2004]

Barré V., Choquet C., Corbiere A., et Iksal S. (2004), *Usage analysis in an e-Learning System: LD Representation Significance*, Actes de la conférence IEEE International Conference on Advanced Learning Technologies (ICALT'2004), Joensuu (Finlande), 30 août - 1er septembre 2004, pp. 570-574.

[Bellemain, 1992]

Bellemain F. (1992), *Conception, réalisation et utilisation d'un logiciel d'aide à l'enseignement de la géométrie: CabriGéomètre*, Thèse de l'Université Joseph Fourier, Grenoble (France).

[Betbeder M-L *et al.*, 2003]

Betbeder M-L et Tchounikine P. (2003), *Symba: a Framework to Support Collective Activities in an Educational Context*, Actes de la conférence International Conference on Computers in Education (ICCE 2003), Hong-Kong, China, pp. 188-196.

[Bruillard, 1997]

Bruillard E. (1997), *Les machines à enseigner*. Paris: Hermes

[Bruillard *et al.*, 2000]

Bruillard E., Delozanne E., P. Leroux, et Delannoy P., Dubourg, X., Jacoboni, P.,Lehuen, J., Luzatti, D., Teutsch, P. (2000), Quinze ans de recherche informatique sur les sciences et techniques éducatives au LIUM, *Sciences et Techniques Educatives*, vol. 7, n° 1, pp. 87-185.

[Carbonell, 1970]

Carbonell J.R (1970), AI in CAI: an artificial approach to computer-assisted instruction, *IEEE Transactions on Man-Machine Systems*, vol. 11, n° 4, pp. 190-202.

[Chevallier, 1992]

Chevallier R. (1992), *Mise en oeuvre d'un modèle dynamique de dialogue dans un Tuteur Intelligent*, Thèse de l'Université du Maine, Le Mans.

[Coffinet-Rasseneur, 2004]

Coffinet-Rasseneur Dorothée (2004), *Saafir : un environnement support à l'appropriation d'une formation à distance par l'apprenant*, Thèse de l'Université du Maine, Le Mans, 230 pages.

[Delorme, 2002]

Delorme F. (2002), Modélisation d'activités interactives complexes pour l'apprntissage des langues assisté par ordinateur selon une approche communicative et actionnelle: les langages Lucrèce et Lautréamont, Université du Maine, Le Mans 117 p.

[Delozanne, 1992]

Delozanne Elisabeth (1992), Explications en EIAO : études à partir d'ELISE, un logiciel pour entraîner à une méthode de calcul de primitives, Université du Maine, Le Mans.

[Despres, 2001]

Despres Christophe (2001), *Modélisation et Conception d'un Environnement de Suivi Pédagogique Synchrone d'Activités d'Apprentissage à Distance*, Thèse de l'Université du Maine, Le Mans, 286 pages.

[Dubourg, 1995]

Dubourg Xavier (1995), *Modélisation de l'interaction en EIAO, une approche évènementielle pour la réalisation du système REPERES*, Thèse de l'Université de caen, Caen.

[El Kechaï *et al.*, 2005]

El Kechaï N. et Despres C. (2005), *METISSE, modèle de description de tâches orienté assistance et suivi de l'apprenant*, Actes de la conférence 16e journées francophones d'Ingénierie des connaissances, Nice (France), 30 mai - 3 juin 2005.

[Graesser *et al.*, 2001]

Graesser A.C., Van Lehn K., Rose C., Jordan P., et Harter D. (2001), Intelligent tutoring systems with conversationnal dialogue *AI Magazine*, vol. 22, pp. 39-51.

[Joab, 1990]

Joab M. (1990), *Modélisation d'un dialogue pédagogique en langage naturel*, Thèse de l'université de Paris 6, Paris.

[Johnson *et al.*, 2000]

Johnson W. Lewis, Rickel J.W., et Lester J.C (2000), Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments, *International Journal of Artificial Intelligence in Education*, vol. 11, pp. 47-78.

[Laperrousaz *et al.*, 2005]

Laperrousaz Christelle, Leroux Pascal, et Teutsch Philippe (2005), *Perception par le tuteur de l'activité d'un apprenant engagé dans une activité collective à distance*, Actes de la conférence EIAH'05
Montpellier (France), 25-27 mai, pp. 413-418.

[Papert, 1981]

Papert Seymour (1981), *Jaillissement de l'esprit: ordinateurs et apprentissage*. Paris: Flammarion

[Pilkington, 1999]

Pilkington Rachel (1999), *Analysing Educationnal Dialogue Interaction: Towards Models that support learning*, Actes de la conférence Workshop at the 9th Internatioanl Conference on Artificial Intelligence in Education, Le Mans.

[Tchounikine, 2002]

Tchounikine Pierre (2002), Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain *I3*, vol. 2, n° 1, pp. 59-95.

[Tchounikine *et al.*, 2004]

Tchounikine Pierre, Baker Mickael, Balacheff Nicolas, Baron Monique, Derycke Alain, Guin Dominique, Nicaud Jean-François, et Rabardel Pierre (2004), Platon-1: quelque dimensions pour l'analyse des travaux de recherche en conception d'EIAH STIC-CNRS 19 p.

[Taurisson *et al.*, 2005]

Taurisson Nel et Tchounikine Pierre (2005), *Une approche de l'apprentissage de l'organisation du travail collectif par la simulation*, Actes de la conférence EIAH'05, Montpellier (France), pp. 153-164.

[Trichet *et al.*, 1999]

Trichet F. et Tchounikine P. (1999), DSTM: a framework to operationalize and refine a Problem-Solving Method modeled in terms of Tasks and Methods, *International Journal of Experts Systems and Methods*, vol. 16, pp. 105-120.

[Wenger, 1987]

Wenger E. (1987), *Artificial Intelligence and Tutoring Systems*: Morgan Kaufman

EIAH langues

[Antoniadis *et al.*, 2004]

Antoniadis Georges, Echinard Sandra, Kraif O., Lebarbé Thomas, et Ponton Claude (2004), *Modélisation de l'intégration de ressources TAL pour l'apprentissage des langues: la plateforme MIRTO*, Actes de la conférence TAL et apprentissage des langues, Grenoble, pp. 57-70.

[Boylan *et al.*, 1998]

Boylan P. et Micarelli A. (1998), Learning language as "culture" with CALL, *CALL, Culture and the language curriculum*, L. G. Calivi, W., Ed. London, Springer, pp. 60-72.

[Chanier *et al.*, 1996]

Chanier T. et Renié D. (1996), Collaboration and computer-assisted learning of a second language, *Computer-Assisted Language Learning*, vol. 8, n° 1, pp. 3-30.

[Chanier, 1998]

Chanier T. (1998), *Relations entre le TAL et l'ALAO ou l'ALAO un "simple" domaine d'application du TAL ?*, Actes de la conférence International conference on natura language processing and industrial application (NLP+IA'98), Moncton, Canada.

[Chanier, 2000]

Chanier T. (2000), Hypermedia, interaction et apprentissage dans des systèmes d'information et de communication : résultats et agenda de recherche, *Apprendre une langue dans un environnement* multimédia, L. Duquette et M. Laurier, Eds. Quéec, Les éditions LOGIQUES, pp. 53-89

[Chapelle, 1998]

Chapelle Carol A. (1998), Multimedia CALL: lessons to be learned from research on instructed SLA, *Language Learning & Technology*, vol. 2, n° 1, pp. 22-34.

[Chapelle, 2000]

Chapelle Carol A. (2000), Interaction, communication et acquisition d'une langue seconde en ELAO, *Apprendre une langue dans un environnement multimédia*, L. Duquette and M. Laurier, Eds. Québec, Les éditions LOGIQUES, pp. 19-52.

[Chapelle, 2002]

Chapelle Carol A. (2002), La conception d'outils pour l'apprentissage des langues: la théorie peut-elle aider?, www.upmf-grenoble.fr/sciedu/actualite/#resumes.

[Demaiziere, 1986]

Demaiziere F. (1986), *Enseignement assisté par ordinateur des langues*, Thèse de l'université Paris VII.

[Demaiziere *et al.*, 2005]

Demaiziere F. et Narcy-Combes J.P. (2005), *le défi de la dénativisation: apport des TIC*, Actes de la conférence Untele Compiègne, France.

[Gueye, 2005]

Gueye Omar (2005), *instrumentation des activités du tuteur en ligne: le cas de croisières, dispositif de formation ouverte et à distance en langues*, Thèse de l'Université du Maine, Le Mans, 367 pages.

[Hamburger, 1994]

Hamburger H. (1994), Foreign language immersion: Science, practice, and a System., *JAIED*, vol. 5(4), n° Special Issue on Language learning, pp. 429-453.

[Harrington, 1996]

Harrington M. (1996), Intelligent computer-assisted language learning, *ON-CALL*, vol. 10, n° 3.

[Heift, 2003]

Heift T. (2003), Multiple Learners Error and Feedback: A Challenge for ICALL Systems, *CALICO*, vol. 20, n° 3, pp. 553-548.

[Holland *et al.*, 1999]

Holland V. Melissa, Kaplan Jonathan D., et A. Sabol Mark (1999), Preliminary tests of language learning in a speech-Intercative Graphics Microworld, *Calico*, vol. 16, n° 3, pp. 339-359.

[Holland *et al.*, 1993]

Holland V. Melissa, Maisano Richard, Alderks Cathie, et Martin Jeffrey (1993), Parsers in tutors: What are they good for ?, *Calico*, vol. 11, n° 1, pp. 28-46.

[Johnson *et al.*, 2004]

Johnson W. Lewis, Marsella Stacy, Mote Nicolaus, Viljhamsson Hannes, Narayan Shrikanth, et Choi Sunhee (2004), *Tactical Language Training System: Supporting the rapid acquisition of foreign language and Cultural Skills*, Actes de la conférence InSTIL/ICALL, Venise, 17-19 june.

[Karlström *et al.*, 2005]

Karlström P., Cerrato Pargman T., et Ramberg R. (2005), *Designing for collaboration in Intelligent Computer Assisted Language Learning*, Actes de la conférence 5th IEEE International Conference on Advanced Learning Technologies, Kaohsiung, Taïwan, pp. 321-322.

[L'Haire *et al.*, 2003]

L'Haire S. et Vandeventer Faltin A (2003), Diagnostic d'erreurs dans le projet FreeText, *Alsic*, vol. 3, n° 2, pp. 21-37.

[Lehuen *et al.*, 2006]

Lehuen J., Kitlinska S. (2006), *La Plateforme de Jeux de Rôles MEPA-2D - Vers la Dimension Globale de la Simulation Langagière pour l'Apprentissage d'une Langue Etrangère en Réseau*, In: Actes de TICE'06, 25-27 octobre 2006, Toulouse (France).

[Lehuen *et al.*, 2002]

Lehuen J, Lemeunier T, et Luzzati Daniel (2002), *Acquisition et études d'un corpus FLE, vers une analyse automatique des erreurs*, Actes de la conférence Quatrième colloque des Usages des Nouvelles Technologies dans l'Enseignement des Langues Etrangères, Compiègne (France), 28-30 mars 2002.

[Mangenot, 2000]

Mangenot F. (2000), communication au 17ème colloque Triangle (Paris, Goethe Institut, 30/1/98), *Triangle 17, Multimédia et apprentissage des langues*, ENS Editions.

[Marcelli, 2004]

Marcelli Agnès (2004), *Temps et apprentissage d'une langue étrangère: vers un modèle bicontextuel d'enseignement/apprentissage, initié en présentiel et continué à distance à l'étranger (approche théorique et mise en oeuvre).* Thèse de l'Université de Franche-Comté, Besançon.

[Michel, 2001]

Michel J (2001), conceptions d'activités interactives et pédagogiques au sein du projet Cordon Bleu, Université du Maine, Le Mans, rapport de stage de DEA.

[Michel *et al.*, 2004]

Michel J et Lehuen J (2004), *Un analyseur hypothético-déterministe pour l'apprentissage et la pratique d'une langue*, Actes de la conférence TAL et apprentissage des langues, Grenoble (France), 22 octobre 2004, pp. 13-22.

[Michel *et al.*, 2005]

Michel J et Lehuen J (2005), *Un environnement de simulation pour la pratique d'une langue étrangère*, Actes de la conférence EIAH'05, Montpellier (France), 25-27 mai.

[Puren, 2001]

Puren Christian (2001), *Les nouvelles technologies face aux nouvelles options didactiques*, Actes de la conférence Colloque apprendre les langues européennes avec les nouvelles technologies, Paris, Goethe-Institut.

[Reeder, 2000]

Reeder F. (2000), *Could you repeat the question?*, Actes de la conférence AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications, Cape-Cod, USA, pp. 144-147.

[Rezeau, 2001]

Rezeau Joseph (2001), *Médiatisation et médiation pédagogique dans un environnmement multimédia. Le cas de l'apprentissage de l'anglais en Histoire de l'Art à l'université*, Thèse de l'Université Victor Ségalen Bordeaux 2, Bordeaux, 617 pages

[Schoelles *et al.*, 1997]

Schoelles Michael et Hamburger Henry (1997), *The NLP role in Animated Conversation for CALL*, Actes de la conférence ANLP: 5th Applied Natural Language Processing Conference, Washington, USA, pp. 127-134.

[Skehan, 2002]

Skehan P. (2002), *focus on form, tasks, and technology*, Actes de la conférence UNTELE, Compiègne.

[Slabbers *et al.*, 2005]

Slabbers N. et Knott A. (2005), *A system for generating teaching initiatives in a computer-aided language learning dialogue*, Actes de la conférence DIALOR, Nancy, France.

[Underwood, 1984]

Underwood J (1984), *Linguistics, Computers, and the Language Teacher*: MA: Newbury House

Apprentissage des langues

[Candelier, 1996]

Candelier Michel (1996), *savoir métalinguistique et intériorisation: ce que la recherche nous dit et ce qu'elle devrait nous dire*, Actes de la conférence Convegno nazionale LEND.

[Chomsky, 1975]

Chomsky N. (1975), *Reflections on language* Temple Smith

[Collins *et al.*, 1982]

Collins A. et Stevens A.L (1982), Goals and strategies for inquiry teachers., *Advances in Instructional Psychology*, vol. 2.

[Cuq, 2003]

Cuq Jean-Pierre (2003), *Dictionnaire de didactique du français langue étrangère et seconde*. Paris, asdifle

[De Pietro *et al.*, 1989]

De Pietro Jean-François, Matthey Marinette, et Py B. (1989), *Acquisition et contrat didactique: les séquences potentiellement acquisitionnelles dans la conversation exolingue*, Actes de la conférence troisième colloque régional de linguistique, Strasbourg, pp. 99-124.

[Dolz *et al.*, 2001]

Dolz J., Scheunwly B., Thévenaz T., et Wirthner M (2001), *Les tâches et leurs entours en classe de français*, Actes de la conférence Colloque DFLM.

[Furstenberg, 1997]

Furstenberg G. (1997), Scénarios d'exploitation pédagogique, *OUDART*, pp. 64-75.

[Gajo *et al.*, 2000]

Gajo L. et Mondada L. (2000), *Interactions et acquisitions en contexte*. Fribourg: Editions universitaires

[Hymes, 1972]

Hymes D. (1972), On communicative competence, *Sociolinguistics*, pp. 269-293.

[Krashen, 1982]

Krashen (1982), *Principles and Practice in Second Language Acquisition*. New York: Pergamon Press

[l'Europe, 2001]

l'Europe Conseil de (2001), *Cadre européen commun de référence pour les langues: apprendre, enseigner, évaluer.* Paris

[Larsen-freeman *et al.*, 1991]

Larsen-freeman et Long M.H. (1991), *An introduction to second language acquisition research*. London: Longman

[Long, 1985]

Long M.H. (1985), Input and second language acquisition theory, *Input in second language acquisition*. Rowley, MA, Newbury house publishers, pp. 377-393.

[Long, 1996]

Long M.H. (1996), The role of linguistic environment in second language acquisition, *Handbook of second language acquisition*, W. C. Ritchie and T. K. Bhatia, Eds. San Diego, Academic Press.

[Matthey, 1996]

Matthey Marinette (1996), *Apprentissage d'une langue et interaction verbale*. Berne: Peter Lang

[Pica, 1994]

Pica T (1994), Research on negociation : What does it reveal about second-language learning conditions, processes and outcomes?, *Language Learning*, vol. 44, n° 3, pp. 493-527.

[Pica *et al.*, 1996]

Pica T et al (1996), Language learners' interaction : how does it adress the input, output and feedback needs of the L2 learners?, *TESOL quarterly*, vol. 30, n° 1, pp. 59-84.

[Swain, 1985]

Swain (1985), Communicative competence: some roles of comprehensible input and comprehensible output in its development, *Input in second language acquisition*, S. M. Gass and C. G. Madden, Eds. Rowley, MA, Newbury House Publishers, pp. 235-253.

[Swain *et al.*, 1995]

Swain et Lapkin (1995), Problems in output and the cognitive processes they generate: a step towards second language learning, *Applied linguistics*, vol. 16, pp. 371-391.

[Vasseur *et al.*, 1992]

Vasseur Marie-Thérèse et Arditty J. (1992), *les activités réflexives en situation de communication exolingue : 10 ans de réflexion*,, Actes de la conférence colloque du réseau européen sur l'acquisition des langues, Lyon.

[Veronique, 1992]

Veronique D. (1992), Etat des lieux et perspectives, *Nouvelles perspectives dans l'étude de l'apprentissage d'une langue étrangère en milieu scolaire et en milieu social*, vol. 1, pp. 13.

[Voguel, 1995]

Voguel K. (1995), *L'interlangue. La langue de l'apprenant*. Toulouse: Presses Universitaires du Mirail

Dialogue homme-machine et linguistique

[Bilange, 1992]

Bilange Eric (1992), *Modélisation du dialogue oral finalisé par une approche structurelle*. Paris: Hermès

[Coulthard, 1985]

Coulthard M. (1985), *An introduction to discourse analysis*

[Harris *et al.*, 1989]

Harris Zellig, Gottfried Michael, Ryckman Thomas, Matick Paul, Daladier Anne, Harris T.N, et Harris S. (1989), *The form of information in science: analysis of an immunology sublanguage*: Kluwer Academic Publishers, 590 pages.

[Lehuen, 1997]

Lehuen Jérôme (1997), *Un modèle de dialogue dynamique et générique intégrant l'acquisition de sa compétence linguistique - Le système COALA*, Thèse de l'Université du Maine, Le Mans.

[Lemeunier, 2000]

Lemeunier Thierry (2000), *L'intentionnalité communicative dans le dialogue homme-machine en langue naturelle*, Thèse de l'Université du Maine, Le Mans, 235 pages.

[Luzzati, 1995a]

Luzzati Daniel (1995a), *Le dialogue verbal homme-machine*. Paris: Masson

[Luzzati, 1995b]

Luzzati Daniel (1995b), *Le dialogue verbal homme machine*. Paris

[Moeschler, 1985]

Moeschler J. (1985), *Argumentation et conversation. Eléments pour une analyse pragmatique du discours*

[Roulet, 1985]

Roulet E. (1985), *L'articulation du discours en français contemporain*. Berne: Peter Lang