TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AFP:	Action for person
AFMO:	Action for movable object
AFNMO:	Action for non-movable object
BN:	Bayesian network
C:	Concepts
CC:	Complex command
CPN:	Colored Petri Nets
CPN-tools:	Colored petri net – tools
Con:	Context
GPS:	Global Positioning System
IL:	Intended location
LO:	Location
M:	Les mots clés de la requête
MO:	Manual Output
MO_j:	Output modality i.
Mo:	Movable object
M_{out}:	Manual output
NMO:	No Movable Object
P:	Person
RB :	Réseau bayésien
RPCS :	Les réseaux de petri colorés
ST:	Sub-task

VO: Vocal Output

VIO: Visual Output

VO_{out}: Vocal output

VI_{out}: Visual output

OWL: Ontology Web Language

XML: Extensible Markup Language

INTRODUCTION

Cadre de recherche

La communication joue un rôle primordial dans notre vie courante. Elle permet aux humains de se comprendre et de communiquer soit comme étant des individus ou encore en tant que groupes indépendants. Cette communication se fait à travers plusieurs modalités naturelles comme la parole, les gestes, le regard, les expressions faciales.

Les humains ont une capacité très développée à transmettre des idées entre eux et de réagir de manière appropriée. Cela est dû au fait du partage de la langue, aussi bien à la compréhension commune du fonctionnement des choses et à la compréhension implicite des situations quotidiennes.

L'un des plus grands défis de l'informatique a toujours été la création des systèmes qui permettent la transparence et la flexibilité de l'interaction homme-machine (Sears et Jacko, 2007) et (Alm, Alfredson et Ohlsson, 2009). Les chercheurs visent toujours à satisfaire les besoins des utilisateurs et à proposer des systèmes intelligents, plus naturels et plus conviviaux.

Mais sans intervention externe, les machines/ordinateurs ne comprennent pas notre langue, ne comprennent pas comment le monde fonctionne et ne peuvent percevoir des informations pour une situation donnée.

Ainsi, des efforts ont été orientés vers la création de systèmes qui facilitent la communication entre l'homme et la machine (Yuen, Tang et Wang, 2002) et de permettre à un utilisateur d'utiliser des périphériques multimédias invoquant les modalités naturelles (regard, parole, geste, etc.) pour communiquer ou échanger des informations avec des applications.

Ces systèmes reçoivent des entrées à partir de capteurs ou de dispositifs électroniques (caméra, microphone, etc.) et ils font l'interprétation et la compréhension de ces entrées, c'est la multimodalité (Ringland et Scahill, 2002) (Carnielli et al., 2008). Un exemple connu de ces systèmes est celui de Bolt "Put that there" (Bolt, 1980a) où il a utilisé le geste et la parole pour déplacer des objets.

Depuis les années quatre-vingt, le développement rapide dans le monde des technologies de l'information a permis de créer des systèmes qui interfèrent d'une manière harmonieuse avec l'utilisateur. Ceci est dû à l'émergence d'une technologie appelée : l'interaction multimodale. Ces systèmes ont permis aux utilisateurs, en particulier pour ceux qui ne peuvent utiliser un clavier ou une souris, les malvoyants, les utilisateurs qui sont équipés d'appareils mobiles, les utilisateurs avec handicap, etc., de se servir de ces modalités naturelles (la parole, le geste, le regard, etc.) pour interagir avec la machine avec une expressivité plus riche et plus varié. Ils sont connus dans la littérature sous le vocable *systèmes multimodaux*. Les systèmes multimodaux améliorent donc l'accessibilité pour une grande variété d'utilisateurs.

Ce projet de recherche s'inscrit dans ce cadre, et plus spécifiquement, il porte sur la création d'un module de fission pour l'interaction multimodale. La fission est une étape cruciale et déterminante dans un système multimodal.

Nous proposons une nouvelle solution de modélisation, une architecture qui facilite le travail du module de fission. Pour cela, nous avons utilisé une ontologie qui décrit l'environnement et prend en considération différents contextes. Nous proposons également l'utilisation d'un réseau bayésien qui permet de traiter les exceptions selon le contexte.

Dans ce qui suit, nous présentons le but principal de notre projet de recherche et les objectifs identifiés pour aboutir à ce but. La section *Contribution* montre les innovations apportées pour la modélisation d'une architecture d'interaction multimodale et plus spécifiquement, la création d'un module de fission pour l'interaction multimodale.

Ce chapitre présente également les différentes phases de la méthodologie de recherche nécessaires pour aboutir au but final, qui consiste en la création d'un système de fission multimodal.

Problématique de recherche

Ce projet de recherche s'inscrit dans la lignée du développement d'un système multimodal muni d'une certaine autonomie et d'une certaine capacité de prise de décision. C'est un domaine en plein essor dont la tendance s'oriente vers la production de machines de plus en plus performantes, faciles à utiliser, intégrant plusieurs modalités et pour une grande variété d'applications. L'objectif étant de converger vers des systèmes multimodaux personne-machine / machine-machine qui renforcent l'interaction. Ces systèmes comportent généralement une interface multimodale d'entrée et une interface multimodale de sortie. Via l'interface de sortie, le système doit être capable de choisir parmi les modalités disponibles, celles qui satisfont au mieux les contraintes de l'environnement, les besoins fonctionnels de la tâche à exécuter et les préférences de l'utilisateur.

Le système sera en mesure d'interpréter une commande complexe et la subdiviser en sous-tâches élémentaires et présenter celles-ci sur les modalités de sorties. Dans ce cas, nous parlons de processus de fission.

Notre problématique de recherche consiste à développer un système expert capable de fournir des services aux différentes applications multimodales en utilisant des techniques de modélisation comme les ontologies, les patterns, le contexte et les réseaux bayésiens. Ce système doit être en mesure de recevoir une commande complexe, de la subdiviser en sous-*tâches* élémentaires et les présenter sur les modalités de sortie disponibles.

Nous allons spécifier et développer un composant de fission pour l'interaction multimodale et présenter un algorithme de fission efficace. Pour réaliser nos objectifs autour du composant de fission, nous allons créer une architecture sensible au contexte capable de gérer plusieurs

modules distribués et capable de s'adapter automatiquement aux changements dynamiques du contexte d'interaction (environnement, système, utilisateur).

Pour atteindre cet objectif, nous énumérons quelques défis qui doivent être abordés afin de développer notre système :

1. Quels sont les modules nécessaires pour la conception de l'architecture d'un système de fission multimodal ?

2. Comment représenter les informations multimodales ?

3. Comment fissionner et interpréter les informations ?

4. Comment le système gère-t-il les données incertaines ou ambiguës pendant le processus de fission ?

5. Quelle est la représentation optimale de l'environnement dans notre architecture ?

Buts et objectifs de recherche

Notre objectif consiste à développer une infrastructure intelligente qui permet aux différentes applications équipées d'appareils mobiles d'utiliser des modalités en sortie, selon leurs disponibilités. Le système est intelligent : il permet de détecter la localisation de l'utilisateur, son profil et ses tâches, et de sélectionner les modalités à être utilisées en sortie, selon le contexte courant.

Le module de fission est un composant fondamental dans un système interactif multimodal. Il a pour rôle de subdiviser une commande complexe en sous-tâches élémentaires et de les présentées à la sortie. Cette présentation peut varier selon le contexte, la tâche et les services.

Nous nous concentrons plus particulièrement sur les aspects conception, spécification, construction et évaluation du module de fission. Ceci comprend :

a. La définition d'une architecture générale d'un système sensible au contexte d'interaction multimodale ;

b. La création d'un système capable d'utiliser un nombre de modalités supérieur à ceux utilisées dans la littérature ;

c. La modélisation des modalités, des règles de fission et du contexte d'interaction;

d. L'utilisation des patterns pour lier les différents modes de la fission au contexte d'interaction ;

e. La définition d'un algorithme de fission ;

f. La validation du système en utilisant les réseaux de pétri colorés (RPCS) (Jensen, 1996) en utilisant le logiciel de simulation CPN tools (colored Petri net) et ;

g. Le développement de deux prototypes : 1) interface pour le contrôle d'un robot qui peut être utilisé pour assister des personnes avec handicap ou des personnes âgées et 2) une interface GPS pour indiquer le trajet à un conducteur de voitures.

Méthodologie de recherche

Pour réaliser nos objectifs cités plus haut, nous devons suivre plusieurs étapes de recherche. La méthodologie de recherche proposée dans la Figure 1.1 présente les phases et les objectifs nécessaires pour la réalisation de chacune d'elles.

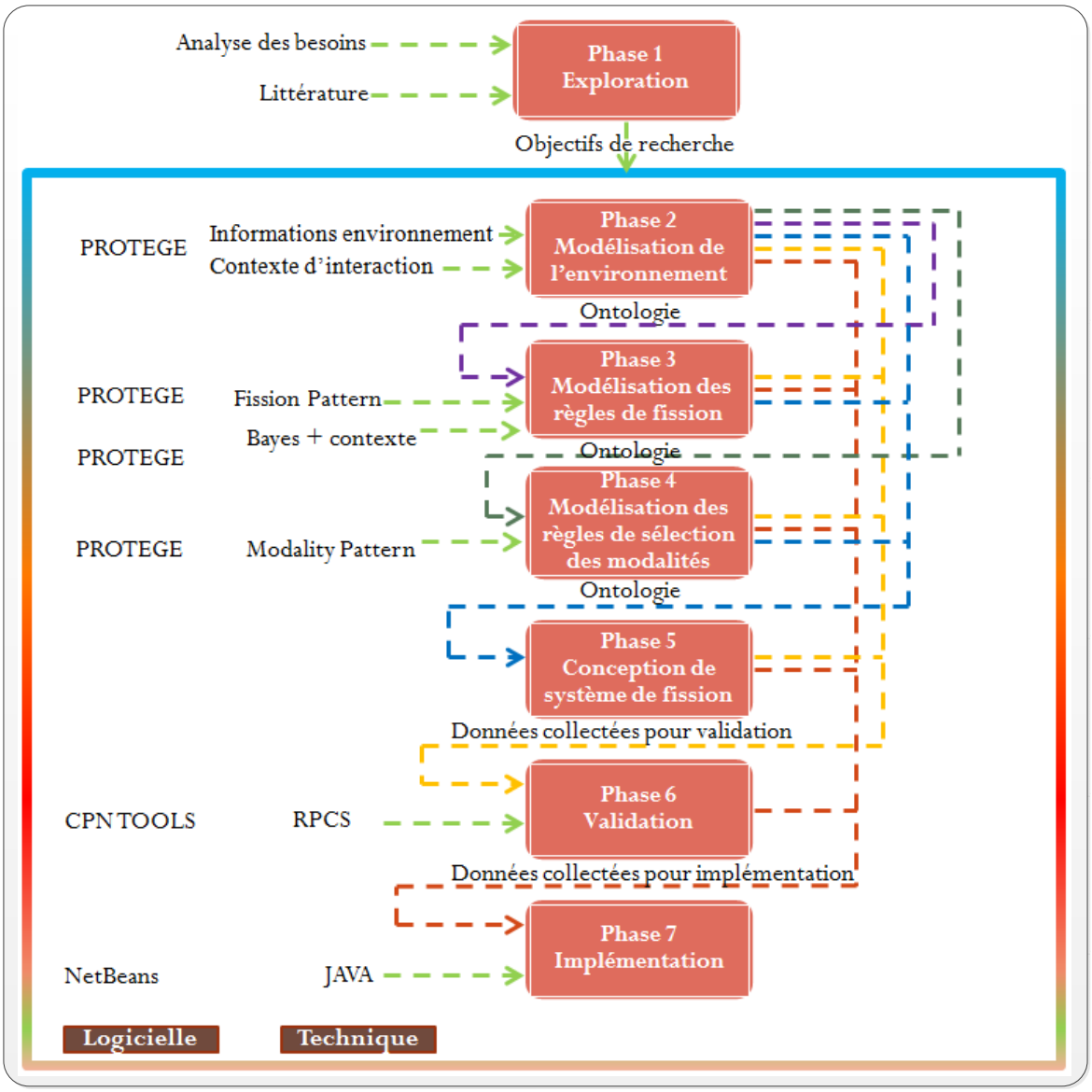


Figure 1.1 Méthodologie

1. L'état de l'art (Phase 1)

Cette partie est consacrée à la présentation, à l'analyse et à la critique des travaux et des idées disponibles dans la littérature. Elle est divisée en quatre sous-parties, la première discute la multimodalité en général. La deuxième partie présente en détail les applications multimodales existantes. La troisième partie discute le travail acquis sur la fission multimodale, les définitions, les différentes architectures (si connues), les types des données

à fissionner. Ensuite nous présentons les ontologies et les patterns. Finalement, une discussion sera présentée sous la forme d'analyse et critique des informations présentées, afin d'aboutir à une conclusion de base nécessaire pour les étapes suivantes.
Cette partie est détaillée dans le chapitre 1 (revue de la littérature) et dans les 4 articles inclus dans cette thèse, partie « related work ».

2. Modélisation de l'environnement (Phase 2)

Dans cette partie, nous modélisons notre environnement sous forme d'ontologie. Le but de l'ontologie est de modéliser un ensemble de connaissances dans un domaine donné avec une forme utilisable par la machine. Notre ontologie détaille l'environnement de la maison. Cette ontologie peut être mise à jour pour cibler d'autres contextes tels que les hôpitaux, les lieux de travail, etc. Notre ontologie décrit :

- les objets existants dans l'environnement ;
- les différentes modalités que les systèmes peuvent utiliser ;
- le contexte d'interaction (utilisateur, environnement et système) qui affecte la sélection des modalités ;
- un vocabulaire sous forme d'instances ;
- des modèles grammaticaux pour vérifier le sens d'évènements ;
- les patterns de fission. Ils permettent de subdiviser une commande complexe en sous-tâches élémentaires ;
- les patterns de modalités. Ils permettent d'associer pour chaque sous-tâche le ou les modalités adéquates ;

- les modèles pour le réseau bayésien.

Notre base de connaissances est modélisée en utilisant le logiciel PROTEGE.

3. Modélisation des règles de fission (Phase 3)

Pour réaliser la fission d'une commande, nous avons utilisé la technique de pattern : Fission pattern. Nous avons défini une quinzaine de Fission patterns dans l'ontologie. Dans le cas d'ambiguïtés ou d'incertitudes, une méthode basée sur le contexte utilisant le réseau bayésien est utilisée. Ces patterns et les modèles de réseau bayésien (RB) sont modélisés dans l'ontologie en utilisant le logiciel PROTEGE.

4. Modélisation des règles de sélections des modalités (Phase 4)

Cette phase est divisée en deux parties :

- sélection de modalités selon le contexte d'interaction : le système a la capacité de sélectionner les modalités et les médias adéquats en se basant sur le contexte d'interaction (utilisateur, système et environnement) ;

- association des modalités disponibles à des sous-tâches : dans cette partie, le système détermine quelles sont les modalités adéquates pour chaque sous-tâche en utilisant les modalités pattern qui sont stockés dans l'ontologie.

5. Conception du moteur de fission (Phase 5)

Le module de fission est un composant fondamental du système interactif multimodal. Il est utilisé essentiellement en sortie de celui-ci. Il a pour rôle de subdiviser les requêtes faites par l'utilisateur en requêtes fragmentaires, de les associer aux modalités appropriées et de les

présenter en sortie selon les médias disponibles. Le sens de la requête peut varier selon le contexte, la tâche et les services.

6. Validation (Phase 6)

Nous avons validé le fonctionnement de notre architecture en utilisant le réseau de Pétri coloré. Nous avons défini dans le réseau les différents composants de notre architecture et nous avons validé le fonctionnement de l'algorithme de la fission qui se base sur l'utilisation des patterns pour 1) sélectionner les sous-tâches élémentaires et 2) associer les modalités adéquates pour chaque sous-tâche.

7. Le développement d'un prototype (Phase 7)

Nous avons développé deux applications réelles pour valider nos approches. Nous avons montré que la solution proposée est applicable dans un environnement réel. Nous avons présenté deux interfaces : 1) interface de contrôle d'un robot; elle est implémentée pour valider le processus de fission en utilisant la technique des patterns et 2) interface pour GPS ; elle est implémentée pour valider la méthode basée sur le contexte en utilisant le réseau bayésien en cas d'ambiguïté ou d'incertitude.

Contributions

Cette thèse est axée sur le processus de fission. Nous proposons une nouvelle solution méthodologique en modélisant une architecture qui facilite le travail d'un module de fission, en définissant une ontologie qui contient différents scénarios et qui décrit l'environnement dans lequel un système multimodal évolue. L'architecture proposée surpasse les faiblesses des architectures étudiées dans l'état de l'art, elle a trois caractéristiques principales :

Transparence : la manipulation d'un grand nombre de modalités qui permet une utilisation générique sur une variété d'applications et de domaines ;

Flexibilité : l'utilisation de l'ontologie permet la description de l'environnement et des scénarios ;

Cohérence : la description de plus grand nombre d'objets et d'évènements dans l'environnement. Ainsi, l'utilisation des règles de fission et des règles de sélection de modalité de sortie. Ces règles sont ajoutées dans l'ontologie. Donc un système basé sur des règles logique.

Pour atteindre notre objectif, nous énumérons les solutions nécessaires pour les défis présentés dans la section problématique (1.2) :

1. Nous avons spécifié, défini et développé tous les composants nécessaires pour le système de fission multimodal ;

2. Nous avons modélisé sémantiquement l'environnement. Nous avons créé une architecture sensible au contexte capable i) de gérer plusieurs modules répartis dans le réseau et 2) de s'adapter automatiquement aux changements dynamiques du contexte d'interaction (utilisateur, environnement, système) ;

3. Nous avons présenté un algorithme qui décrit le mécanisme de fission. Il comprend les règles de fission et les règles pour le choix de modalités de sortie ;

4. Nous avons introduit une nouvelle méthode basée sur le contexte en utilisant un réseau bayésien pour résoudre le problème de l'incertitude durant le processus de fission dans un système multimodal ;

5. Nous avons adopté une solution basée sur l'ontologie. Les modalités, les scénarios, les objets et leurs caractéristiques sont stockées dans l'ontologie qui décrit la relation entre eux.

Pour conclure, notre contribution principale consiste en :

- l'utilisation de l'ontologie pour résoudre notre problème qui consiste à modéliser les données et les rendre dynamiques et faciles à mettre à jour. Cela est détaillé dans le chapitre trois (article 2) ;

- l'utilisation de pattern de fission pour sélectionner les sous-tâches adéquates à la commande complexe et l'utilisation de pattern de modalité pour sélectionner les modalités convenables à chaque sous-tâche. Cela est détaillé dans le chapitre deux et trois (article 1 et 2) ;

- l'utilisation d'une nouvelle méthode basée sur le contexte en utilisant un réseau bayésien pour surmonter le problème d'ambigüité ou d'incertitude pendant le processus de fission. Cela est détaillé dans le chapitre quatre (article 3) ;

- nos approches ont été validées en présentant deux interfaces. Dans la première, nous avons présenté notre algorithme concernant les patterns ; dans la deuxième, nous avons validé notre nouvelle méthode concernant le réseau bayésien (chapitre 4).

Organisation de la thèse

L'organisation de cette thèse est la suivante :

Le premier chapitre présente l'état de l'art qui aborde les travaux réalisés dans le domaine de l'interaction multimodale.

Les quatre chapitres qui suivent sont des travaux publiés.

Le deuxième chapitre est un article qui a été publié dans le journal « Journal of Emerging Trends in Computing and Information Sciences » :

A. Zaguia, A. Wehbi, C. Tadj, A. Ramdane-Chérif, « *Multimodal Fission for Interaction Architecture* », Journal of Emerging Trends in Computing and Information Sciences, Vol. 4, No. 1, January 2013, pp.152-166.

Dans cet article, nous avons présenté une architecture qui est très utile dans un système multimodal. Nous avons introduit un algorithme efficace, basé sur l'utilisation de la technique de pattern, pour le processus de fission. Ces patterns sont modélisés dans une base de connaissances. Ils facilitent l'exécution des tâches et l'optimisation du temps d'exécution.

Le troisième chapitre est un article qui a été publié dans le journal « Journal of Software Engineering and Applications » :

Atef Zaguia, Ahmad Wahbi, Moeiz Miraoui, Chakib Tadj, Amar Ramdane-Cherif « *Modeling Rules Fission and Modality Selection Using Ontology* », Journal of Software Engineering and Applications, Vol. 6, No. 7, July 2013, pp.354-371.

Dans cet article, nous avons justifié et argumenté l'utilisation de l'ontologie, comme le partage de la compréhension commune de la structure de l'information, la modélisation des connaissances et la réutilisation de celles-ci dans un domaine donné. Nous avons présenté en détail notre ontologie pour résoudre notre problème qui consiste à modéliser nos données, les rendre dynamiques, flexibles et faciles à mettre à jour. Nous avons aussi présenté notre architecture qui est capable d'identifier les différentes modalités de sortie et qui permet de représenter les sous-tâches élémentaires à travers ces modalités. Notre architecture a été modélisée par le formalisme des réseaux de Petri colorés et simulée par CPN-Tools.

Le quatrième chapitre est un article qui a été soumis dans « Journal on Multimodal User Interfaces» :

Atef Zaguia, Chakib Tadj, Amar Ramdane-Cherif « Context-Based method using Bayesian Network in a multimodal fission system ». Janvier 2014

Dans cet article, nous avons présenté une nouvelle méthode basée sur le contexte en utilisant le réseau bayésien pour résoudre le problème de l'incertitude durant le processus de fission dans un système multimodal. La méthode proposée permet de surmonter le problème d'ambiguïté ou d'incertitude. Un exemple concret a été illustré dans le but de montrer l'efficacité de la contribution en utilisant l'outil CPN-Tools.

Le cinquième chapitre est un article qui a été soumis dans « International Journal of Soft Computing and Engineering» :

Atef Zaguia, Chakib Tadj, Amar Ramdane-Cherif « Prototyping Using Pattern Technique and Context-Based Bayesian Network in Multimodal Systems». Janvier 2014

Dans cet article, nous avons présenté le prototypage d'une application multimodale selon notre architecture. Nous avons implémenté deux applications réelles pour valider nos approches. Nous avons montré que la solution proposée est applicable dans un environnement réel. Nous avons présenté deux interfaces: 1) une interface pour le contrôle d'un robot a été mise en œuvre pour valider le processus de fission par la technique de pattern stocké dans l'ontologie, et 2) une interface GPS a été mise en œuvre pour valider la méthode fondée sur le contexte en utilisant un réseau bayésien dans le cas d'incertitude ou d'ambiguïté.

Enfin, le sixième chapitre est consacré à la conclusion de cette thèse. Dans ce chapitre, nous dressons le bilan de notre recherche et nous donnons les perspectives en ce qui concerne la fission multimodale.

CHAPITRE 1

REVUE DE LA LITERATURE

1.1 Introduction

Depuis les années quatre-vingt, le développement rapide dans le monde des technologies de l'information a permis, de créer des systèmes qui interfèrent d'une manière harmonieuse avec l'utilisateur, ceci est dû à l'émergence d'une technologie appelée : l'interaction multimodale. Cette technologie permet à l'utilisateur de se servir de ces modalités naturelles (le parole, le geste, le regard, etc.) pour interagir avec la machine avec une expressivité plus riche et plus variée : ce sont les systèmes multimodaux.

Les systèmes multimodaux représentent une déviation remarquable de l'utilisation des systèmes classiques, comme les windows-icones, vers une interaction homme/machine, en offrant à l'utilisateur plus d'intuitivité, de flexibilité et de portabilité.

Le premier système multimodal a été créé en 1980 par Richard Bolt « Put-That-There » (Bolt, 1980b). Le système est équipé d'un microphone et d'un écran. Il permet de déplacer ou de modifier l'affichage d'objets situés sur l'écran, en utilisant la commande vocale accompagnée de pointages sur l'écran.

Depuis la création de ce système, une variété de systèmes multimodaux, équipés de diverses modalités, sont apparus. Ces systèmes ont permis aux utilisateurs, en particulier pour ceux qui ne peuvent utiliser un clavier ou une souris, les malvoyants, les utilisateurs équipés d'appareils mobiles, les utilisateurs avec handicaps, etc., de bien profiter des avantages de ces systèmes.

Dans ce qui suit, nous présentons la définition de la multimodalité et quelques exemples de systèmes multimodaux. Nous décrirons ensuite les principaux modules que nous retrouvons dans les systèmes multimodaux : « la fission » et « la fusion ».

1.2 Caractérisation des interactions multimodales

1.2.1 Le média

Dans le dictionnaire de la langue française (Robert, 2010), le média est un moyen de diffusion, de distribution ou de transmission de signaux porteurs de messages écrits, sonores, visuels (presse, cinéma, radiodiffusion, télédiffusion, vidéographie, télédistribution, télématique, télécommunication, etc.).

De façon générale, un média est défini comme un support physique permettant de transmettre une information.

Dans (Djendi, 2007) l'auteur nous a montré que la littérature offre plusieurs définitions du média et chaque auteur possède sa propre conception de média :

> **Une finalité purement matérielle** : le média est vu comme un capteur ou un effecteur d'un système informatique. Il est communément appelé dispositif d'entrée/sortie ;
>
> **Une finalité purement technique** : le média est considéré comme un procédé physique et/ou logiciel utilisable comme véhicule ou support d'information ;
>
> **Une finalité 'technico-humaine' a deux niveaux d'abstraction** : le média est défini comme un couplage d'un dispositif physique, aux qualités sensorielles humaines, et aussi comme un système représentationnel (Djendi, 2007, p.12).

1.2.2 La modalité

D'après (Djendi, 2007), la modalité est définie dans la littérature sur trois principales tendances :

1. La modalité est vue comme un processus informatique d'analyse et de synthèse, défini sur des ensembles de données d'entrée-sortie ;

2. La modalité est une structure des informations échangées, telle qu'elle est perçue par l'être humain ;

3. La modalité est une technique d'interaction dépendante :

   a. Des capacités sensorielles de l'utilisateur;

   b. Des dispositifs physiques ou logiques engagés dans l'interaction (Djendi, 2007, p.14).

Pour conclure, nous pouvons dire que la modalité désigne la substance de l'information et le média les supports ou les véhicules de l'information :

- média : microphone, écran, clavier, etc ;

- modalité : parole, vision, etc.

1.2.3 La multimodalité

La multimodalité est définie comme étant une coopération entre plusieurs modalités, afin d'améliorer la communication homme-machine.

D'après (Djendi, 2007) « la multimodalité représente le plus souvent, pour les informaticiens, la capacité d'un système interactif à utiliser plusieurs canaux de communication lors de l'interaction entre l'utilisateur et le système, et cela au cours d'une même session ».
Selon Dumas (Dumas, Lalanne et Oviatt, 2009), les systèmes multimodaux traitent deux ou plusieurs modes d'entrée combinés par l'utilisateur (comme la parole, le stylet, le toucher, le geste, le regard et le mouvement de la tête et du corps) d'une manière coordonnée avec le système multimédia de sortie. Ils représentent de nouvelles classes d'interfaces qui visent à reconnaitre les formes naturelles du langage et du comportement humain, et qui intègrent une ou plusieurs technologies basées sur la reconnaissance (par exemple la parole, le stylet et la vision).

La multimodalité réfère au processus dans lequel les différents dispositifs et les gens sont capables d'interagir par voie auditive, visuelle, gestuelle ou par le toucher (Pous et Ceccaroni, 2010).

Les systèmes multimodaux interactifs permettent aux utilisateurs d'interagir avec les ordinateurs, les machines, etc. à travers différentes modalités comme la parole, le geste, etc.

Nous présentons dans ce qui suit les 10 lois pour concevoir des systèmes multimodaux.

1.2.4 Les dix lois

Dans (Oviatt, 1999), l'auteure présente les 10 lois nécessaires pour guider la conception des systèmes multimodaux :

1ère loi : « Si vous construisez un système multimodal, les utilisateurs vont interagir d'une manière multimodale ».

D'après l'auteure, les utilisateurs favorisent une interaction multimodale à une interaction unimodale. Elle a confirmé cette loi par des statistiques qui montrent que 95% à 100% des utilisateurs ont préféré interagir d'une manière multimodale lorsqu'ils ont été menés à choisir entre l'utilisation en entrée de la parole ou le stylet.

Nous sommes en accord avec l'auteure, car si l'utilisateur était face à un système multimodal équipé d'un écran tactile et un microphone, il utiliserait toutes les modalités disponibles puisqu'elles lui faciliteraient l'atteinte de ses objectifs. Par exemple pour sélectionner des fichiers, il est plus facile de dire « sélectionne ça » et on pointe sur les fichiers que de dire « sélectionne le Fichiers A…K » ;

2ème loi : « la parole et le pointage sont des modèles multimodaux dominants».

En effet, l'utilisateur a tendance à choisir la commande vocale et le pointage malgré la disponibilité d'autres modalités. Ceci est probablement dû au phénomène naturel de ces commandes qui se rapprochent de la communication humaine ;

3ème loi : « les entrées multimodales impliquent des signaux simultanés ».
D'après l'auteure, « l'évidence expérimentale révèle que les signaux multimodaux ne se produisent pas souvent temporellement au cours d'une communication entre l'homme et la machine. Ainsi, les informaticiens ne doivent pas compter sur des signaux convenablement coïncidant afin de parvenir à créer des architectures multimodales efficaces. » ;

4ème loi : « la parole est le mode d'entrée primaire dans un système multimodal »
La parole est considérée comme une modalité auto-suffisante. Les autres modalités sont considérées comme des accompagnements redondants qui portent peu de nouvelles ou importantes informations ».

Parfois les modalités secondaires sont aussi considérées utiles lorsque le signal primaire est dégradé (par exemple dans un environnement bruyant). Dans un tel cas, les autres modalités pourraient fournir de l'information complémentaire lorsque la confiance dans la reconnaissance vocale est faible ;

5ème loi : « le langage multimodal ne diffère pas linguistiquement du langage unimodal »
« Le langage multimodal est différent et souvent plus simplifié que le langage unimodal. » ;

6ème loi : « l'intégration multimodale implique une redondance du contenu entre les différentes modalités ».
L'auteure suggère de ne pas s'appuyer sur des informations dupliquées lors du traitement de langage multimodal ;

7ème loi : « l'utilisation de plusieurs modalités permet de surpasser les erreurs commises par le module de reconnaissance ».

Pendant le processus de reconnaissance, il pourrait y avoir des erreurs qui influencent le fonctionnement du système. L'utilisation de plusieurs modalités en parallèle permet de surmonter les erreurs. Par exemple si l'utilisateur dit « Write four » et le système l'interprète « Write for » ; si l'utilisateur utilisait le stylo en parallèle et il écrivait « 4 » le problème de la reconnaissance et de l'interprétation serait réglé ;

8ème loi : « les commandes multimodales des utilisateurs sont intégrées d'une manière uniforme ».
L'auteure montre que d'après une étude récente, les utilisateurs adoptent soit un modèle d'intégration simultané ou un système d'intégration séquentiel lors de la combinaison des informations entrées par la voix et le stylet ;

9ème loi : « les différents modes d'entrées sont capables de transmettre du contenu comparable » ;

10ème loi : « renforcer l'efficacité est le but principal des systèmes multimodaux »
D'après l'auteure, les systèmes multimodaux ne peuvent pas souvent accroitre l'efficacité. Les avantages peuvent résider ailleurs, tels que diminuer les erreurs, augmenter la flexibilité et augmenter la satisfaction des utilisateurs.

1.2.5 Applications multimodales

1.2.5.1 Les évènements catastrophiques

Dans (Caschera et al., 2009) les auteurs présentent un système multimodal capable en cas de catastrophes naturelles d'aider les gens à trouver des endroits sécuritaires. Ce système permet de fournir des services d'urgence dans des situations critiques. Il permet également aux utilisateurs de s'enregistrer au système « RISCOM » en utilisant les appareils mobiles à l'aide de la voix et du toucher.

Les auteurs ont présenté trois scenarios pour expliquer le fonctionnement de leur système :

1. Le premier scénario consiste à s'enregistrer au système en utilisant la voix et le toucher.
   En premier lieu, l'utilisateur peut chercher le réseau social à partir de son téléphone en utilisant la voix « Réseau Social RISCOM ». Après l'affichage du résultat de la recherche sur l'écran de son téléphone, il défile les résultats avec son doigt, affiche et sélectionne le réseau désiré. Il peut ensuite dire / sélectionner « Enregistrer RISCOM » pour l'enregistrement.
   En cas d'urgence, l'utilisateur recevra sur son appareil mobile les instructions pour se rendre aux lieux sécuritaires. Une carte des lieux sera affichée et l'utilisateur peut utiliser la voix pour bien afficher la carte en disant « Agrandir » ;

2. Le 2ème scénario concerne la collecte des informations pour des statistiques. L'institut de sismologie envoie des questionnaires-textes aux téléphones des utilisateurs. Ces derniers répondent par la voix ;

3. Le 3eme scénario simule la situation après un tremblement de terre. Le point de contrôle envoie des alertes audio, des textes et des messages visuels vers les téléphones portables des utilisateurs, pour les informer sur les endroits sécuritaires les plus proches. En outre, au cas où ces derniers rencontrent dans leur chemin des cas d'urgence ou entendent des demandes de secours dans des bâtiments effondrés, ils peuvent envoyer des demandes d'assistance au point de contrôle en attachant des photos ou des vidéos pour clarifier la situation.

Le système multimodal ne peut cependant servir que les utilisateurs inscrits dans le réseau social « RISCOM » alors qu'en cas d'urgence il est impératif que l'assistance couvre toutes les personnes menacées pour sauver le maximum de vies. Nous pouvons par exemple avoir recours à un module supplémentaire qui permet de détecter les personnes équipées d'appareils mobiles mais qui ne sont pas inscrits à « RISCOM ».

1.2.5.2 Système de conception de salles de bain

Dans (Pfleger, 2004), l'auteur présente un système fréquemment utilisé dans la construction des maisons et notamment la conception de salles de bain. La Figure 1.1 montre l'interaction entre l'utilisateur et l'ordinateur.

Figure 1.1 Démonstration du système
Tirée de Pfleger (2004, p.4)

L'utilisateur, en utilisant un microphone et un stylet peut créer le design de la salle de bain.
« L'interface multimodale de ce système inclut à l'entrée la parole et le stylet. Ils sont utilisés de manière intuitive et intégrée. Le feedback est généré par la voix, le graphique et les expressions faciales de la tête parlante » (Pfleger, 2004, p.2).

Après la conception, il y aura une visualisation en 3D. L'interaction entre le système et l'utilisateur est conçue de façon à ce que le système assiste l'utilisateur de manière continue pendant la conception du plan de la salle de bain (Figure 1.2).

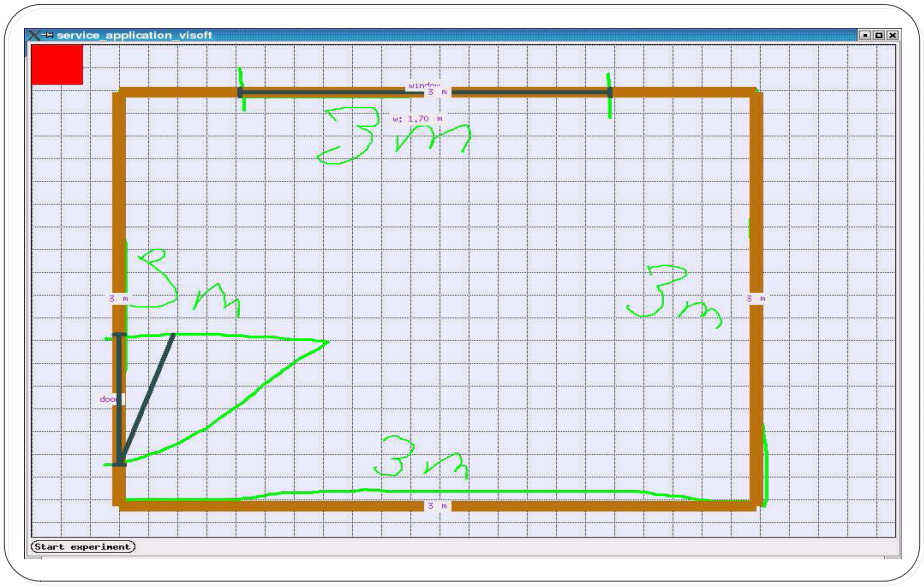


Figure 1.2 Exemple de capture d'écran de fonctionnement du système
Tirée de Pfleger (2004, p.4)

Cette figure présente les murs, la direction d'ouverture de la porte et la position de la fenêtre que l'utilisateur a créées en utilisant la voix et stylet.

1.2.5.3 Système multimodal utilisé en robotique

Dans (Giuliani et Knoll, 2008), l'auteur présente un exemple d'interaction multimodale personne - robot. Il décrit un robot muni de deux bras. Il reçoit des ordres d'une personne par la voix. Il peut manipuler ses deux bras pour prendre ou déplacer des objets situés sur la table, comme le montre la Figure 1.3.

Le scénario que les auteurs présentent dans (Giuliani et Knoll, 2008) est réservé pour la construction et l'installation de panneaux de signalisation de chemin de fer. C'est un exemple de coopération entre l'homme et le robot.

Figure 1.3 Exemple d'interaction multimodale entre un robot et un humain
Tirée de Giuliani et Knoll (2008, p.2)

Le rebot reçoit des ordres d'un humain pour déplacer des objets.

Ici, l'implémentation et le contrôle à distance d'un tel robot est très bénéfique surtout dans des endroits inaccessibles ou exposant la personne à un danger.

1.2.5.4 Enseignement assisté par ordinateur

Dans (Wang, Zhang et Dai, 2006), l'auteur propose un système multimodal qui utilise le stylet et la voix pour aider les enfants à apprendre la langue chinoise.

Le système présente un moyen d'apprentissage original pour les enfants car il aide ces derniers à apprendre tout en jouant.

1.2.6 Analyse et discussion

Les systèmes multimodaux améliorent l'accessibilité pour différents utilisateurs. La multimodalité permet l'amélioration de la reconnaissance et de la compréhension par l'ordinateur des commandes de l'utilisateur.

Nous remarquons que le nombre de modalités utilisé par toutes ses applications est limité à deux. L'augmentation du nombre de modalités sera un atout pour faciliter ou améliorer l'usage des différentes applications et de cibler un grand nombre de public.

Il est à noter que ces applications ne prennent pas en considération les erreurs dûes à des facteurs externes tels que le bruit, la luminosité de l'endroit, etc.

Pour surmonter ce problème, certains auteurs comme (Dey, 2001), (Efstratiou et al., 2001), (Hina et al., 2011) et (Atef Zaguia et al., 2010) ont proposé d'ajouter d'autres modules dans les systèmes multimodaux tel que « Context d'interaction ». Ce module, comme le montre la Figure 1.4, est composé de trois sous modules :

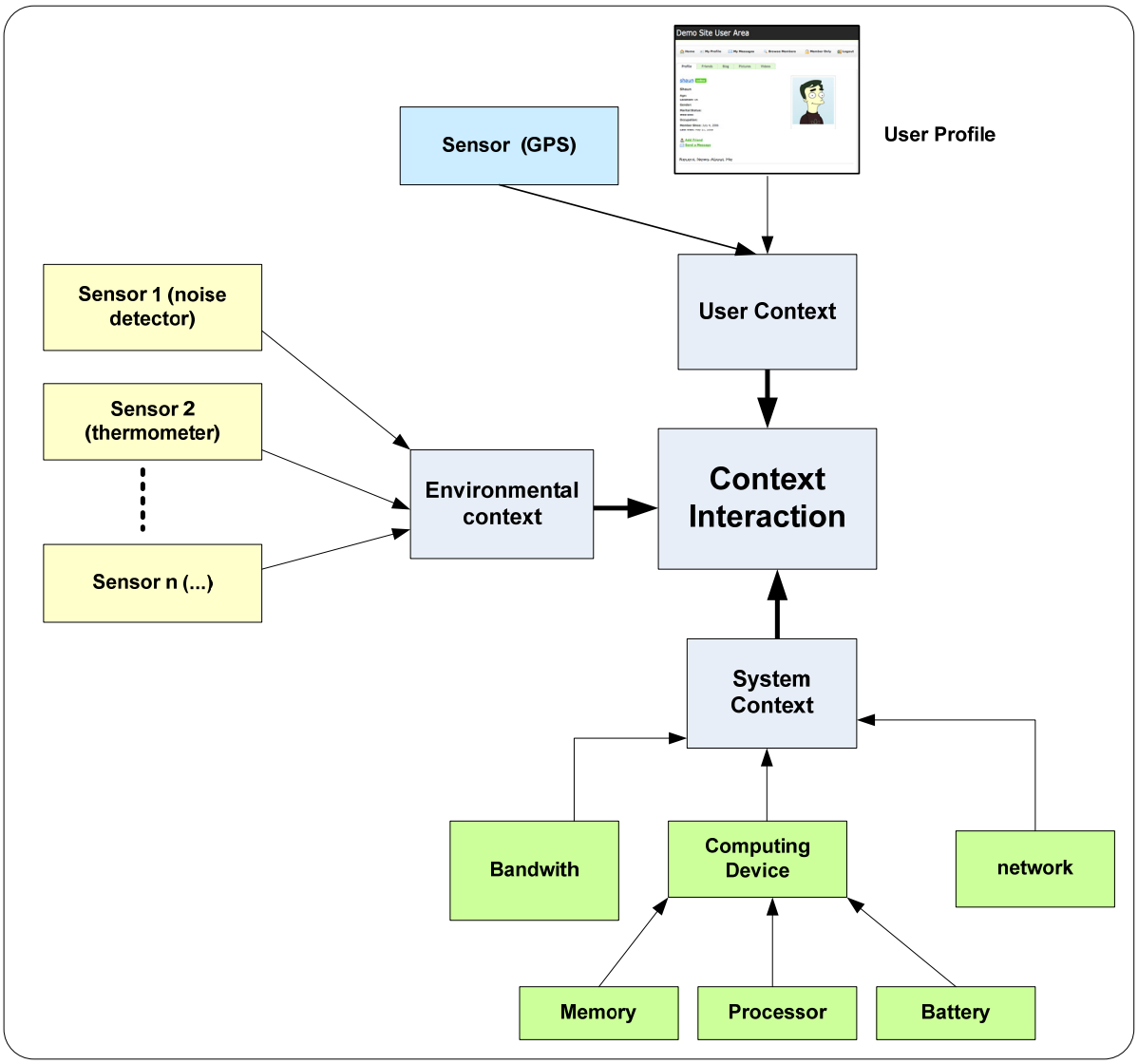


Figure 1.4 Module contexte d'interaction
Tirée de Atef Zaguia et al. (2010, p.7)

Profil utilisateur : Ce module permet de détecter l'emplacement et l'état de l'utilisateur. Il repère la capacité de l'utilisateur à utiliser certaines modalités. Par exemple le système désactive la modalité affichage s'il détecte que l'utilisateur est malvoyant et désactive la modalité commande vocale s'il détecte que l'utilisateur est dans une bibliothèque ;

Contexte de l'environnement : Ce module détecte l'état de l'environnement de l'utilisateur. Par exemple, la détermination du niveau de bruit. Il est entendu que l'utilisation de la modalité audio (entrée ou sortie) est affectée par cette information ;

Contexte système : La capacité et le type de systèmes que nous utilisons sont des facteurs qui déterminent ou limitent les modalités qui peuvent être activées.

1.3 Fusion/Fission

L'objectif de recherche dans la multimodalité consiste à développer un système flexible capable de manipuler plusieurs modalités. Cet objectif ne serait atteint que si nous réalisions la fusion de toutes les commandes venant des différentes modalités en entrée et la fission sur les modalités de sortie.

D'après la littérature, les modules *Fission* et *Fusion* sont cruciaux pour chaque système multimodal (Wahlster, 2003) comme le montre la Figure 1.5.

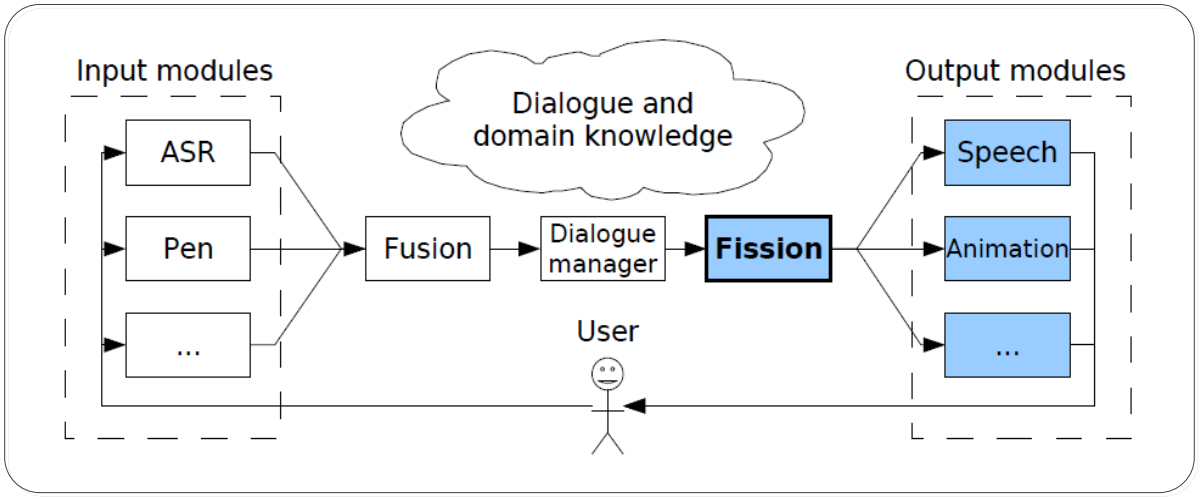


Figure 1.5 Architecture d'un système de dialogue multimodal
Tirée de Foster (2005, p.3)

1.3.1 La fusion

1.3.1.1 Définition de la fusion

C'est un processus de combinaison de commandes multimodales en entrée pour aboutir à une seule commande. (Djendi, 2007).

1.3.1.2 Présentation d'une architecture pour la fusion

Dans cette partie, nous présentons l'architecture du système multimodal cité plus haut « Enseignement assisté par ordinateur ».

Comme le montre la Figure 1.6, il y a deux modalités en l'entrée : parole et stylet. Lorsque des événements sont produits par les modalités à l'étape 1, il y aura reconnaissance des commandes de chaque modalité (étape 2). Les données sont envoyées au module principal de l'architecture pour la fusion (étape 3).

Dans cette étape, le système vérifie si les deux commandes sont équivalentes (par exemple l'utilisateur dit « Écrit 4 » et avec le stylet il écrit « 4 »)

- si oui, il n'y aura pas de fusion et la commande est envoyée au module « result » ;

- sinon, le module « fusion » effectue la fusion selon le contexte et la connaissance puis transfère la commande fusionnée au module « result».

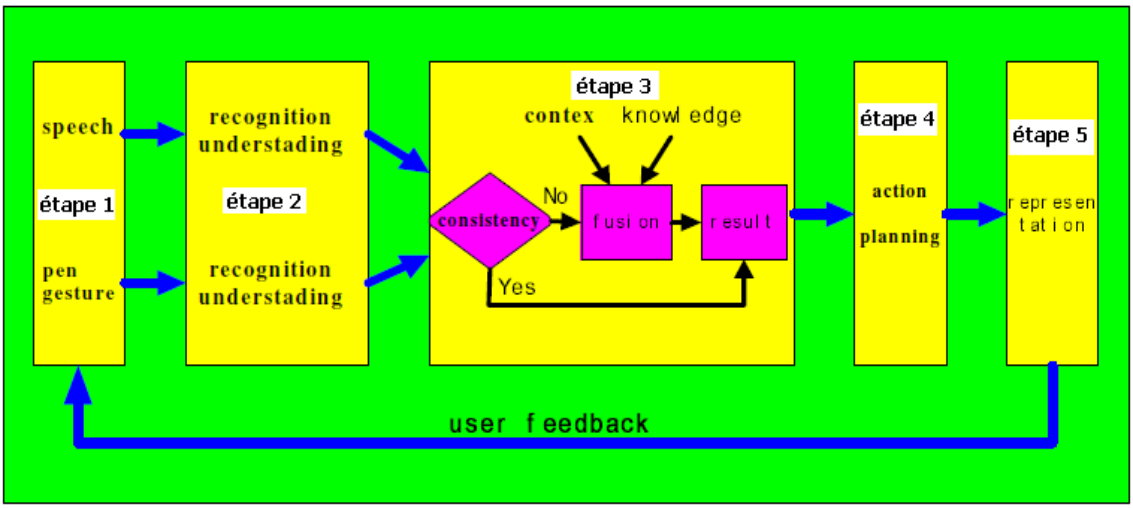


Figure 1.6 Architecture du système Enseignement assisté par ordinateurs
Tirée de Wang, Zhang et Dai (2006, p.2)

L'étape 4 correspond donc à l'étape de la planification de l'action selon le résultat, alors que l'étape 5 consiste à représenter l'information et à fournir un feedback à l'utilisateur.

1.3.1.3 Analyse de cette architecture

Cette architecture se limite à un nombre limité de modalités. Ceci représente un inconvénient pour étendre son utilisation à plusieurs modalités. De plus, les auteurs ne prennent pas en considération les fusions survenues antérieurement. Donc l'ajout d'une base de connaissances pour sauvegarder les commandes qui ont été fusionnées est nécessaire afin d'optimiser le temps de calcul du système.

1.3.2 La fission

1.3.2.1 Définition de la fission

D'après (Poller et Tschernomas, 2006), la fission c'est la partition de la sortie en tâches pour les différentes modalités.

D’après (Ertl, Falb et Kaindl, 2010), la fission c’est la manière de diviser les données importantes et les présenter sur des modalités de sortie.

Patrizia (Grifoni, 2009) a défini la fission comme « un processus qui considère les morceaux de l’information. Ils sont combinés sur les différentes modalités tout en trouvant une façon à les présenter et les structurer. ».

Foster (Foster 2002) définit la fission comme « le processus de réaliser un message abstrait à travers la sortie en fonction de combinaisons des modalités disponibles ».

D’après (Landragin, 2007), la fission multimodale est liée à la répartition de l’information entre plusieurs modalités. Le rôle principal de la fission multimodale est de déterminer le message qui sera généré avec chaque modalité.

Tous ces auteurs sont d’accord sur le fait que la fission est la manière de segmenter les données qui vont être présentées à l’utilisateur selon les modalités disponibles ainsi que le contexte.

Donc nous pouvons dire que l'objectif de la fission multimodale est de passer d'une présentation, indépendamment des modalités, à une présentation multimodale coordonnée et cohérente.

1.3.2.2 Les étapes de la fission

D’après (Grifoni, 2009), pour concevoir le processus de fission, on passe par trois principales étapes :

Sélectionner et structurer le contenu : Cette étape consiste à sélectionner et à organiser le contenu qui sera présenté ;

Sélectionner la modalité : Consiste à spécifier les modalités qui peuvent afficher ou présenter la commande ;

Coordination des sorties : Cette étape permet de coordonner les sorties pour chaque canal afin de créer une présentation cohérente.

Dans (Rousseau et al., 2006), les auteurs décrivent un modèle conceptuel « WWHT (What-Which-How-Then) » pour la conception d'un système multimodal et la présentation de l'information à la sortie. Ce modèle est basé sur les concepts « Quelle », « Lequel », « Comment » et « Donc/Alors » :

- quelle information à présenter ?
- quelle modalité ou combinaisons de modalités choisir pour présenter l'information ?
- comment présenter l'information en utilisant les modalités choisies ?
- comment manipuler l'évolution de la présentation résultante ?

Dans COMIC (Foster, 2005), Foster présente les étapes de la fission :

- sélection de contenu ;
- structuration ;
- sélectionner modalité (s) ;
- coordination de la sortie.

Pour expliquer d'avantage le processus de la fission, l'auteur dans (Poller et Tschernomas, 2006) a présenté un simple exemple : c'est de boire une tasse de café. Pour une personne, cette tâche n'est pas une action atomique. Le cerveau humain divise, au premier lieu, cette tâche en de sous-tâches appropriées. Par exemple, aller à la cuisine ou préparer la machine. Chaque sous-tâche est alors divisée en de nouvelles sous-tâches jusqu'à ce que les tâches deviennent atomiques. Par exemple, le nerf X envoie un signal au muscle Y. La Figure 1.7 résume cet exemple.

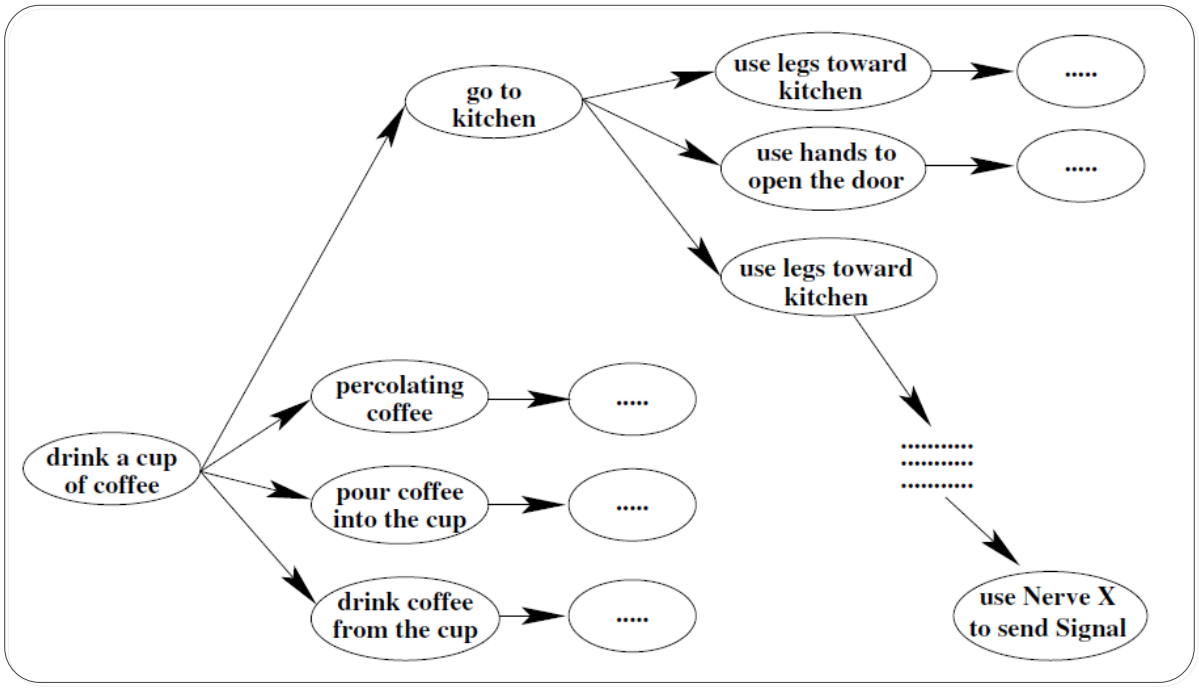


Figure 1.7 Une tâche divisée en sous-tâches par le cerveau humain
Tirée de Poller et Tschernomas (2006, p.6)

1.4 Projets Existants

1.4.1 SMARTKOM

Smartkom (Poller et Tschernomas, 2006) est un système multimodal qui combine en entrée la parole, le geste et la biométrie. En sortie la parole, le geste et des graphiques. Ce système permet un affichage visuel qui inclut du texte en langage naturel et un avatar parlant.

Tout au long de l'utilisation de ce système, l'utilisateur obtient un ensemble cohérent et une expérience agréable grâce à un agent d'interaction personnalisé (avatar), appelé Smartakus Figure 1.8.

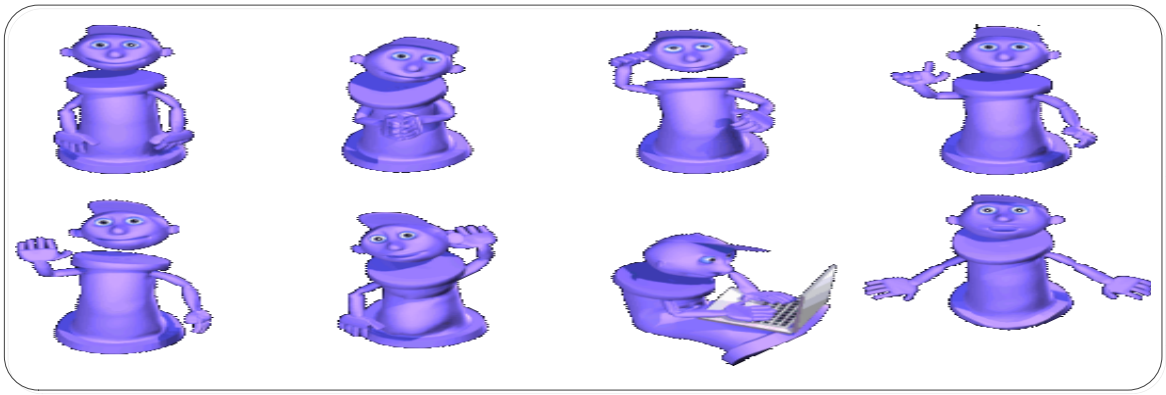


Figure 1.8 Quelques exemples de Smarkatus
Tirée de Poller et Tschernomas (2006, p.5)

Dans (Poller et Tschernomas, 2006), les auteurs ont présenté trois différents scénarios d'applications de SmartKom (Figure 1.9) :

SmartKom PUBLIQUE : Ce scénario représente un kiosque multimodal. L'utilisateur peut utiliser ce système pour scanner des objets, envoyer des courriels, effectuer des appels, etc. ;

SmartKom MAISON : Le système agit comme un système d'information intelligent à la maison. L'utilisateur sera équipé d'une tablette avec laquelle il peut contrôler la télévision,

avoir des informations concernant un programme donné d'une chaine TV, enregistrer des émissions, etc. tout en utilisant ses modalités naturelles ;

SmartKom MOBILE : Ce système se comporte comme un guide touristique ou un navigateur GPS pour un utilisateur équipé d'un PDA.

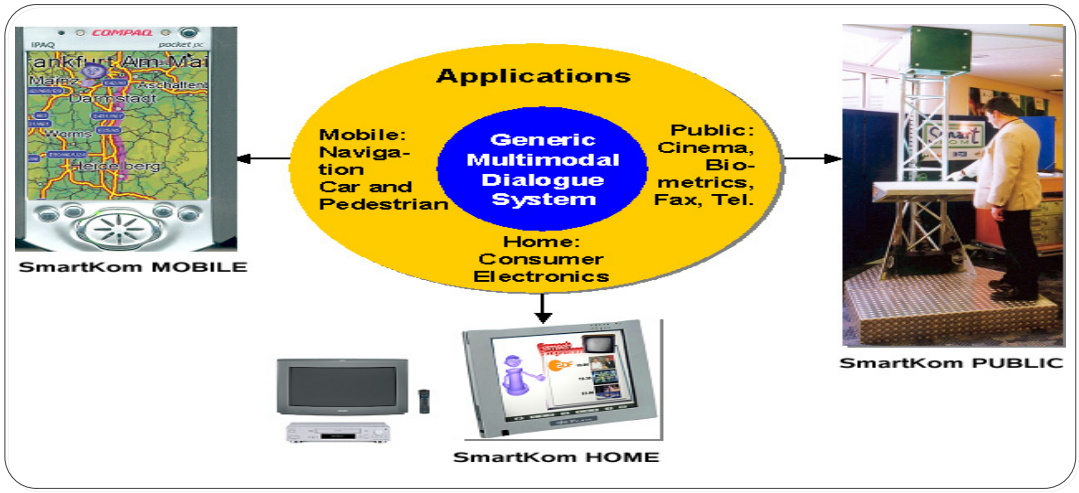


Figure 1.9 Applications SmartKom
Tirée de Poller et Tschernomas (2006, p.6)

L'architecture de ce système est composée de plusieurs modules comme le montre la Figure 1.10. Un système qui repose sur cette architecture est à « initiative mixte et coopérative ».

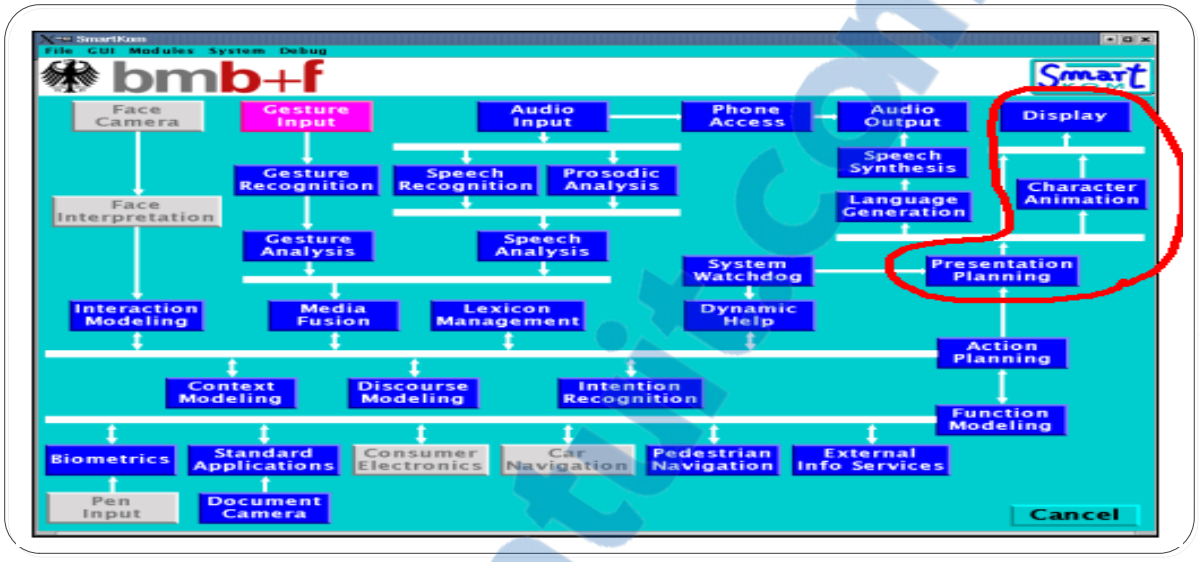


Figure 1.10 Les modules de SmartKom
Tirée de Poller et Tschernomas (2006, p.5)

Ce système agit comme un partenaire pour l'utilisateur puisqu'il essaye de reconnaitre et de réaliser les tâches que l'utilisateur veut accomplir.

La Figure 1.11 présente en détail les éléments impliqués dans la fission multimodale. La sortie pour le système (fission) est contrôlée principalement par les modules suivants :

Planificateur de présentation : « Presentation Manager » pour la (Figure 1.10) et « Presentation Planner » (Figure 1.11). Ce module définit ce que devrait être affiché pour l'utilisateur. Il gère ainsi la distribution des présentations sur les différentes modalités disponibles. Il décompose la présentation complexe en des tâches primitives et en même temps, il fait l'étape de la fission dépendamment des modalités disponibles. Cela signifie qu'il décide quelle partie de la présentation doit être instanciée comme les graphiques, le geste ou la parole en sortie. « Pour cela, il utilise plus d'une centaine de stratégies de présentation et tient compte du contexte du discours, du modèle de l'utilisateur et des conditions d'utilisation ». Pour les modalités qui utilisent le langage naturel (oral ou textuel) pour communiquer avec l'utilisateur, les tâches de présentations sont effectuées en texte en dehors du planificateur de présentation (Text Generation dans la Figure 1.11 et Language

Generation dans Figure 1.10). Le texte généré est ensuite synthétisé et envoyé vers le module « Display Manager » ;

« Display Manager » et « Character Animation Module » : Ces deux modules s'occupent de l'affichage d'une présentation cohérente soit pour SmarKatus (avatar) ou pour la synchronisation de la parole en sortie avec le geste et mouvement de l'avatar.

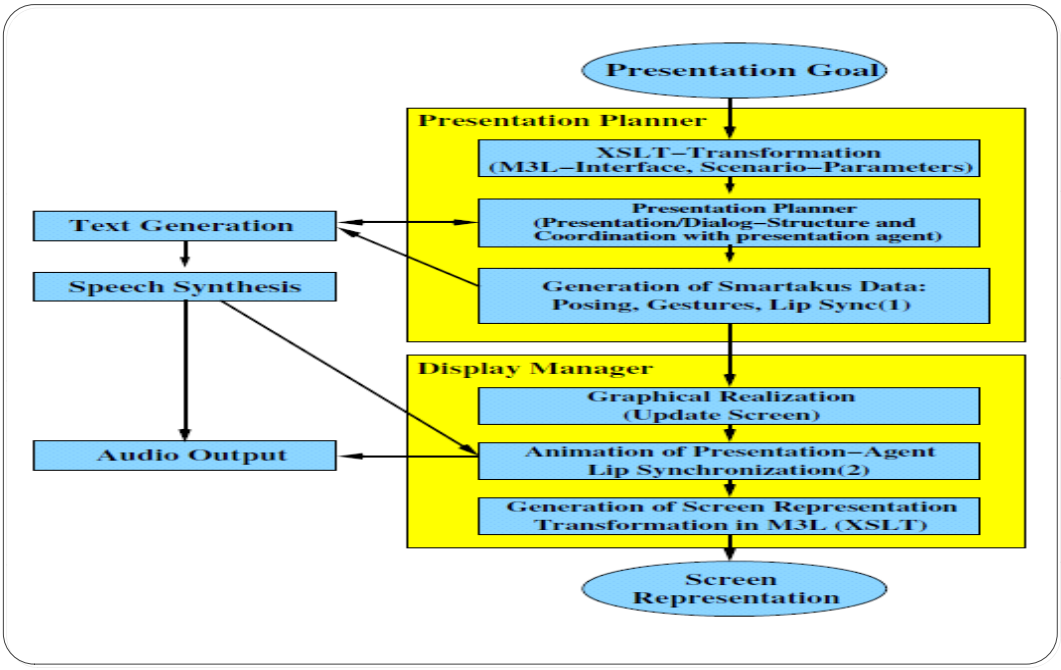


Figure 1.11 Système de fission de SmartKom
Tirée de Poller et Tschernomas (2006, p.8)

1.4.2 COMIC

C'est un logiciel qui aide à concevoir les salles de bains. La sortie de ce système est assurée par une représentation graphique de la salle de bain et une tête parlante (Figure 1.12). Une description détaillée de ce système a été présentée dans la section 1.2.5.2.

Dans (Foster, 2005), l'auteure présente une description technique de la sortie du système de dialogue multimodal COMIC. Elle décrit l'étape de la segmentation des données. Elle montre également comment préparer et exécuter en parallèle les segments générés.

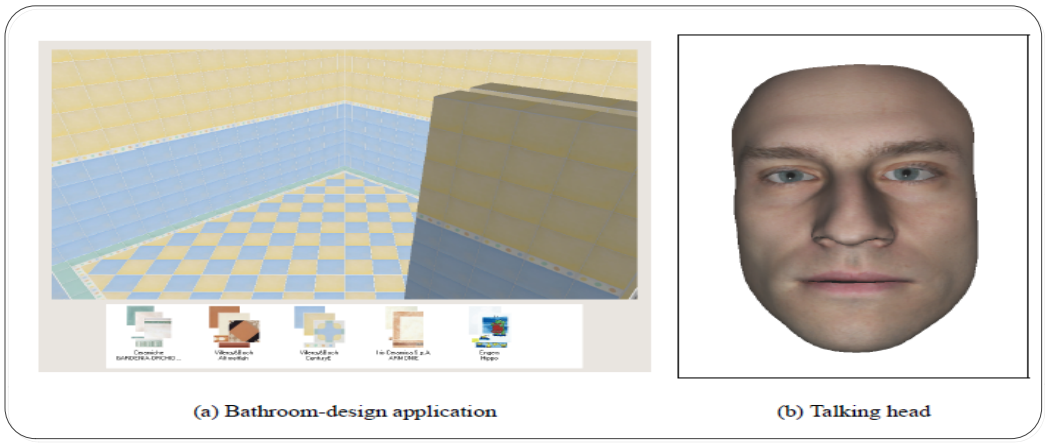


Figure 1.12 Interface de système COMIC
Tirée de Foster (2005, p.3)

Quand le module fission reçoit une entrée du module « Dialogue Manager », il sélectionne et structure un contenu multimodal pour créer un plan de sortie, en utilisant une combinaison des scripts et des segments de sortie générés automatiquement.

Les segments sont présentés par des classes abstraites comme le montre la Figure 1.13.

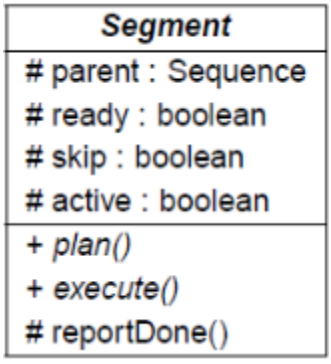


Figure 1.13 Class abstraite de segment
Tirée de Foster (2005, p.4)

Cette classe contient trois méthodes

- plan() : démarre la préparation des sorties ;
- execute() : produit les sorties préparées ;
- reportDone() : indique que la sortie est complétée.

Chaque segment contient des flags qui contrôlent son traitement :

- ready : ce flag est marqué (set) en interne une fois que le segment a fini toutes les étapes de préparation et il est prêt pour la sortie ;
- Skip : ce flag est marqué si le segment rencontre des problèmes durant la planification. Il indique que le segment sera ignoré au moment de la production de la sortie ;
- Active : ce flag est marqué par le segment parent et indique que ce segment doit être en sortie dès que possible.

Le diagramme dans la Figure 1.14 montre le fonctionnement de ces méthodes et de ces flags.

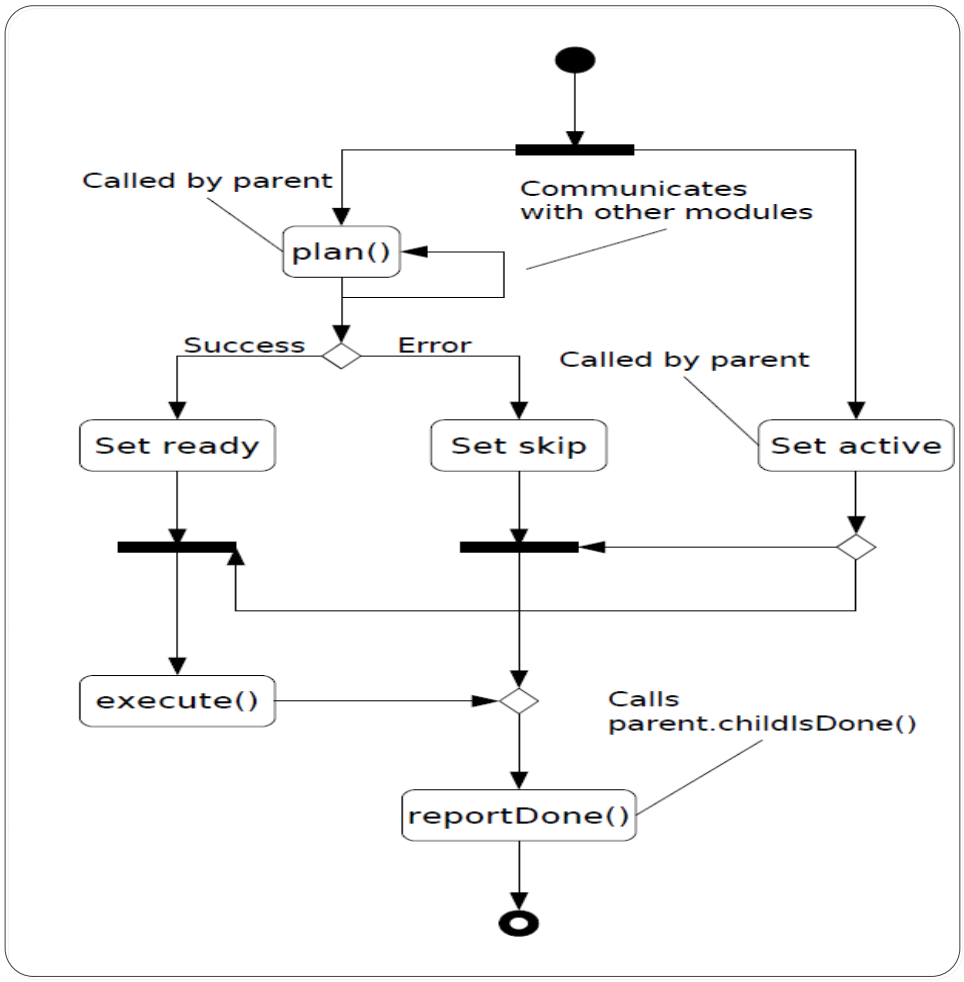


Figure 1.14 Fonctionnement de la préparation et segmentation de la sortie
Tirée de Foster (2005, p.)

1.4.3 Simulateur de conducteur

Le système multimodal présenté dans l'article (Benoit et al., 2009) est un « Simulateur de conducteur ». Ce système a comme entrée des données vidéo et des signaux biologiques et en sortie des audio sonores, des messages visuels et des vibrations du volant (Figure 1.15). Le système réagit en temps réel pour les situations de fatigue et de stress du conducteur.

Figure 1.15 Système multimodal : Simulateur de conducteur
Tirée de Benoit et al. (2009, p.7)

La fission proposée est très simple. Des mesures sont prises en temps réel. Celles-ci sont comparées à des valeurs de référence. Des alertes sont alors générées si certains seuils sont atteints. Par contre dans notre cas, nous traitons une commande complexe et le but du module fission consiste à la subdiviser en sous-tâches élémentaires et présenter chaque tâche élémentaire par la modalité (s) disponible (s) et adéquate (s).

1.4.4 Conclusion

L'évolution du domaine de recherche concernant l'exploration de différentes modalités (surtout la commande vocale) et la création de nombreuses interfaces novatrices ont permis de donner une grande importance à l'interaction multimodale.

D'après la revue de la littérature, nous avons noté l'important rôle des deux modules « fusion » et « fission » dans les systèmes multimodaux. Le module « fusion » permet de fusionner les commandes venant de différentes modalités, en une seule commande compréhensible par la machine. Le module « fission » permet de segmenter la commande générée par la machine en des tâches plus simples selon les modalités de sortie disponibles et appropriées.

Cependant, la plupart des systèmes multimodaux utilisent un nombre très limité de modalités de sortie et leurs architectures sont spécifiques aux applications ciblées. Dans certains cas, ils utilisent une base de connaissances statique ; dans d'autres cas, ils utilisent des scenarios prédéfinis pour réaliser la fission. Aussi, la plupart des systèmes multimodaux étudiés focalisent, si ce n'est exclusivement, sur le processus de la fusion. Ce point est supporté par : « il n'y pas beaucoup de recherche effectuées concernant la fission multimodale parce que la plupart des applications utilisent un nombre limité de modalités de sortie, donc des mécanismes de sortie simple et direct sont souvent utilisés » (Costa et Duarte, 2011) et par « la fission multimodale est un sujet de recherche qui n'est pas souvent abordé dans la communauté scientifique » (Perroud et al., 2012).

Donc, notre travail consiste à développer un système multimodal qui serait capable de manipuler plusieurs modalités et de présenter des architectures plus avantageuses pour les modules « fission ». Nous adaptons une nouvelle solution pour le module de fission en modélisant des patterns qui explorent les différentes modalités de sortie et les différents scenarios possibles; et par la création d'une base de connaissances qui contient ces patterns.

Il est à noter que la plupart des systèmes multimodaux présentés dans la littérature n'ont pas mentionnée des solutions pour surmonter le problème d'ambiguïté ou incertitude durant le processus de fission ou fusion.

Dans la section suivante nous définissions les patterns et l'ontologie en général et nous présentons notre méthode pour la résolution d'incertitude.

1.5 Pattern

Définition du pattern :
Dans (Alexander, C. et al, 1977) le pattern « c'est de décrire un problème qui se produit souvent dans un environnement puis de trouver une solution détaillée de telle sorte qu'on peut utiliser cette solution un million de fois ».

« Le pattern aide à transporter les connaissances (knowledge) et fournit une solution à un problème qui survient dans un certain contexte » (Grone, 2006).

« Une configuration structurelle récurrente qui résout un problème dans un contexte, ce qui contribue à l'intégrité de certains ensembles, ou un système, qui reflète une certaine valeur esthétique ou culturelle. » (O.Coplien et Harission, 2005).

D'après les auteurs (O.Coplien et Harission, 2005), un pattern c'est une règle : le mot « configuration » doit être lu comme « une règle à configurer » mais elle est plus qu'une simple règle : c'est un genre spécial de règle qui contribue à la structure globale du système et qui coopère avec d'autres patterns pour créer une structure émergente.

1.6 Raisonnement incertain

Dans la littérature, il existe plusieurs méthodes qui traitent le problème d'ambigüité ou d'incertitude des données. Nous présentons dans ce qui suit les deux plus répandues : la logique floue et les réseaux bayésiens.

1.6.1 Logique floue

« La logique floue est une extension de la logique booléenne créée par Lotfi Zadeh en 1965 (Zadeh, 1965) en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans

la vérification d'une condition, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue confère une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes » (Buckley et Eslami, 2002), (Mukaidono, 2001).

Elle est utilisée dans des domaines très variés tels que l'aide à la décision, le diagnostic (domaine médical, etc.), les bases de données (objets flous et/ou requêtes floues), la reconnaissance de forme, etc.

La logique floue est basée sur des variables floues dites *variables linguistiques* à valeurs linguistiques dans l'univers du discours U (le référentiel). Chaque valeur linguistique constitue alors un ensemble flou de l'univers du discours.

Les systèmes à logique floue utilisent une expertise exprimée sous forme d'une base de règles du type: Si….Alors… Ils sont caractérisés par une fonction d'appartenance (Hibou, 2006) :

$$\mu: x \in U \rightarrow \mu(x) \in [0,1] \qquad (1.1)$$

1.6.2 Réseau Bayésien

Un réseau bayésien est un graphe dans lequel les nœuds représentent des variables aléatoires, et les liens des influences entre variables. Le graphe est acyclique : il ne contient pas de boucle. Les flèches représentent des relations entre variables qui sont soit déterministes, soit probabilistes. Chaque nœud X_i a une distribution de probabilités conditionnelles $P(X_i \mid parents(X_i))$ qui quantifie les probabilités d'apparition des valeurs d'un nœud en fonction de ses parents.

Les réseaux Bayésiens sont utilisés dans plusieurs domaines tels que le diagnostic (médical et industriel), l'analyse de risques, la détection des spam et le data mining.

Ils décrivent les relations causales entre variables par un graphe. Les nœuds dans le graphe représentent les variables aléatoires et les arcs la relation entre ces nœuds.
Son but est de calculer la probabilité :

$$P(cause|effet) = \frac{P(effet|cause)xP(cause)}{P(effet)} \quad (1.2)$$

À titre d'exemple, supposons qu'une alarme de maison ne se déclenche que s'il y a un tremblement de terre ou un cambriolage.

Pour construire un réseau bayésien pour cet exemple, nous devons définir en premier lieu le graphe du modèle puis on définit les tables de probabilité de chaque variable, conditionnellement à ses causes (Figure 1.16).

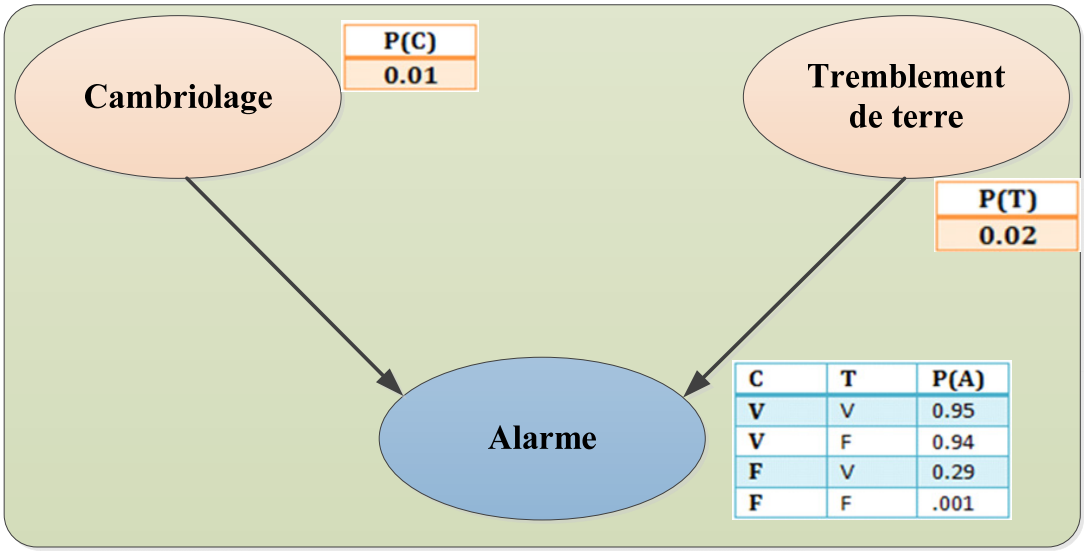


Figure 1.16 Réseau sémantique Bayésien de l'alarme

Dans cet exemple, Alarme correspond à effet ; Cambriolage et tremblement de terre sont les causes. Si on veut calculer la probabilité qu'il y ait cambriolage et déclenchement de l'alarme, il faut calculer la probabilité P(C=V/A=V), où C et A correspondent à cambriolage et alarme respectivement.

1.6.3 Conclusion

L'utilisation d'un réseau bayésien dans la fouille de données « data mining » s'est avérée être une solution efficace et performante pour surmonter le problème d'ambiguïté (Daouadji, 2011). En plus, vu la nature incertaine des données, Bayes a « probabilisé tout ce qui est incertain et même des phénomènes non aléatoires. » (Saporta, 2006). De ce fait, nous adoptons cette méthode pour l'implémenter dans le processus de fission. Dans ce qui suit, nous montrons comment cette méthode est utilisée dans la fouille de données « data mining », puis nous montrons l'adaptation de cette méthode dans notre projet de recherche pour résoudre le problème d'incertitude ou d'ambigüité dans le processus de fission.

1.7 Fouille De Données (Data Mining)

En général, il existe deux méthodes de recherche : 1) la première est une recherche traditionnelle qui se fait sur des mots clés. D'une part, à une information stockée est associée un ou plusieurs mots clés. D'autre part, une demande de l'utilisateur est formulée sous forme de mots clés. Le processus de recherche se résume alors à mettre en correspondance les mots clés de la requête de l'utilisateur et les mots clés des informations stockées. 2) La deuxième méthode de recherche est plus "intelligente" car elle est capable de prendre en compte le sens des mots clés, plutôt que de les considérer comme une simple suite de caractères.

L'idée générale est d'aller au-delà d'une simple mise en correspondance stricte entre les mots clés de la requête de l'utilisateur et les mots clés des informations stockées, pour fournir des résultats habituellement ignorés. C'est là où la notion d'ontologie intervient, en organisant sous forme de graphe un ensemble de concepts (les mots-clés) par des relations sémantiques (ex: est-une-sorte-de, est-analogue-à, est-synonyme-de, etc.). C'est une façon technique de simuler la connaissance.

Par exemple, si nous faisons une recherche sur le mot-clé « lion », le système utilise l'ontologie créée généralement par un expert. Il découvre une relation de généralisation avec

le concept "carnivore", et donc présenter, des informations associées avec le mot clé « carnivore ».

Pour conclure, un tel moteur de recherche nous donne l'impression de "comprendre" ce qu'est un lion et qu'il soit même capable de prendre l'initiative d'augmenter les résultats sur un sujet fortement connexe.

La découverte de ressources est un domaine de recherche particulièrement important de nos jours puisqu'il sert à fournir un sens aux données. Il est aussi passionnant, puisqu'il se base sur la capacité des ordinateurs à apprendre et à s'améliorer avec le temps.

Les données seules n'ont presque aucune valeur. Alors que la quantité de données augmente de manière exponentielle, les gens sont en fait assoiffés de connaissance. La connaissance est obtenue par la compréhension des données. Plus on a de données, plus il est difficile d'en tirer de la connaissance.

L'utilisation d'un graphe bayésien sémantique est importante pour effectuer une recherche efficace (Daouadji, 2011). Au départ, il extrait de la requête les mots clés. Chaque mot clé est relié à un ou plusieurs concepts avec une certaine probabilité et chaque concept désigne une ressource ; le lien entre les concepts et les ressources sont des liens sémantiques (Figure 1.17).

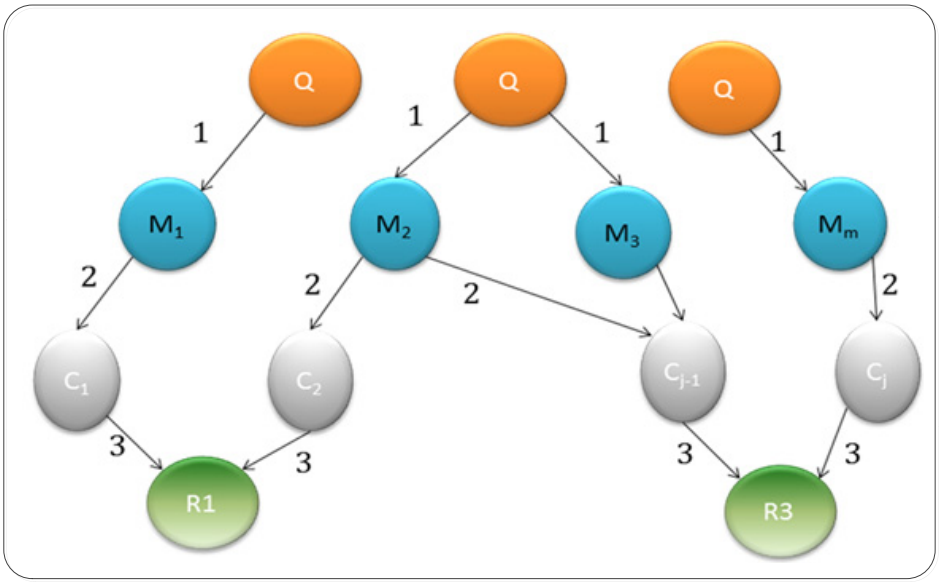


Figure 1.17 Réseau bayésien pour une requête donnée

Dans la Figure 1.17, les numéros 1, 2 et 3 représentent l'extraction des mots clés, la relation probabiliste avec les concepts et les liens sémantiques avec les ressources, respectivement.

Pour chacun des concepts, la probabilité a posteriori des mots clés de la requête est calculée en utilisant l'équation (1.3) suivante :

$$P(C|M) = \frac{\mathrm{P(M|C)P(C)}}{\mathrm{P(M)}} \begin{cases} C: les\ concepts \\ M: les\ mots\ clés\ de\ la\ requête \end{cases} \quad (1.3)$$

$avec\ M = \mathrm{m_1, m_2 \ldots, m_n} \qquad n: nombre\ des\ mots\ clefs$

Où

- $P(C|M)$: probabilité a posteriori ;
- $\mathrm{P(M|C)}$: vraisemblance ;
- $\mathrm{P(M)}$: évidence ;
- $\mathrm{P(C)}$: probabilité a priori.

À la fin, on compare les probabilités qu'on a calculées pour chaque concept et on retourne la ressource liée au concept dont la probabilité est maximale.

Exemple :
Supposons que nous avons deux concepts Animal et Auto. Assumons que nous ayons la situation suivante :

- C1 {Animal} et C2 {Auto} ;

- mots clés M = {vitesse, prix, sauvage, âge, année de construction} ;

- probabilité de vraisemblance (On l'estime):

  - $P\,(Vitesse|C1) = P\,(Vitesse|C2) = 0.5$;
  - $P\,(Prix|C1) = 0.1$;
  - $P\,(Prix|C2) = 0.9$;
  - $P\,(Sauvage|C1) = 0.99$;
  - $P\,(Sauvage|C2) = 0.01$;
  - $P\,(Age|C1) = 0.70$;
  - $P\,(Age|C2) = 0.30$;
  - $P\,(A.Cons|C1) = 0.01$;
  - $P\,(A.Cons\,|C2) = 0.90$.

Supposons que l'utilisateur exécute la requête suivante : {Tous les animaux sauvages et leurs vitesses}.

Pour résoudre ce problème, nous construisons le réseau de Bayes (Figure 1.18) et nous calculons les probabilités pour les deux concepts :

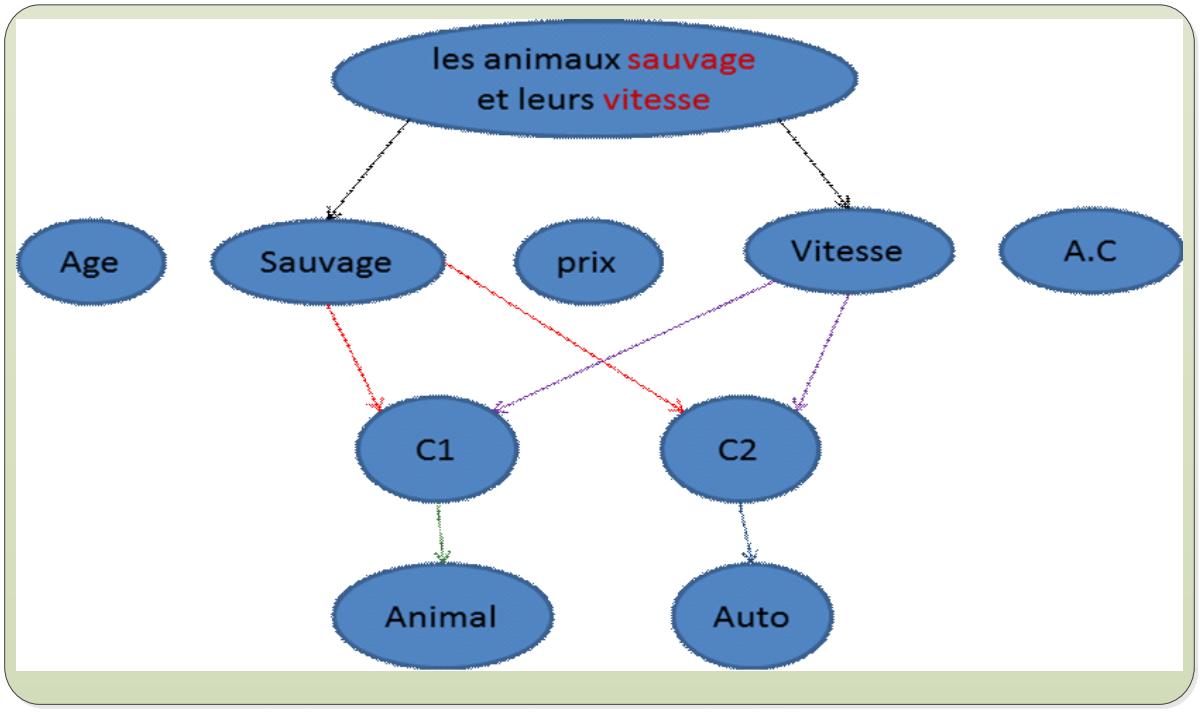


Figure 1.18 Réseau bayésien pour les concepts {Animal, Auto}

- $P(C1|S,V) = P(C1)\ x\ P(S|C1)\ x\ P(V/C1) \quad = 1\ x\ 0.5\ x\ 0.99\ =\ 0.495$

- $P(C2|S,V) = P(C2)\ x\ P(S|C2)\ x\ P(V|C2) \quad = 1\ x\ 0.5\ x\ 0.01\ =\ 0.005$

Notons que $P\ (C1|S,V)\ >\ P(C2|S,V)$. Ceci nous permet de conclure que le concept Animal est la solution la plus adéquate pour la requête.

N.B. $P(C1) =\ P(C2) = 1$ pour donner le même poids aux deux concepts.

En conclusion, l'utilisation des graphes sémantiques bayésiens a abouti à des résultats précis. Le réseau de Bayes semble être performant et permet d'effectuer une recherche efficace.

En nous inspirant de cette méthode, nous présentons une méthode basée sur le contexte avec l'utilisation d'un réseau bayésien pour résoudre le problème d'ambiguïté ou d'incertitude.

Dans notre recherche, nous avons représenté l'adaptation du réseau bayésien avec des informations contextuelles telles que temps, le statut de l'utilisateur, la température, la localisation, etc., par l'équation :

$$Con_i \underset{j=1}{\overset{m}{\rightarrow}} (C_j, P_j),\ i = 1, \cdots, n \tag{1.4}$$

Où n et m représentent le nombre de contextes et le nombre des concepts ambigus respectivement, Con = contexte, C = concept et P = probabilité. Le symbole $\rightarrow$ représente la relation entre le contexte et concept. Chaque contexte est connecté avec un ou plusieurs concepts avec une probabilité correspondante. Cette méthode est détaillée dans le chapitre quatre.

1.8 Les ontologies

Le terme "Ontology" est extrait de la philosophie. Ses racines remontent aux études métaphysiques d'Aristote sur la nature de l'être et du savoir. Sa signification est « l'étude de l'être en tant qu'être, indépendamment de ses déterminations particulières. C'est-à-dire l'étude des propriétés générales de ce qui existe » (Robert, 2011).

À partir des années 70, l'écriture des "ontologies conceptuelles" a commencé. Le terme ontologie a été adopté par plusieurs programmeurs ((Wilensky, 1978), (Lehnert, 1978), etc.) dont le but est de structurer l'information en données compréhensibles par l'ordinateur.

Les ontologies sont devenues par la suite omniprésentes dans des domaines variés tels que l'intelligence artificielle, le Web sémantique, le génie logiciel, l'informatique biomédicale et l'architecture de l'information.

L'ontologie, souvent connue comme le modèle d'un domaine de connaissance, est la description formelle et explicite des concepts d'un domaine bien particulier. D'après la

littérature, il existe plusieurs définitions de l'ontologie. Gruber dans (Gruber, 1991) a défini l'ontologie comme :

> dans le contexte du partage de connaissances, j'utilise le terme ontologie pour signifier une spécification d'une conceptualisation. Autrement dit, une ontologie est une description (comme une spécification formelle d'un programme) des concepts et les relations qui peuvent exister... Elle a certainement un sens différent du mot utilisé dans la philosophie. (Gruber, 1991, p.2)

Pour Costa, Laskey et AlGhamdi l'ontologie est

> « une représentation formelle explicite des connaissances sur un domaine d'application. Cela comprend:
>
> a) les types des entités qui existent dans le domaine ;
>
> b) les propriétés de ces entités ;
>
> c) les relations entre les entités ;
>
> d) les processus et les événements qui se produisent avec ces entités.
>
> » (Costa, Laskey et AlGhamdi, 2006).

Uschold et Gruninger (Uschold et Gruninger, 1996) ont défini l'ontologie comme des connaissances partagées pour certains domaines d'intérêt. Sowa (Sowa, 1995) définit l'ontologie comme un produit d'une étude sur des choses qui existent ou peuvent exister dans certains domaines.

Une des définitions de l'ontologie la plus connue est celle de Gruber (1993, p.18) « Une ontologie est la spécification d'une conceptualisation d'un domaine de connaissances ».

Cette définition s'appuie sur deux dimensions :

- une ontologie est la *conceptualisation* d'un domaine, «c'est-à-dire un *choix* quant à la manière de décrire un domaine. » (wikipedia, 2011) ;

- c'est par ailleurs la *spécification* de cette conceptualisation, «c'est-à-dire sa *description formelle*. » (wikipedia, 2011).

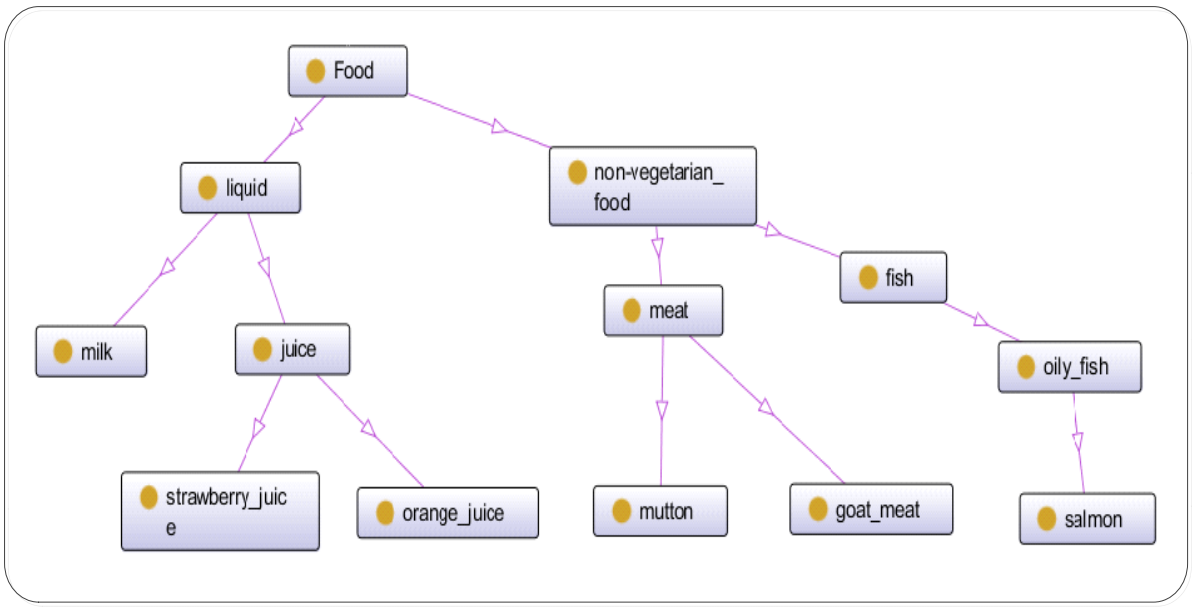


Figure 1.19 Exemple d'ontologie

En informatique, une ontologie, moyen efficace pour représenter un ensemble de connaissances sous une forme utilisable par une machine, est un ensemble structuré de concepts qui sont organisés dans un graphe (Figure 1.19) dont les relations peuvent être :

- des relations sémantiques ;

- des relations de composition et d'héritage.

Pour conclure, l'atout principal de l'ontologie est l'interopérabilité sémantique. Cela signifie que les systèmes ne vont pas juste échanger les données entre eux dans un format donné (par exemple la chaîne de caractères " Canada ") mais aussi doivent également avoir le même sens pour les deux parties (c'est un pays).

Dans ce que qui suit, nous présentons les éléments essentiels de l'ontologie :

C: Classes (concepts) : la Figure 1.20 montre un exemple des concepts ;

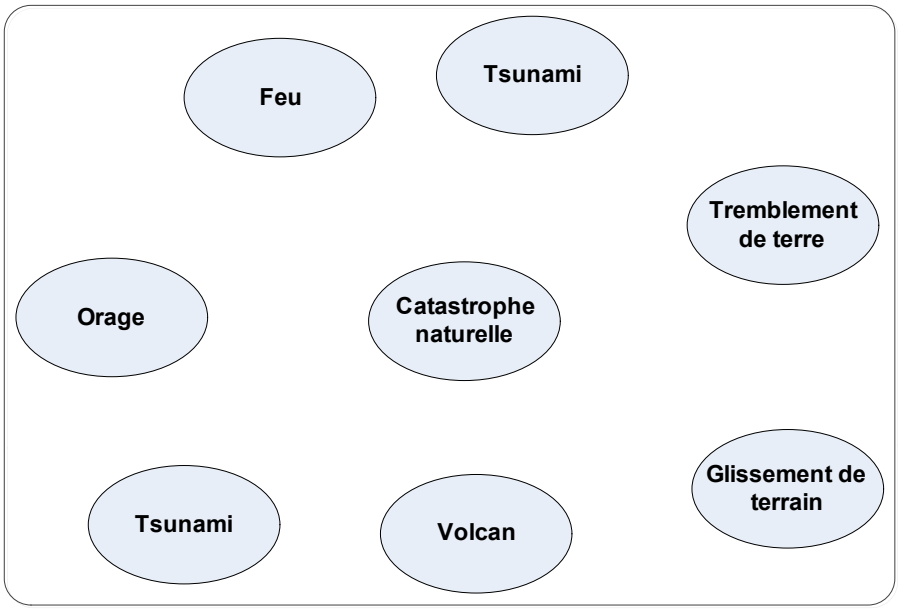


Figure 1.20 Exemple de concepts

P: Attributs (propriétés) : la Figure 1.21 montre un exemple d'attributs ;

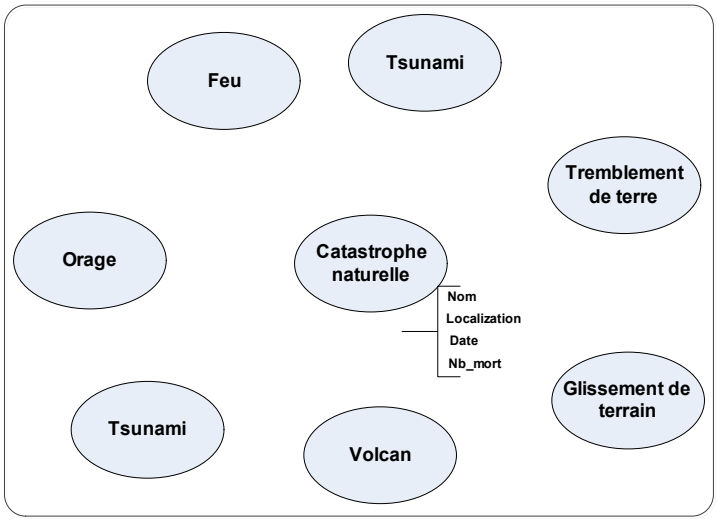


Figure 1.21 Exemple d'attributs

H: Structure hiérarchique (is-a, part-of relations), la Figure 1.22 montre une structure hiérarchique ;

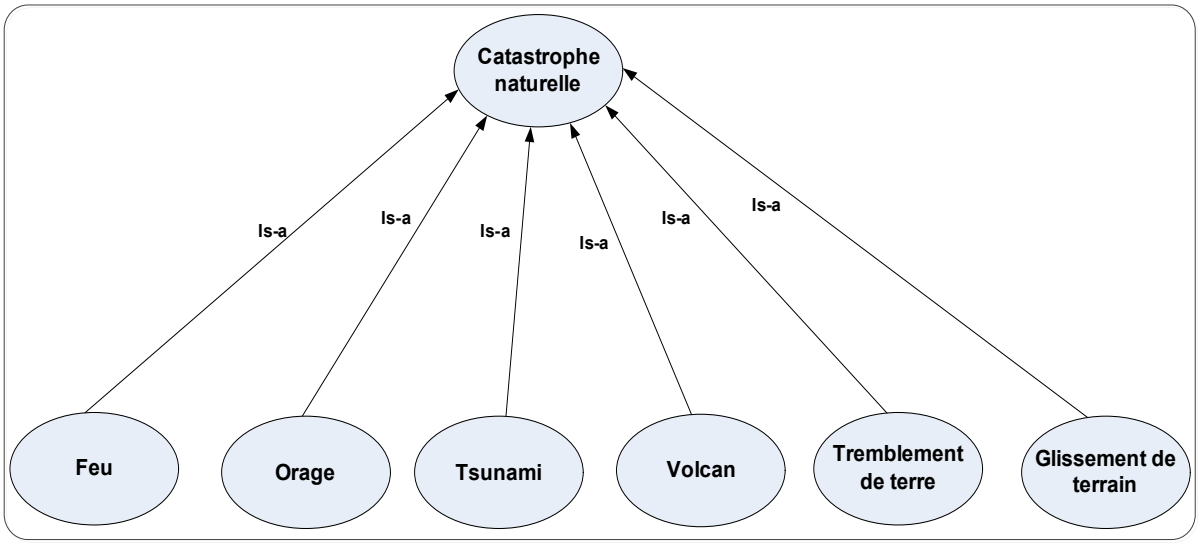


Figure 1.22 Exemple de structure hiérarchique

I: Instances: la Figure 1.23 montre un exemple d'ontologie avec des instances.

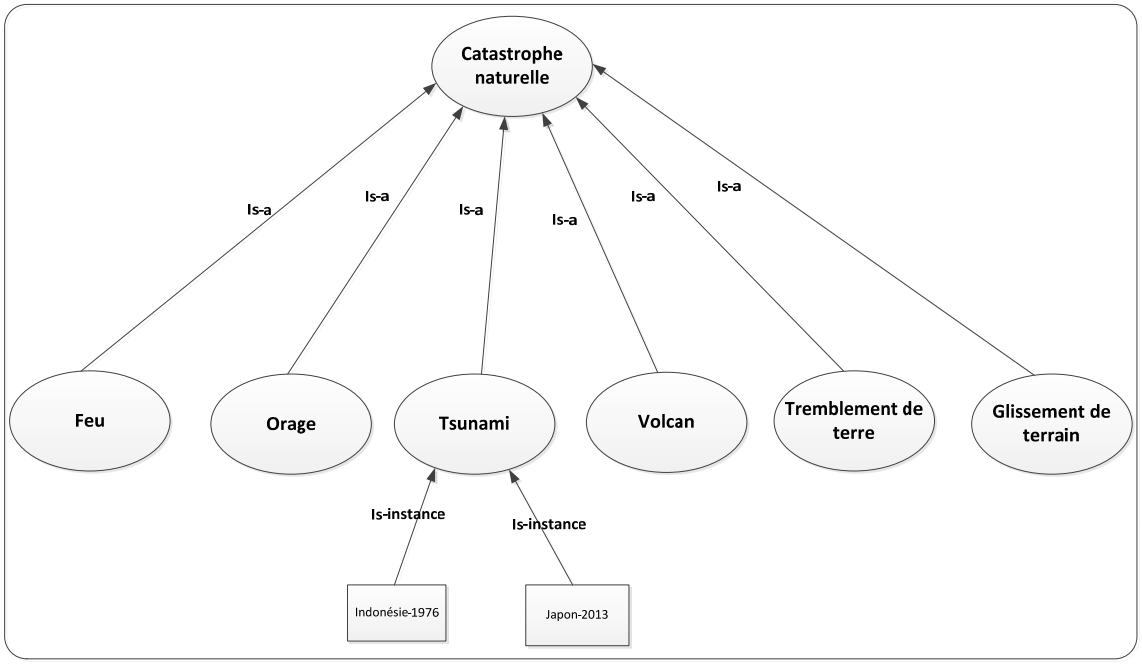


Figure 1.23 Exemple d'ontologie avec des instances

Mais récemment, plusieurs auteurs ((Ding et Peng, 2004), (Gu, Pung et Zhang, 2004)) ont mentionné des limitations des formalismes d'ontologies puisque ces dernières ne peuvent pas représenter l'incertitude ou des données incomplètes. « L'appréciation de la limite du formalisme des ontologies accroît, puisque celles-ci ne peuvent pas représenter l'incertitude. Il y a un accroissement de la demande des utilisateurs pour des ontologies avec le pouvoir d'exprimer l'incertitude. » (Costa, Laskey et Laskey., 2005).

Considérons par exemple « Washington est située à l'est des États-Unis ». Dans cet exemple, Washington réfère à une ville aux États-Unis et non au premier président des États-Unis ou l'équipe de basket. L'ontologie standard énumère alors plusieurs possibilités pour le sens du mot « washington », mais elle n'a pas la capacité de vérifier sa plausibilité relative dans un contexte donné.

Donc traiter l'incertitude est essentiel pour l'interopérabilité (capacité de deux ordinateurs de nature différente de travailler ensemble et de communiquer), le partage et la réutilisation des connaissances.

1.9 Conclusion

Dans ce chapitre nous avons montré les faiblesses des travaux présentés dans la littérature et les raisons de l'utilisation des réseaux bayésiens pour surmonter l'incertitude. Cette partie a été détaillée dans les sections 1.2.6, 1.4.4 et 1.6.3. Ensuite nous avons défini les technologies que nous avons adoptées pour résoudre notre problématique.

La revue de la littérature présentée a été essentielle pour comprendre toutes les notions requises : l'interaction multimodale, la définition du processus de fission que nous voulons mettre en œuvre, la définition des composants de raisonnement capables de comprendre l'environnement, les langages de la représentation des connaissances et de la description afin de modéliser, d'une manière sémantique et plus naturelle, l'architecture et les systèmes présents dans l'environnement.

Dans le chapitre suivant, nous présentons notre architecture pour le système de fission multimodal et nous introduisons notre algorithme basé sur l'utilisation de la technique de pattern.

CHAPITRE 2

MULTIMODAL FISSION FOR INTERACTION ARCHITECTURE

Atef Zaguia[1], Chakib Tadj[1], Amar Ramdane-Cherif[2],Ahmad Wahbi[1,2]

[1]MMS Laboratory, Université du Québec, École de technologie supérieure

1100, rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3 Canada

[2]LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France

This article is published in the Journal of Emerging Trends in Computing and Information Sciences, Vol. 4, No. 1, January 2013, pp.152-166.

Résumé

Depuis les années quatre-vingt, le développement rapide dans le monde des technologies de l'information a permis de créer des systèmes qui interfèrent avec l'utilisateur d'une manière harmonieuse. Cela est dû à l'émergence d'une technologie connue sous le nom interaction multimodale.

Cette technologie permet à l'utilisateur d'utiliser des modalités naturelles (parole, geste, etc.) pour interagir avec la machine dans un environnement informatique plus riche. Ceux-ci sont appelés des systèmes multimodaux. Ces systèmes représentent une déviation remarquable de l'utilisation des systèmes conventionnels, tels que windows-icons, à une interaction homme-machine, plus de naturel et la flexible.

En général, ces systèmes intègrent une interface multimodale en entrée et en sortie. Via l'interface de sortie, le système devrait être en mesure de choisir parmi les modalités disponibles, celles qui répondent aux meilleures contraintes environnementales. Il devrait également être en mesure d'interpréter une commande complexe et le diviser en sous-tâches élémentaires et de les présentés à travers les modalités de sortie: C'est la fission multimodal.

Notre travail spécifie et développe des composants de fission pour une interaction multimodale et présente un algorithme efficace de fission en utilisant les patterns.
Mot clés : fission multimodal, pattern, interaction homme-machine.

Abstract

Since the eighties, the rapid development in the world of information technology has made possible to create systems that interfere with the user in a harmonious manner. This is due to the emergence of a technology known as multimodal interaction.

This technology allows the user to use natural modalities (speech, gesture, eye gaze, etc.) to interact with the machine in a richer computing environment. These are called multimodal systems. These systems represent a remarkable deviation from using conventional systems, such as windows-icons, to a human-machine interaction, providing to the user more naturalness, flexibility and portability.

Generally, these systems integrate multimodal interface in input and output. Via the output interface, the system should be able to choose among the available modalities, those that meet the best environmental constraints. It should also be able to interpret a complex command and divide it into elementary sub-tasks and present them in the output modalities: it is called multimodal fission.

Our work specifies and develops fission components for a multimodal interaction and presents an effective fission algorithm using patterns, when various output modalities (audio, display, Braille, etc.) are available to the user.

Keywords: multimodal fission, pattern, human-computer interaction.

2.1 Introduction

Since the advent of computers, one of the biggest challenges in informatics has always been the creation of intelligent systems that enable transparency and flexibility of human-machine/machine-machine interaction (Sears et Jacko, 2007) (Alm, Alfredson et Ohlsson, 2009).

In our days, computers and intelligent systems have become increasingly ubiquitous. Users are looking for systems that are easy to use and intelligent.

Communication plays a primordial role in our common life. It allows humans to understand each other and to be connected as individuals or as independent groups. This communication is done across several natural modalities as speech, gestures, gaze and facial expressions.

Humans have a highly developed ability to transmit ideas between each other and to react in an appropriate way. This is done thanks to the sheared information.

But without external intervention, machines do not understand our language, do not understand how the world works and cannot collect information about a given situation. Researchers aim to always satisfy user's needs and provide systems that are smarter, more natural and easier to use.

Therefore various efforts have been geared toward the creation of systems that facilitate the communication between humans and machines and to allow to the user to make use of multimedia devices relying on natural modalities (sight, speech, gesture, etc.) to communicate or exchange information with applications, machines, etc.

These systems receive inputs from the sensors and gadgets (camera, microphone, etc.) and they make the interpretation and understanding of these inputs or multimodalities. A known example of such systems is Bolt system (Bolt, 1980a) "Put that there" where it used the gesture and speech to move objects. These systems generally comprise a multimodal input interface and a multimodal output interface. Via the output interface, the system should be able to choose among the available modalities, those that satisfy the best of environmental constraints, functional requirements of the task and user preferences. Several multimodal applications have been created ((Beinhauer et Hipp, 2009), (Caschera et al., 2009) and (Robert, Khaled et Tharam, 2005)).These systems represent an effective solution for users, particularly for those who cannot use a keyboard or a mouse, the partially sighted users, the users who are equipped with mobile apparatuses, the users who are disabled, etc., to use their natural modalities (word, gesture, look, etc.) to interact with the machine with a richer and more various expressiveness what they call the multimodal systems. Therefore the multimodal systems ameliorate accessibility for different users.

Each multimodal system (Meng et al., 2009) is based on tow essential components (Figure 2.1): fusion (Zaguia et al., 2010b), (Wehbi et al., 2011), (Xu et al., 2009) and fission (Costa et Duarte, 2011)), ((Ertl, Falb et Kaindl, 2010).

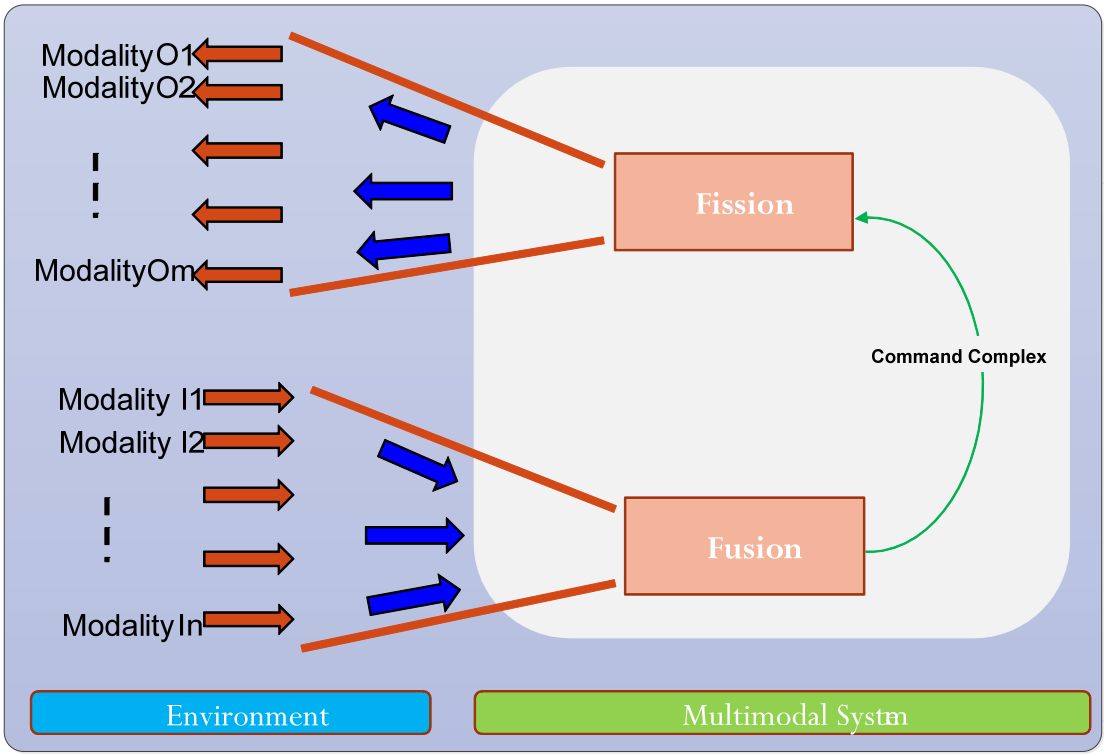


Figure 2.1 General architecture of multimodal system

Fusion usually refers to a process that combines events at the entrance to understand request of a user in his environment and to achieve a single but complex command.

Using the fission process, the system interprets a complex command, divides it into elementary sub-tasks and presents these sub-tasks as elementary output modalities. For instance, if a robot receives a command "Move an object A to a position (X,Y)", the system subdivides this complex command to:

a. Move toward "A" using mobility mechanism,

b. Take "A"using manipulator,

c. Head to position (XY) using mobility mechanism,

d. Drop "A" using manipulator.

The mobility mechanism and manipulator are services related to the output modalities.
In our work, we focus specially on 1) the services connected to the output: Multimodal Fission and 2) the creation of multimodal interaction system.

The rest of this paper is organized as follows. Section II presents our research problematic. Section III takes note of other research works that are related to ours. Section IV discuses the modalities selection and multimodal fission system. Section V presents a simulation for a scenario. The paper is concluded in section VI.

2.2 Challenges and proposed solution

Our main objective is to develop an expert system, capable of providing services to different multimodal applications. The system receives a complex command, subdivides it into elementary sub-tasks and presents them to output modalities. Here, we enumerate some challenges that need to be addressed and the proposed solutions in order to develop our system.

a. What are the modules required for the architecture of a fission system? Here we will specify, define and develop all necessary components of the system. We will also show how they communicate ;

b. How do we represent the multimodal information? We will model semantically the environment. To achieve our goals around the fission component, we create an context sensitive architecture able to manage multiple distributed modules and automatically adapts to dynamic changes of the context of interaction (user, environment, system) (Zaguia et al., 2010a) ;

c. How do we perform the fission process? We will introduce an algorithm that describes the fission mechanism. It includes the rules of fission and the rules of selection of output modalities ;

d. The fourth problem how to validate our architecture (formalism)? We focus more closely on the design, specification, construction and evaluation of our fission architecture.

2.3 Related work

In general, modality refers to the mode of interaction between man and machine for input and output data. With the use of traditional computing, human-computer interaction is limited to a traditional mouse, keyboard and screen.

Multimodality is an effective solution that enriches the human-machine communication. It allows a) a more flexible interaction between the user and the machine and b) a use of natural modalities to interact with the machines.

Current researches try to find solutions to subdivide the complex commands into elementary subtasks and present them to the output modalities. Many studies have addressed this problem. The system presented in (Benoit et al., 2009) is a multimodal system "driver simulator". This system has data video and biological signal as inputs and audio sound, visual messages and wheel vibration as outputs. The system reacts, in real time, to the situations of fatigue and stress of the driver. In (Foster, 2005), the author presents a system commonly used in the construction of houses and especially the design of bathrooms. The interface of this multimodal system includes in input speech and pen. They are used in an intuitive and integrated way. The feedback (output) is generated by voice, graphics and facial expressions of the talking head (Foster, 2005).

After the design, there will be a 3D visualization. The interaction between the user and the system is designed in a way that the system supports the user in a continuous manner during the design of the bathroom. In (Poller et Tschernomas, 2006), SmartKom is a multimodal system that combines the inputs speech, gesture and biometrics and manage the outputs speech, gesture and graphics. This system provides a visual display that includes the natural language text and a speaking avatar. Throughout the use of the system, the user gets a consistent and enjoyable experience through personalized interaction agent (avatar), called Smartakus. In the paper the authors present three different scenarios of SmartKom applications:

SmartKomPUBLIC: This scenario represents a multimodal kiosk. The user can use the system to scan objects, send emails, make calls, etc ;

SmartKomHOUSE: The system acts as an intelligent information system at home. The user is equipped with a tablet which controls the television, can have access to information about a given TV station programs, record programs etc. while using natural modalities ;

SmartKomMOBILE: The system behaves as a tour guide or a GPS navigator for a user with a PDA.

Most multimodal systems studied in the literature use only two modalities and their application specific architectures are targeted. So most of the multimodal systems studied focus on the fusion, but no advantage for fission. In some cases, they use a static database or in other cases, they use predefined scenarios to achieve fission.

In (Benoit et al., 2009), the proposed fission system is very simple. They test if the values entered are in some intervals and through this test, the system will generate alerts. In our case, we have a complex command and the goal of the fission module is to subdivide it into sub-tasks. Every elementary task will be presented in the available output and adequate modality (ies).

2.4 Modalities selection and multimodal fission system

We present first the definitions found on multimodal fission in the literature.

Poller (Poller et Tschernomas, 2006), defined the fission as "the partitioning of a presentation into tasks for different media, called multimedia fission". Foster (Foster, 2002) defined it as the process of realising an abstract message through the output based on the combination of available modalities. For Landragin (Landragin, 2007), multimodal fission is related to the distribution of information across multiple modalities. The main role of the multimodal fission is to determine which message will be generated with each modality. The objective of multimodal fission is to move from an independent presentation of modalities to a coordinated and coherent multimodal presentation.

In general, a fission process is presented in Figure 2.2. It consists of 3 essentials modules:

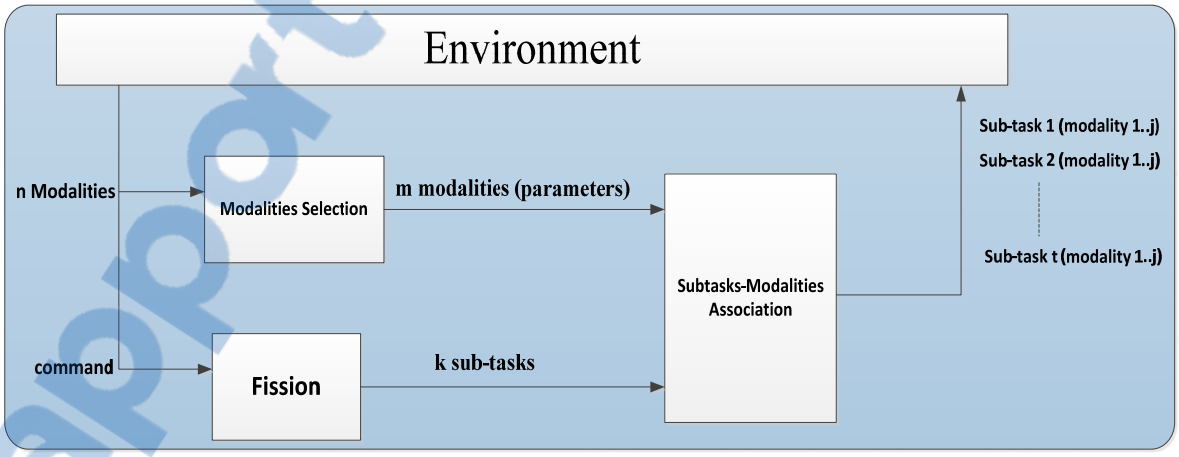


Figure 2.2 General view of the fission process

« Modalities selection »: The purpose of this module is to select which modalities can be used according to the state of the environment. This module is detailed in section 2.4.2 ;

« Fission »: performs the fission as described earlier. The input of this module is the command and the output the elementary subtasks ;

« Subtasks-Modalities Association»: its role is to associate for each subtask (output for fission module) to the modalities selected by the module « Modalities selection ». This module is detailed in section 2.4.2.

2.4.1 Multimodal fission architecture

In this section, we describe the proposed approach and the modules involved in the design and implementation of the architecture of the multimodal fission. The architecture is able to interact with multiple applications as shown in Figure 2.3.

For instance, suppose Application 1 is a system that controls a remote robot. As shown in Figure 2.3, the user generates a command using a phone or a computer (stage 1), the application (stage 2) sends the command via internet or social network to our multimodal system. The system receives the command (Stage 3), performs the fission and sends the elementary sub-tasks to the robot (Stage 4 and 5).

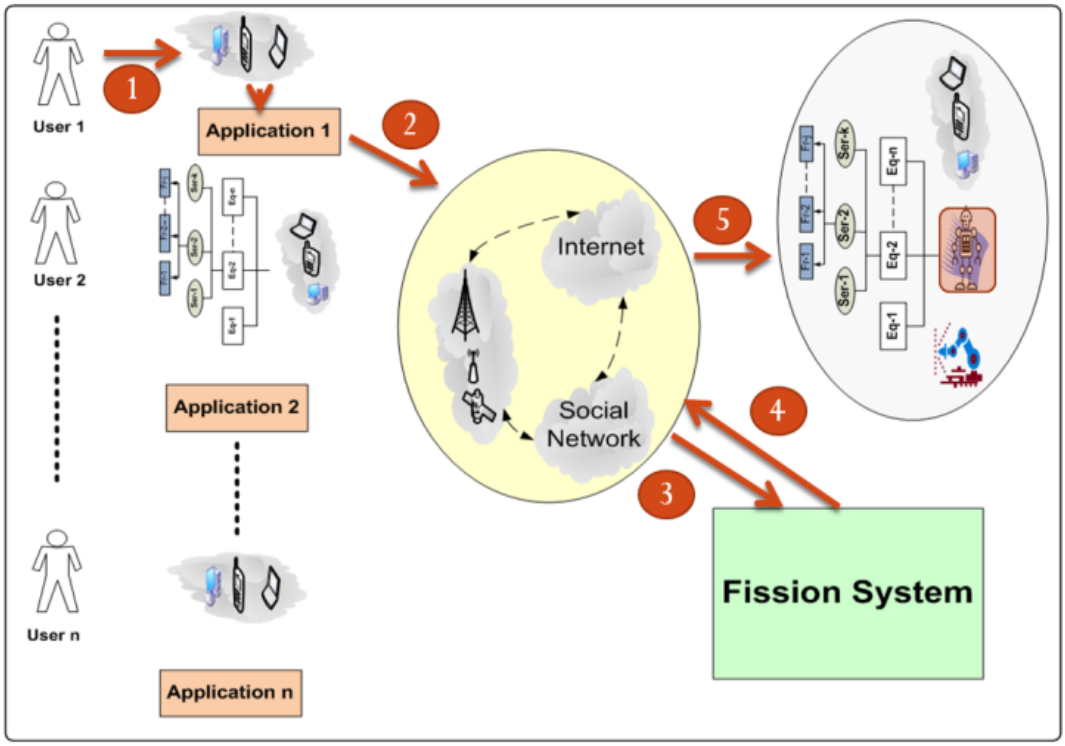


Figure 2.3 Interaction of our system with several applications

The proposed system architecture illustrated in Figure 2.4 is modular and distributed. It contains five main modules (Figure 2.4).

Detection/Interaction: This module will interact with the environment to allow modules "Fission process" and "Scenario selection component" to achieve fission. It detects any variation in the environment, for instance the change of the noise level that affects the selection of the audio modality ;

Fission process: represents the fission rules/algorithm necessary to realise the fission ;

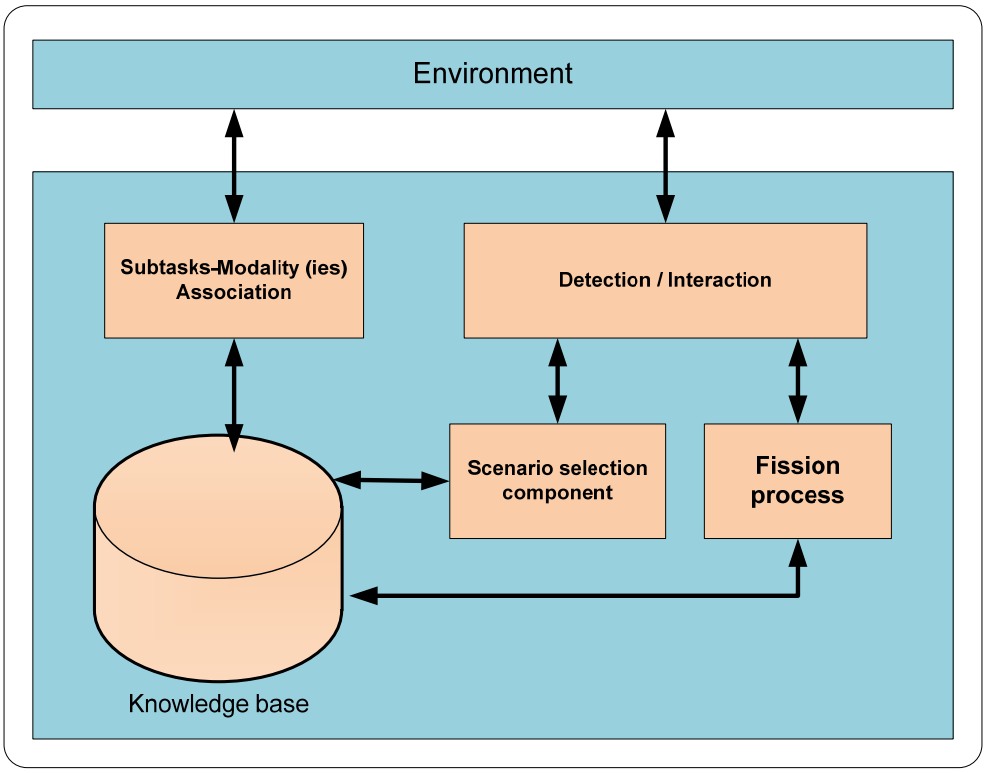


Figure 2.4 Multimodal fission system architecture

Subtasks-Modality (ies) Association: using patterns, this module allows us to select scenarios occurred previously and stored in the knowledge base ;

Modality selection: This module allows selection of the appropriate modalities available for each sub-task ;

Knowledge base: describes the environment, the modality patterns and the patterns of scenarios that occurred previously. It is described briefly in 2.4.3.

A detailed architecture of the system is shown in Figure 2.5. It contains six main modules: "Parser", "Fission", "Grammatical Analysis", "Feedback", "Synchronization" and " Pattern of Modality Selection."

These modules communicate together using XML files (Wyke, Rehman et Leupen, 2002) to exchange information. They are loaded onto a computer, robot, or any device that can communicate via the Internet, social networking, etc.

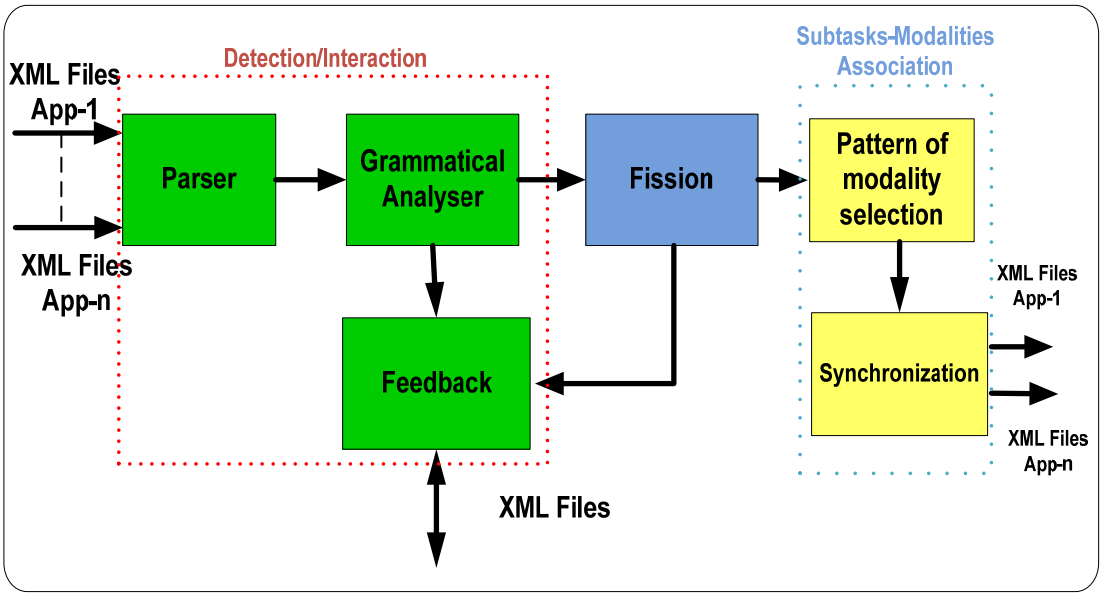


Figure 2.5 Framework of the multimodal fission

The system is composed of the following elements:

Parser: this module has as input XML files from different applications (for instance a robot control system or GPS system). Its role is to extract important information from the XML file (modality available, context, command... etc.) usable by the fission ;

Fission: based on the parameters of each modality and taking into account the rules of fission, this module determines whether the fission is possible and presents in the output the sub-tasks of each command ;

Grammatical Analyser: this module aims to check if the command is grammatically correct;

Feedback: this module corrects the errors made by the user or by the recognition module by sending a feedback to the user ;

Synchronization: this module is designed to synchronize between the outputs for each modality ;

Pattern of Modality Selection: selects the appropriate modalities for each sub-task.

2.4.2 Modality Selection and interaction context

Before we select the appropriate modality (ies) for every sub-task, we need to select the available modality using the interaction context.

In multimodal systems, context plays an important role to select the appropriate modalities. The selection module is called "interaction context". This module is shown in Figure 2.6.

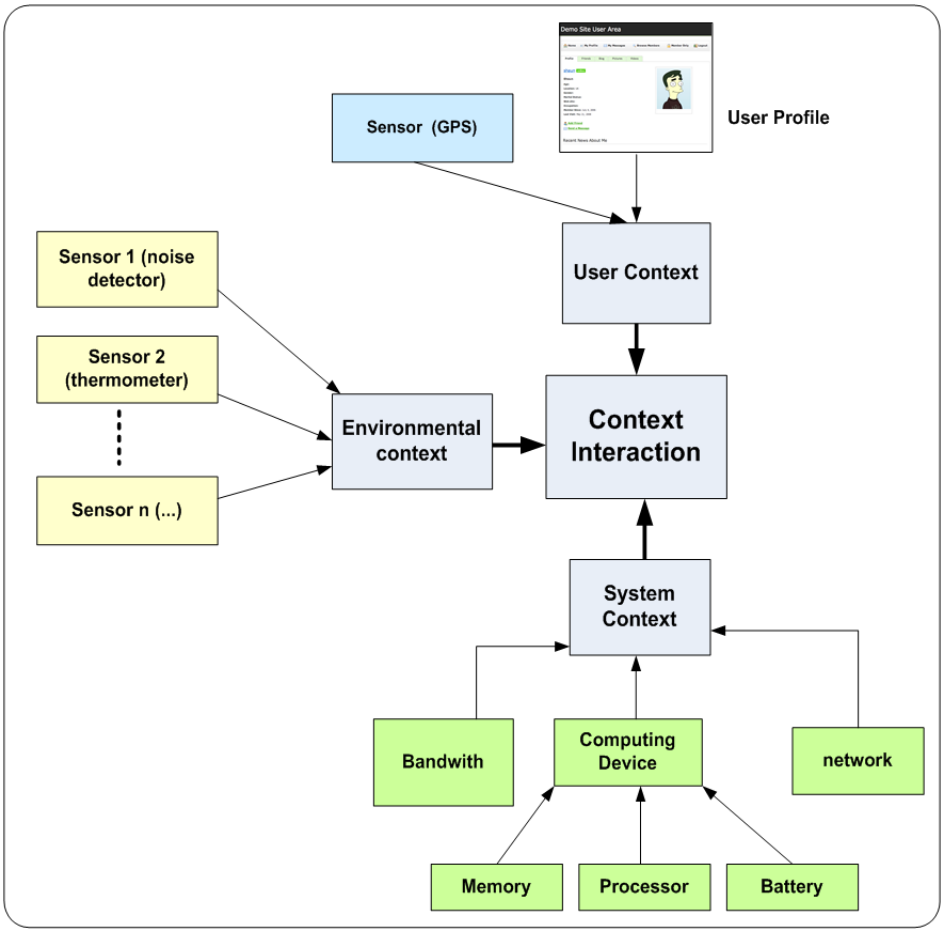


Figure 2.6 Interaction context module

This module is composed of three sub-modules:

User context: This module detects the location and status of the user. It defines the user's ability to use certain modalities. For example, the system disables the display modality if it detects that the user is usually impaired and disable the audio modality if it detects that the user is in a library ;

Environment context: It describes the state of the environment such as determination of the noise level. It is understood that the use of the audio method (entered or left) is affected by this information. If the noise level is high, the audio modality will be disabled ;

System context: The capacity and the type of the system that we use, are factors that determine or limit the modalities that can be activated.

A modality is appropriate to a given instance of interaction context if it is found to be suitable to every parameter of the user context, the environmental context and the system context.

The suitability of specific output modality is shown by the relationships given below extracted from Table2.1 to Table 2.5. Symbols √ and × are used to denote suitability and non-suitability, respectively.

$$VO = (user \neq deaf) \wedge (location \neq at\ work)$$

$$MO = (user \neq manually\ handicapped) \wedge (location \neq on\ the\ go) \wedge (computer \neq cellphone/PDA \vee computer \neq iPad)$$

$$VIO = (user \neq visually\ impaired) \wedge (workplace \neq dark) \vee (workplace \neq dark)$$

Table 2.1 User location and its suitability to modalities

Modalities	At Home	At Work	On the go
Vocal Output (VO_{out})	√	x	√
Manual Output (M_{out})	√	√	x
Visual Output (VI_{out})	√	√	√

Table 2.2 User handicap/profile and its suitability to output modalities

Modalities	Regular User	Deaf	Mute	Manually Handicapped	Visually Impaired
Vocal Output (VO_{out})	√	x	√	√	√
Manual Output (M_{out})	√	√	√	x	√
Visual Output (VI_{out})	√	√	√	√	x

Table 2.3 Noise level and its suitability to output modalities

Modalities	Quiet	Noisy
Vocal Output (VO_{out})	√	x
Manual Output (M_{out})	√	√
Visual Output (VI_{out})	√	√

Table 2.4 Brightness or darkness of the workplace and how it affects the selection of appropriate output modalities

Modalities	Workplace Bright	Workplace Dark	Workplace Very Dark
Vocal Output (VO_{out})	√	√	√
Manual Output (M_{out})	√	x	x
Visual Output (VI_{out})	√	√	√

Table 2.5 The type of computing device and how it affects the selection of appropriate output modalities

Modalities	PC/Laptop	Ipad	Cellphone/PDA
Vocal Output (VO_{out})	√	√	√
Manual Output (M_{out})	√	x	x
Visual Output (VI_{out})	√	√	√

In this work, the proposed system detects available media devices and produces result in which appropriate modalities are noted. An example is shown in Figure 2.7.

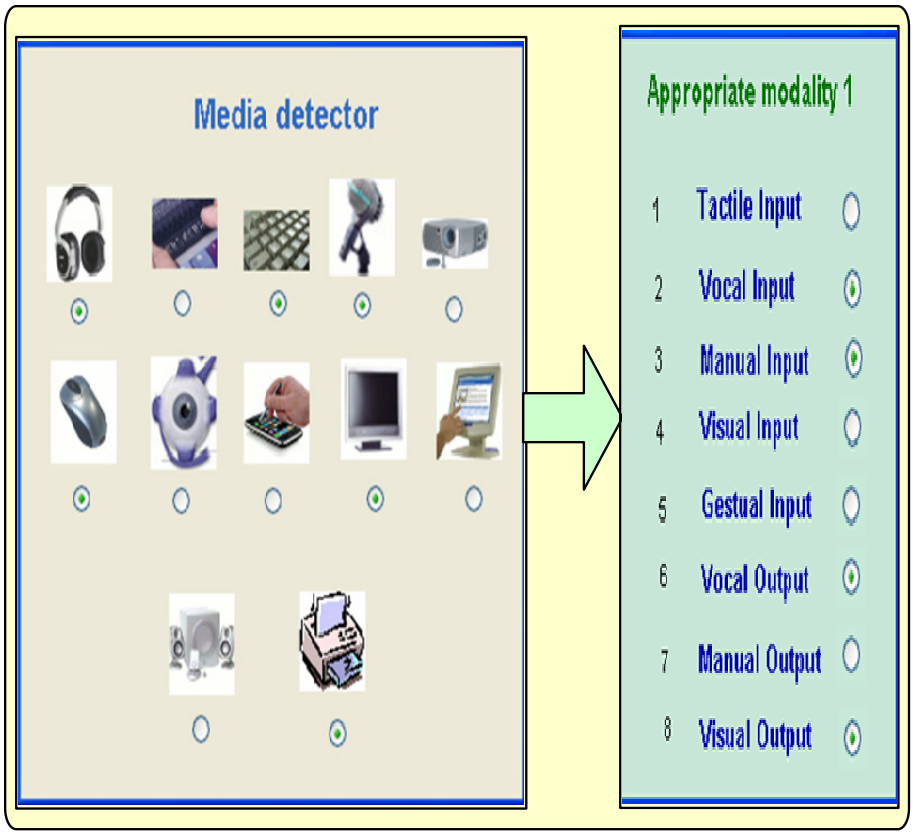


Figure 2.7 System detection of appropriate modalities based on available media devices

The system detects the values of all related interaction context parameters and accordingly selects the appropriate modalities. A sample of such scheme is shown in Figure 2.8.

Figure 2.8 System detection of appropriate modalities based on the instance of interaction context

Figure 2.9 illustrates how the optimal modalities are established – that is, finding the intersection between appropriate modality 1 (a set of modalities) and appropriate modality 2 (a set of modalities). In the cited case, the optimal modalities are vocal input, visual and vocal output. For more details concerning interaction context, the reader can refer to (Zaguia et al., 2010c).

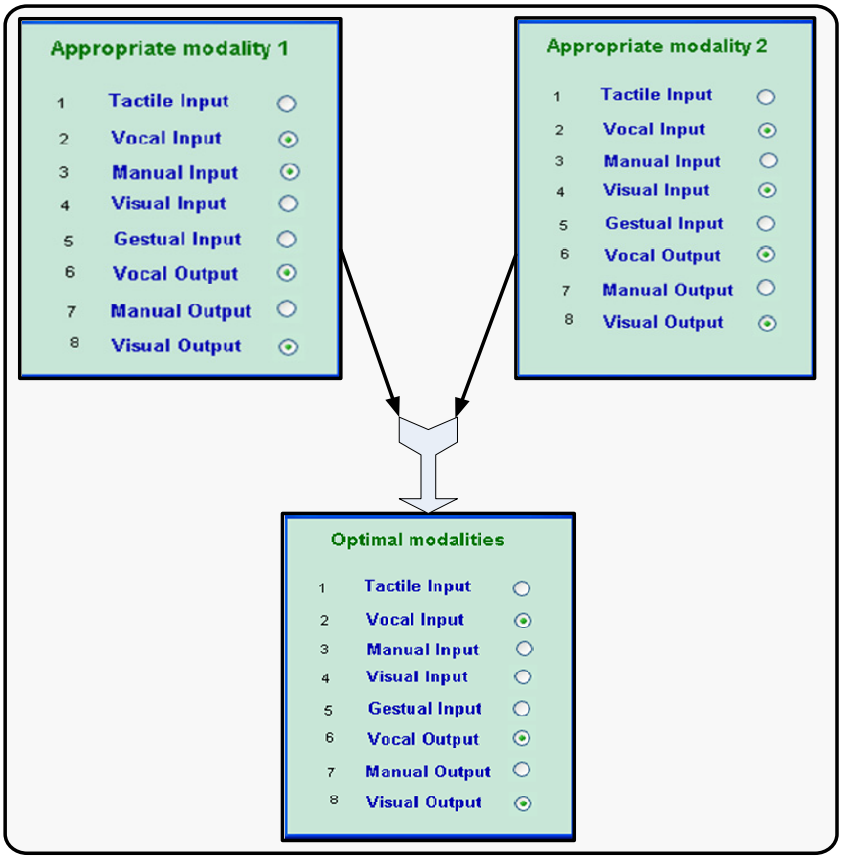


Figure 2.9 Optimal modalities – the results of the intersection between the set of appropriate modality 1 and the set of appropriate modality 2

2.4.3 Multimodal Fission

This module is the crucial component of our architecture. In this section, we describe our fission module with focus on the use of patterns as a solution to the described problems of fission systems find in literature.

We start by defining the pattern found in literatures and then we present how we use it to resolve our problem.

2.4.3.1 Pattern

In this section, we define the pattern as found in the literature and we show how we adapt it in our case.

Alexander (Alexander, Ishikawa et Silverstein, 1977b) has defined the pattern as a problem that often occurs in an environment. For Grone (Grone, 2006), "patterns help transporting knowledge and provide common names for solutions".

Generally patterns are defined with two parts, namely, problem and solution. Therefore, we must define the problem and the solution so that we can talk about patterns.

1. Subtasks-Modalities Association

In this section, we present the use of patterns to associate the adequate modality (ies) to subtask.

Patterns are predefined models that describe a modality (ies) selection. In our work, a modality pattern is composed of: a) *Problem* composed of the components: Application, Parameter, Priority, Combination, Scenario and Service and b) *Solution* composed of the chosen modality as shown in Figure 2.10.

Our goal is to determine the best modality (ies) suited to a specific context (the solution: here is to choose a modality). We should define all the parameters that affect the choice of modality (problem: taking into account a number of parameters to choose this modality). So through these steps, we can model our patterns. Patterns will be stored in a knowledge base. The significance of each component in Figure 2.10 is as follows:

Application*:* Each modality is connected to a given application ;

Parameters*:* parameters of the pattern must contain elements that suit for a specific modality (entities, attributes, properties). These elements ensure the equivalence between data of a modality and the pattern itself (start time, end time, etc.) ;

Priority*:* each modality has a priority depending on the application ;

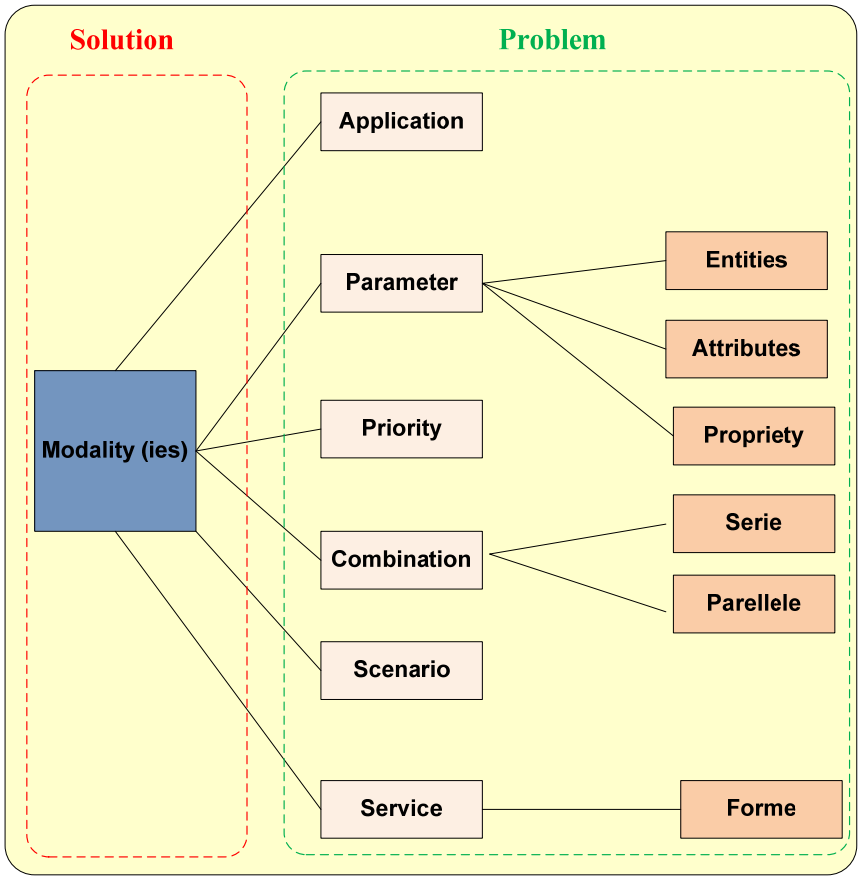


Figure 2.10 Pattern of modality (ies) selection

Combination*:* modalities can be combined either in serial or in parallel with other modalities;

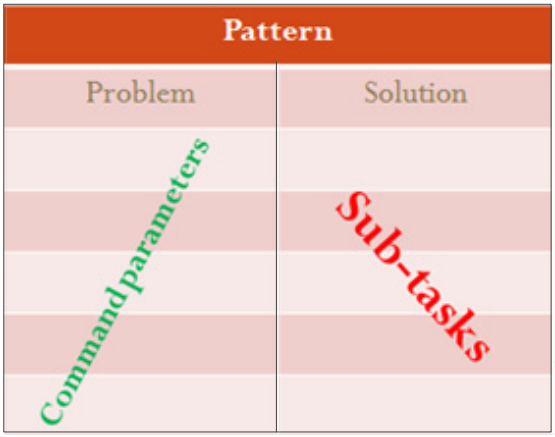
Scenario*:* each modality is connected to a scenario or sub-task ;

Service*:* each modality offers a service that can change shape. For example, if a modality offers the service display, the system increases or decreases the brightness (shape) depending on the level of light in the room.

2. Pattern of Sub-tasks Selection

For the pattern of sub-tasks, selection is composed of command parameters (problem) and the solution will be the sub-tasks (Table 2.6). Our patterns will be stored in a knowledge base.

Table 2.6 Pattern of sub-tasks selection



Our goal is to select the adequate subtasks for each command. The command parameters (problem) are the words that compose the command. For example if the command is “Bring me the cup” the parameters of this command are “Bring-me-Cup” (problem) and the solution is {move to the cup-take the cup-move to the position (me)-depose the cup}.

2.4.3.2 Fission algorithm / Fission rules

In general, the fission rule is simple: if a complex command (CC) is presented, then a set of sub-tasks with the modalities (and its parameters) are deducted.
Multimodal fission can be represented by the function:

$$f: F \rightarrow ExK$$

$$\forall cc \in F, \exists STi \in K \; and \; MOj \in E,$$

$$f(STi, MOj) = cc$$

With: $i \in [1..n] \; et \; j \in [1..m]$

$$f: CC = \sum_{i=1}^{n} ST_i \left(\left\{ \bigcup_{j=1}^{k} MO_j \right\}, \left\{ \bigcap_{j=1}^{l} MO_j \right\} \right) \quad (2.1)$$

with: ST = sub-task.

MO = output modality.

CC= complex command.

l and k are different from m and n because it depends on the sub-tasks. For example, for some sub-task we will use just two terms even if we have three modalities available.

In equation (1), the symbol $\bigcup$ indicates that we can use either one or several modalities to present a sub-task. For example, if we present a text to the user, we use audio or display. The symbol $\bigcap$ indicates that we use the available modalities together to present a sub-task.

Stages of the fission process are described in Figure 2.11, which shows the proposed fission algorithm. We assume that each XML file contains data for output modalities, their corresponding parameters and the complex command.

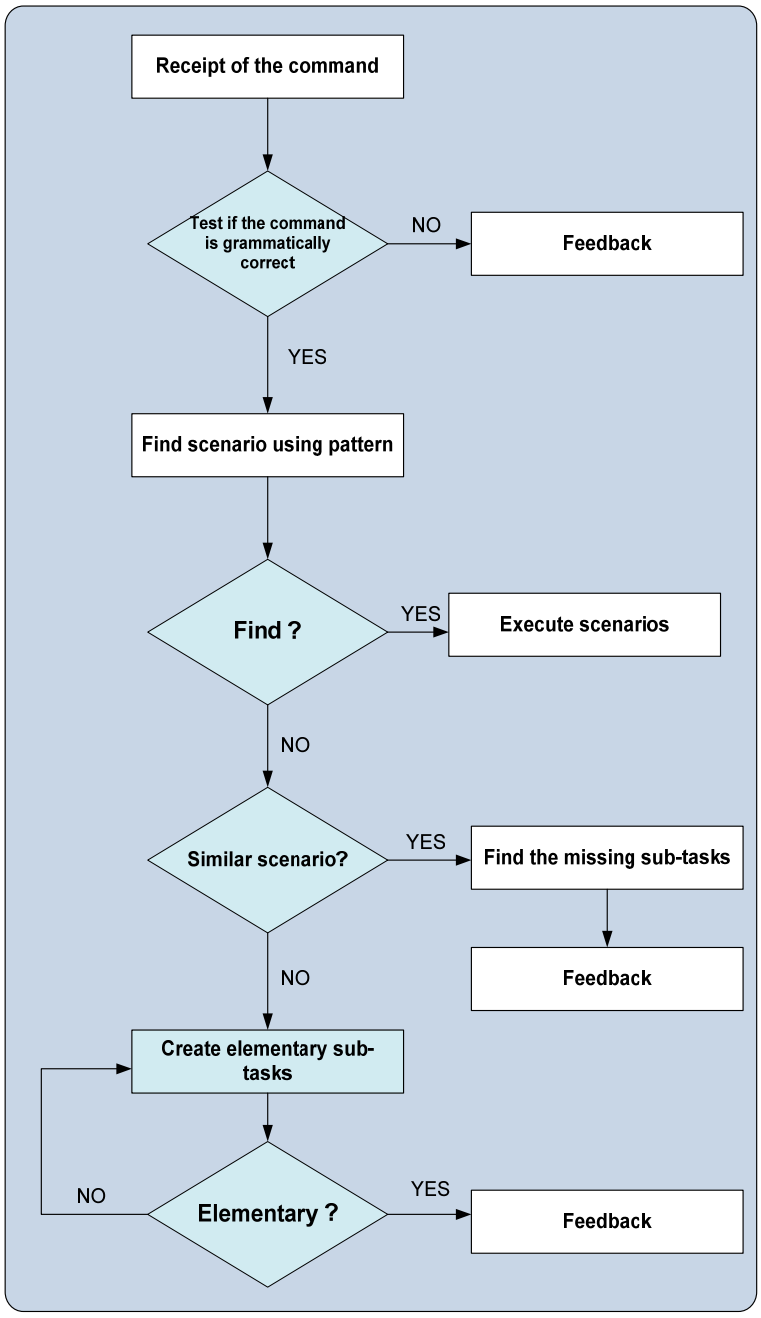


Figure 2.11. Fission algorithm

When the system receives an XML file, as shown in Figure 2.12, it extracts the command and it checks if it is complete by checking grammar rules. We defined many grammar rules among these rules for instance:

- AFMO →MO for example “drop the cup” ;

- AFP→P→MO for example “give me the pen”.

with AFMO = action for movable object, MO = movable object and P= person.

If this command is incomplete, the system sends a feedback to the user to correct the error. Otherwise, a search pattern (pattern of subtasks selection) scenarios in the knowledge base starts.

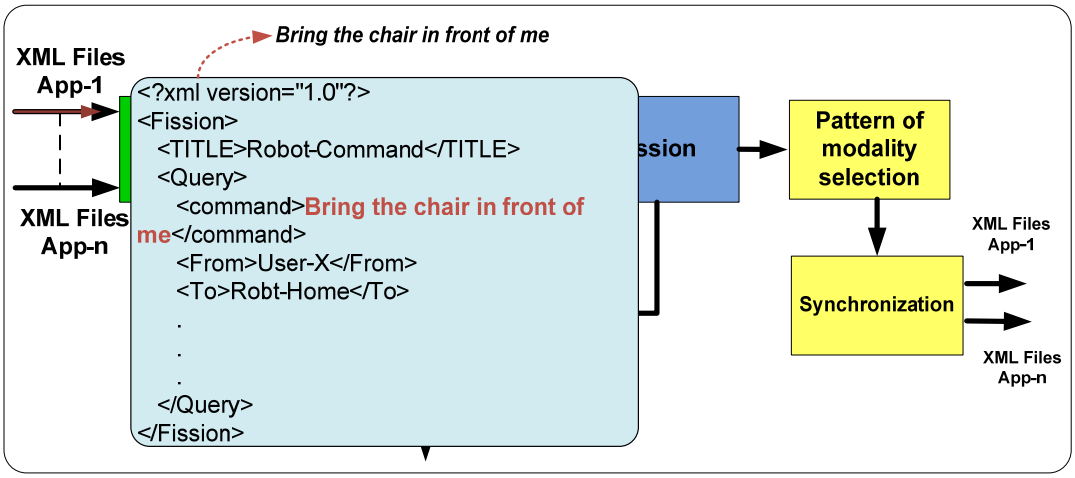


Figure 2.12. Extracting the command from XML file

For example, if the command is "bring the chair in front of me," in this case, we look sub-tasks with sending a query (problem: bring object position) to the knowledge base.

If the scenario is found, the system performs sub-tasks. Otherwise, the system will search if there is a similar scenario. For instance, if in our knowledge base there is a scenario "prepare coffee" and the system receives a command "prepare coffee with milk" in this case, we have similar command so we take the result for "prepare coffee" and the system creates missing sub-tasks related to milk using our knowledge base (
Figure 2.13). From

Figure 2.13, the subtask related to liquid (milk) is “add milk” so this subtask is added to subtasks of “prepare coffee" and then request feedback from the user.

In the case where the system cannot find a similar command, it creates elementary sub-tasks. The system asks the user to confirm if the result is correct.

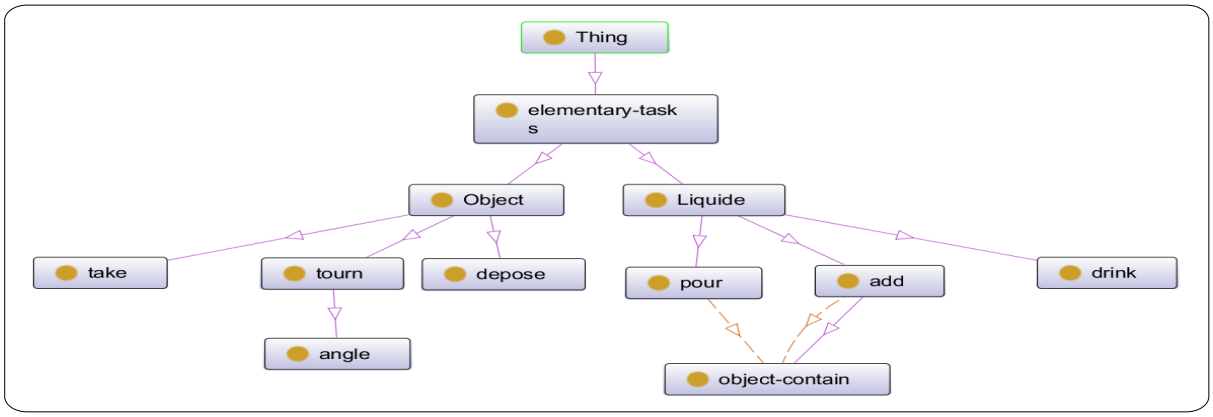


Figure 2.13 Knowledge Base of elementary tasks

Suppose the system receives a command "Bring the chair in front of me" with the current context = living room. We can define a query (problem: bring object position), which will contain the necessary elements to search in the knowledge base if the scenario has already occurred.

With Action = *Bring*, Object = *the chair* and position= *in front of me*. Suppose we have patterns registered in our knowledge base as shown in Figure 2.14.

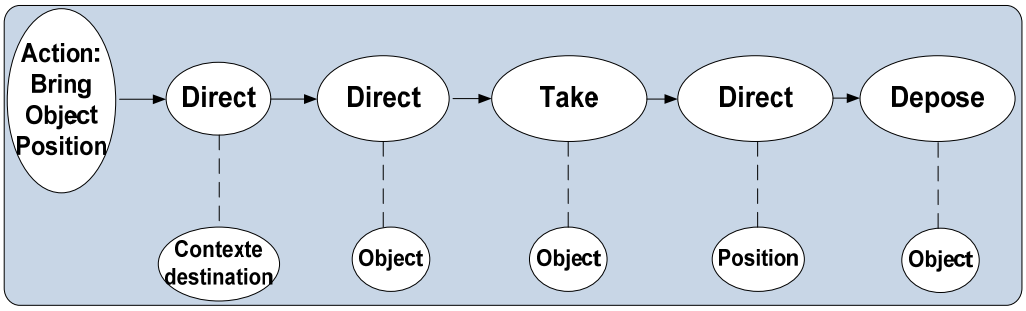


Figure 2.14 Example of pattern

The search result is shown in Table 2.7. In Figure 2.14, "destination context" is the same as the "current context", adding "direct (destination context)" will allow us to manage orders for other different contexts position. For example, if the command was "Bring me a spoon" and the current context = *living room*. Since the spoon is in the kitchen so the destination context = *kitchen*.

Table 2.7 Pattern with the solution

Pattern	
Problem	Solution
Bring	Direct to context-Destination
Object	Direct to Pos-object
Position	Take object
	Direct to position
	Depose object

2.5 Simulation

A handicap user sitting in the living room, uses his mobile device to send a message "bring me that" and point with his hand to a cup, to a robot located behind him. An XML file is created and sent to the fusion module as shown in Figure 2.15. The file contains the command, the available modalities and the context. After the process of fusion, this module sends the complex command to the fission module as shown in Figure 2.15: "bring me the cup" with the coordinate of the cup and the user.

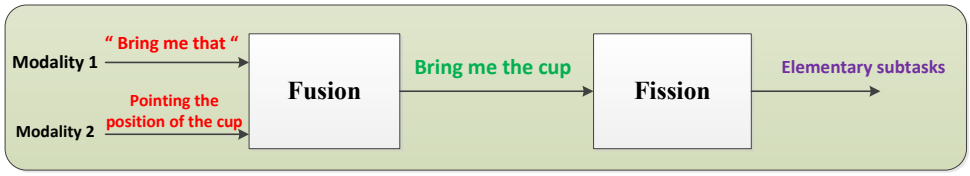


Figure 2.15 Example of scenario

To validate the stages of fission process, we use the colored Petri Net (CPN) (Jensen, 1987). The CPN is more advantageous than the ordinary Petri Net.

There are many simulation tools, often free, and developed in the context of thesis or scientific research. We used CPN Tools (Zhu, Tong et Cheng, 2011), one of the most used software, simulation of the high-level Petri network.

A CPN is a graphical structure linked to computer language system. The CPN is a Petri net in which the tokens are colored. The color is an information attached to the token. This information allows to distinguish between tokens and it can be any type (integer, real, String, Boolean, list etc.). It is based on a set of conditions and expressions that permit to the tokens to change their colors (change the state of the place).

Figure 2.18 shows the diagram of the example "Bring me the cup". This diagram shows the stages that the CPN validates to prove that system works properly.

Figure 2.19 shows the framework of the multimodal fission with CPN. In the input, we have the command "bring me the cup" as a string token. In the following diagrams, we present the validation of the fission module.
Figure 2.20 shows the parser module. We extract every word from the command to perform grammatical analysis.

Figure 2.21 to Figure 2.23 show CPN for the communication between T_Grammar and Knowledge Base to validate if a command is correct and the creation of query for the search of sub-tasks. In our case the answer is "AFP person Mo" with:

- AFP: action for person= Bring ;
- person= me ;
- Mo: movable object = cup.

T_Fission module uses query that contains elements as shown in Table 2.8 to search the adequate pattern-subtasks in the knowledge base (Figure 2.25).

Table 2.8 Query for "AFM person Mo" with
AFM = bring, person=me and Mo= cup

Problem
AFM
Person
Mo

Figure 2.24 presents the query **"AFM person Mo"** sent to the knowledge base to find the matching pattern.

Figure 2.25 presents the solution of the problem **"AFM person Mo": ["1-move to the object", "2-take the object","3-move to the position", "4-depose the object"];**

Figure 2.26 shows the final subtasks to be executed by the system:

1. Move to the object ;
2. Take the object ;
3. Move to the position ;
4. Depose the object.

Figure 2.16 and Figure 2.18 show the declarations of variables and functions used with CPN for this scenario.

```
Standard declarations
  colset INT
  colset UNIT
  colset BOOL
  colset STRING
  colset Action_Verb_List = list STRING ;
  var command : STRING;
  colset ListCommand = list STRING;
  var valChar, valCh, listMot, ListSubTask : ListCommand;
  var x: INT;
  var test_grammair, testStr :STRING;
  var verb_Ac: Action_Verb_List;
  val AFMO = ["put","change"] ;
  val AFNMO = ["clean", "close","open"] ;
  val AFP = ["bring","answer", "ask","open"] ;
  val AOb = ["box", "chair","table", "cup"] ;
  val ObL = ["cofee", "jus","water"] ;
  val Person = ["me","him", "her"]
```

Figure 2.16 Declaration of variable

```
val posit = ["here", "kitchen","water"] ;
var ttt , zzz, kkk, ccc,h, valOnto, valPattern: STRING;
var pattern, subTask, patternP : STRING;
val pattern1 = " AFMO SMO IL AMO";
val pattern2 = " AFP person Mo ";
val Spattern1 = ["1-move to the object",
 "2-take the object","3-move to the position", "4-depose the object"];
val Spattern2 = ["1-move to the small object",
 "2-take the  small object","3-move to the average object", "4-depose the small object"];
val suprim = ["the", "of"];
val stop = "zzzzzz";
fun testGrammar(valOnto) = if mem AFMO valOnto then "AFMO"
  else if mem AFNMO valOnto  then "AFNMO"
  else if mem Person valOnto then "person"
  else if mem AFP valOnto then "AFP"
  else if mem AOb valOnto  then "Mo"
  else if mem posit valOnto  then "position"
  else if valOnto  = "the" then ""
  else if valOnto  = stop then stop
  else ""
fun patternOnto (valPattern) = if valPattern = pattern1 then Spattern1
  else if valPattern = pattern2 then Spattern2
  else []
```

Figure 2.17 Declaration of variables

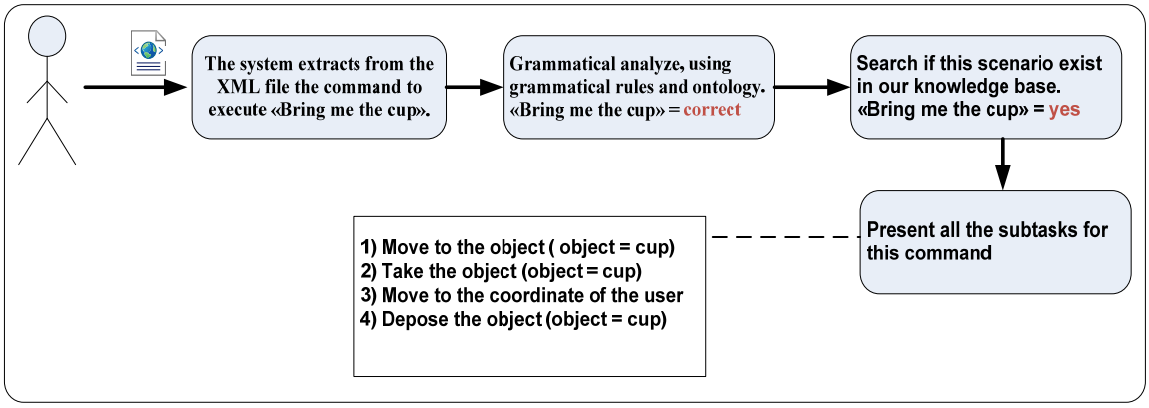


Figure 2.18 Example of scenario

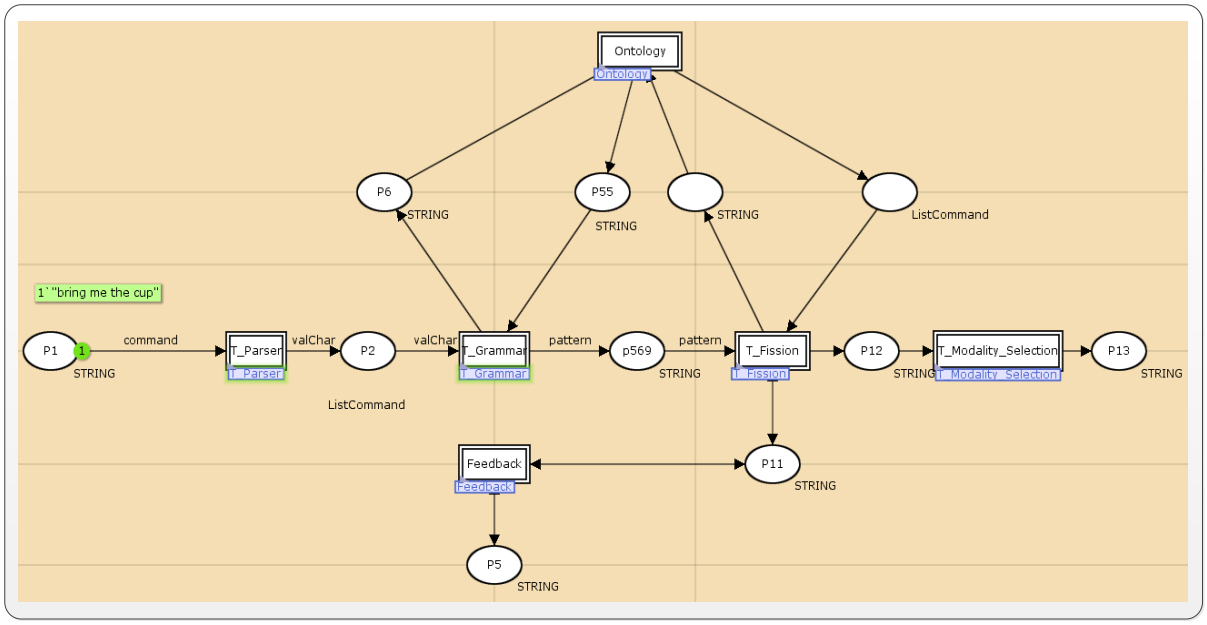


Figure 2.19 Framework of the multimodal fission with CPN

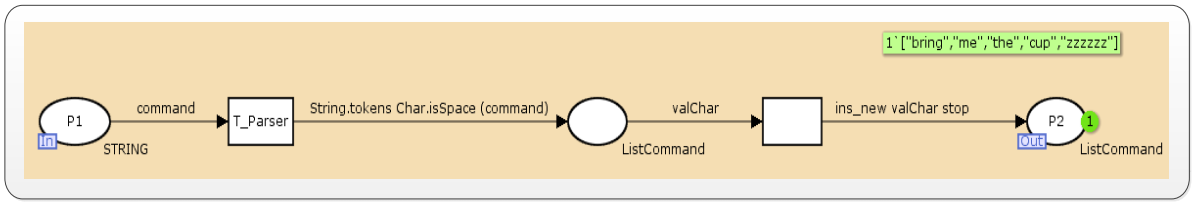


Figure 2.20 Colored Petri Net showing the operation of parser module

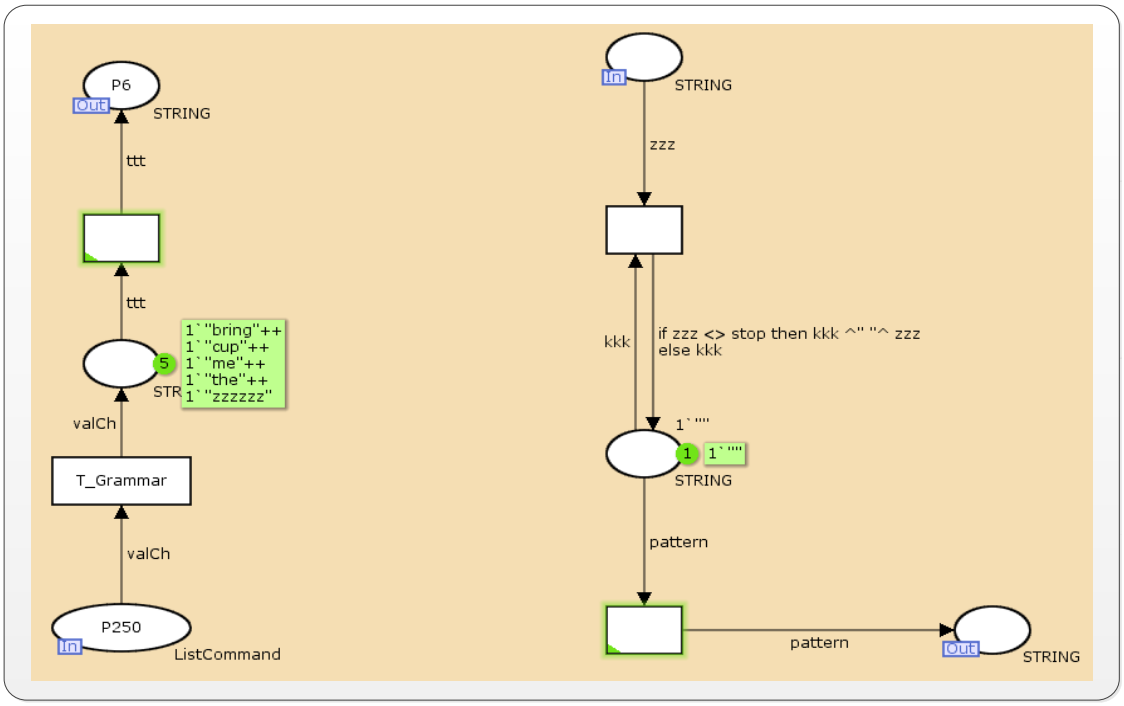


Figure 2.21 Colored Petri Net showing the operation of Grammar module

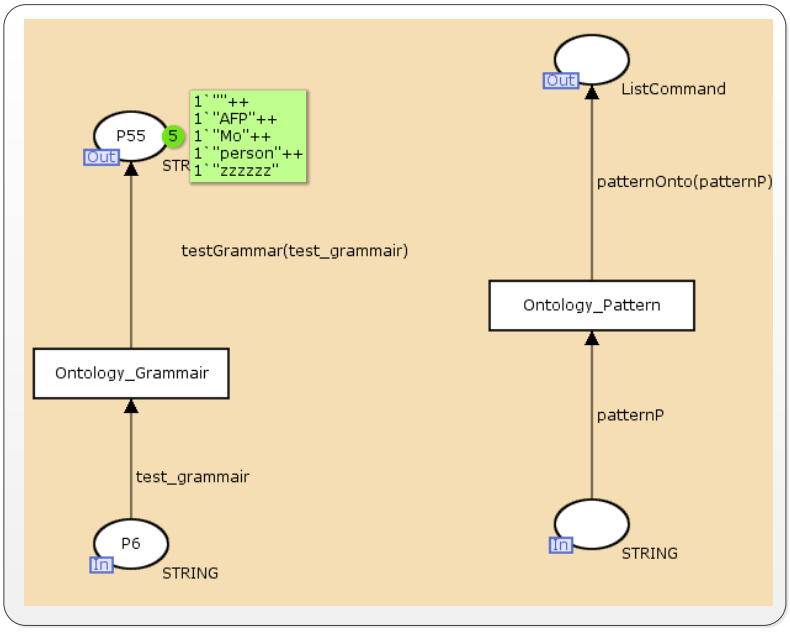


Figure 2.22 Colored Petri Net showing the operation of ontology concerning the grammar

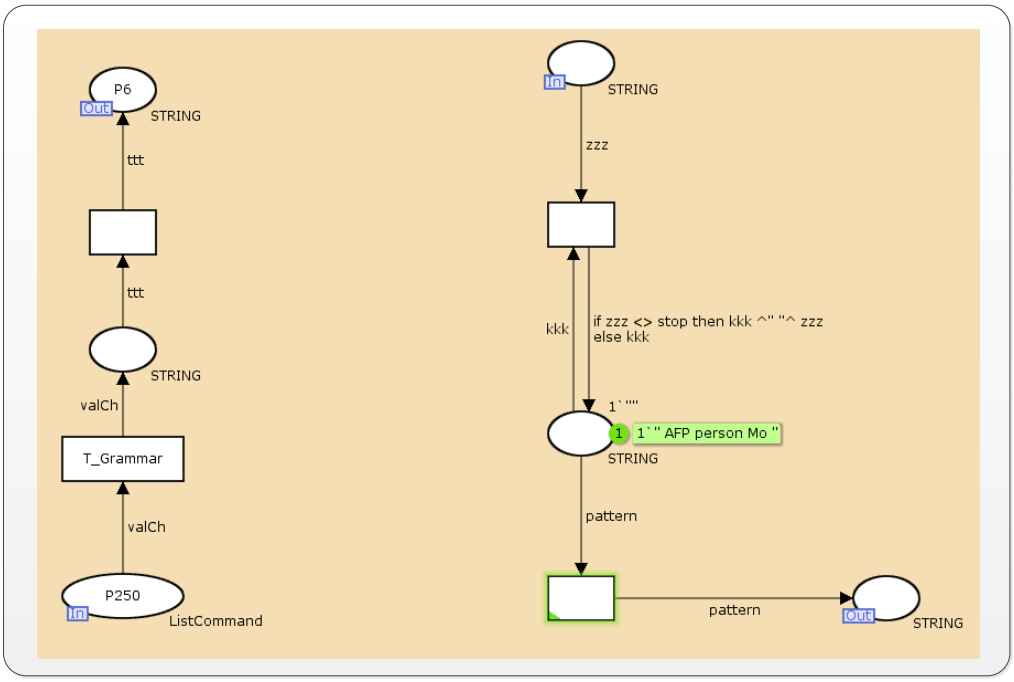


Figure 2.23 Colored Petri Net showing the creation of pattern to find sub-tasks

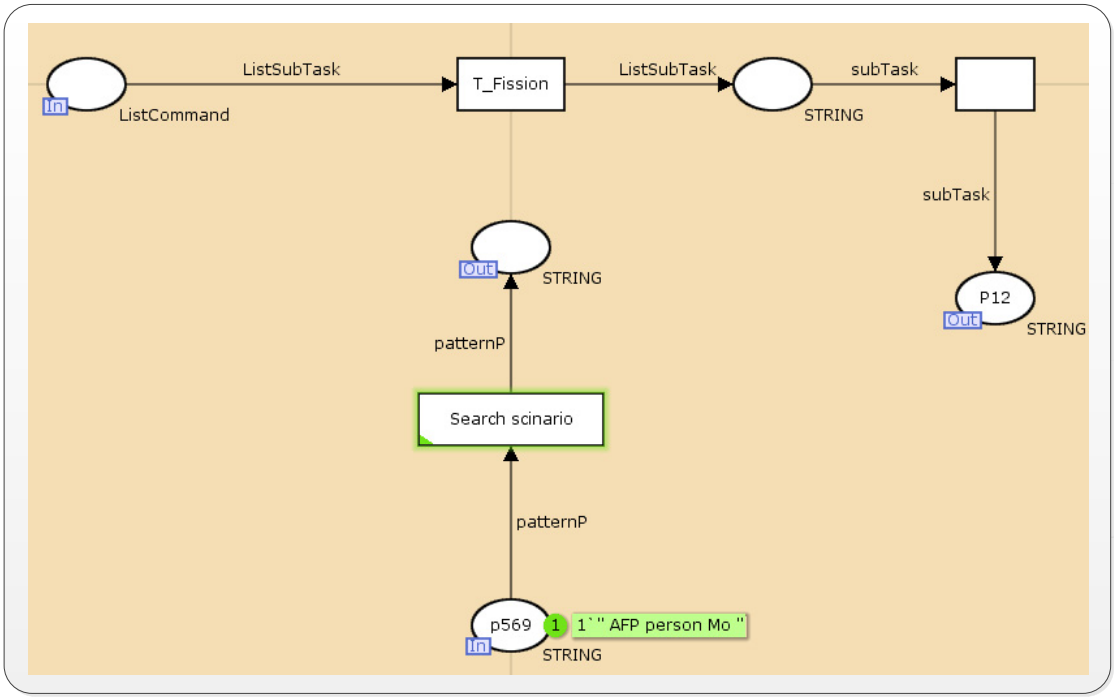


Figure 2.24 Colored Petri Net showing the search of sub-tasks using pattern

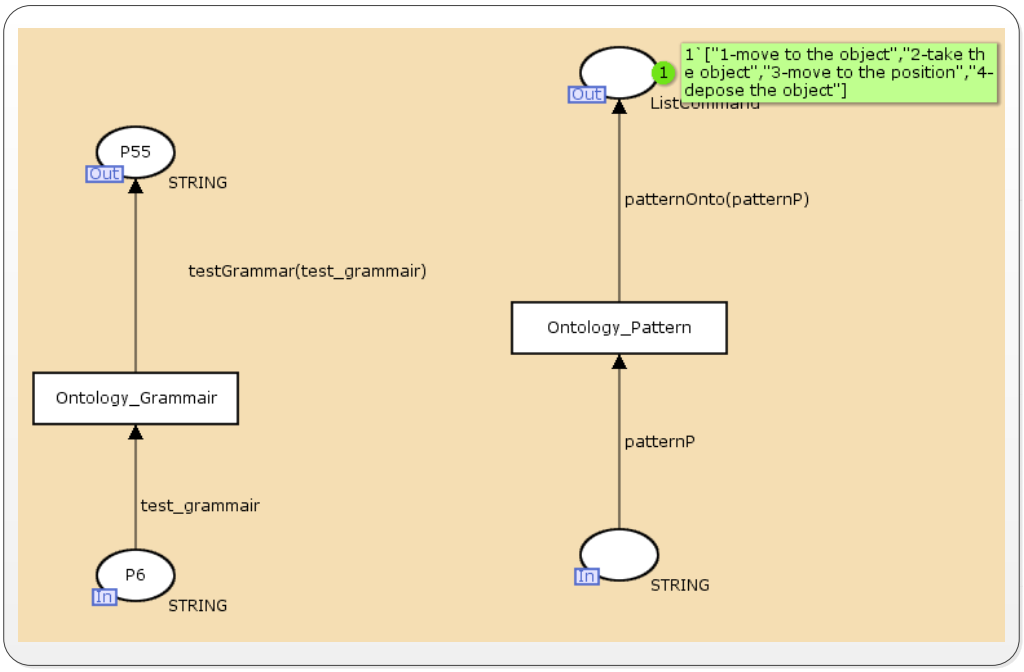


Figure 2.25. Colored Petri Net showing the sub-tasks found for the pattern "AFP person Mo"

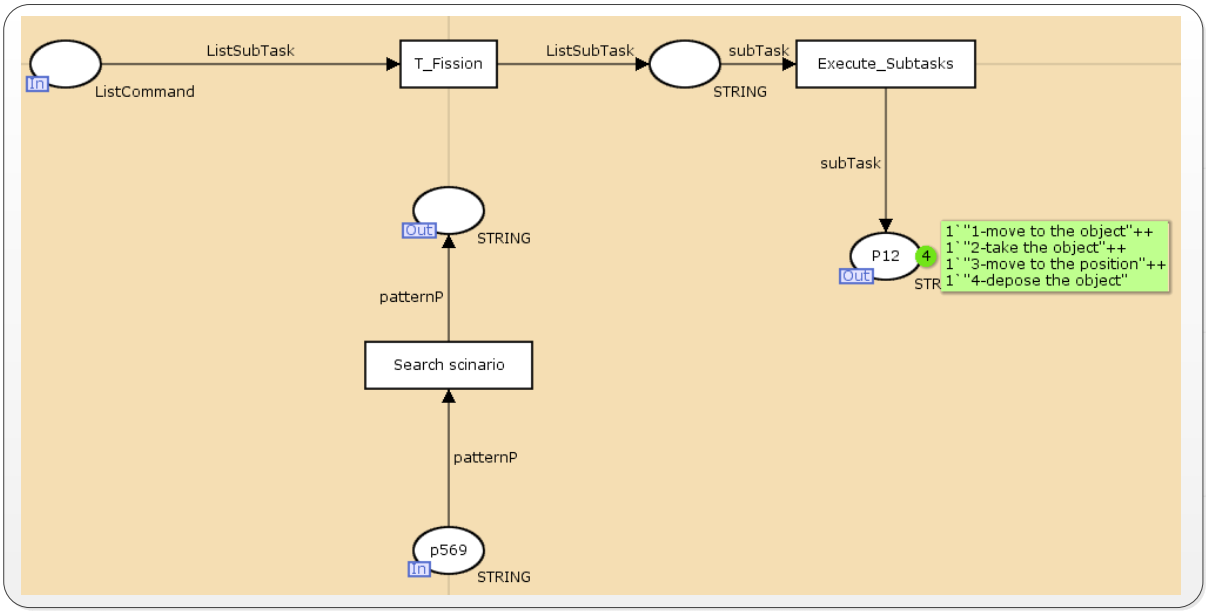


Figure 2.26 Colored Petri Net showing the execution of sub-tasks

2.6 Conclusion

In this article, we presented an architecture which is very useful in a multimodal system. We presented an effective algorithm for the fission process. We have shown the important role of the fission module. This module subdivides a complex command into elementary sub-tasks and presents them to the output modalities. The proposed solution facilitates the work of the fission module, using predefined patterns stored in a knowledge base or ontology. These patterns save time and facilitate the executions of the tasks. They can also be used by other researchers in their own work. A concrete example is illustrated in order to show the effectiveness of the contribution, using CPN tool.

This project has several scientific benefits. It will promote research in this area, contribute to the advancement of existing knowledge in the human-machine interaction domain generally and fission specifically, and provide researchers with a fission engine and a set of patterns that can be very useful. On the other hand, like any scientific research project, the essential goal remains humanity with making life easier for others especially for the elderly, disabled or sick persons. The created architecture can be deployed on robots, mobile devices, computers, etc.

CHAPITRE 3

MODELING RULES FISSION AND MODALITY SELECTION USING ONTOLOGY

Atef Zaguia[1], Chakib Tadj[1], Amar Ramdane-Cherif[2], Ahmad Wahbi[1,2], Moeiz Miraoui[3]

[1]MMS Laboratory, Université du Québec, École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3 Canada
[2]LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France
[3]Institut Supérieur des sciences appliquées et de technologie de Gabes

This article is published in the Journal of Software Engineering and Applications, Vol. 6, No. 7, July 2013, pp.354-371.

Résumé

Les chercheurs en science de l'informatique et génie informatique consacrent une partie importante de leurs efforts dans la communication et l'interaction homme-machine. En effet, avec l'avènement de traitement multimédia et multimodal en temps réel, l'ordinateur n'est plus considéré comme un outil de calcul uniquement, mais aussi comme une machine pour le traitement, la communication, la collecte et le contrôle. Beaucoup de machines assistent et supportent de nombreuses activités de la vie quotidienne.

L'objectif principale de cet article est de proposer une nouvelle solution méthodologique en modélisant une architecture qui facilite le travail de système multimodal en particulier pour le module de fission. Pour réaliser de tels systèmes, nous utilisons l'ontologie pour intégrer les données sémantiquement. L'ontologie fournit un vocabulaire structuré servant de support pour la représentation des données.

Ce document fournit une meilleure compréhension du système de fission et l'interaction multimodale. Nous présentons notre architecture et la description de la détection des modalités optimales. Ceci est fait en utilisant un modèle ontologique qui contient différents scenarios et qui décrit l'environnement dans lequel un système multimodal existe.

Mots clés : système multimodal, fission multimodale, modalité, ontologie, contexte d'interaction, pattern.

Abstract

Researchers in computer science and computer engineering devote a significant part of their efforts on communication and interaction between man and machine. Indeed, with the advent of multimedia and multimodal processing in real time, the computer is no longer considered only as a computational tool, but as a machine for processing, communication, collection and control. Many machines assist and support many activities in daily life.

The main objective of this paper is to propose a new methodological solution by modeling an architecture that facilitates the work of multimodal system especially for a fission module. To realize such systems, we rely on ontology to integrate data semantically. Ontologies provide a structured vocabulary used as support for data representation.

This paper provides a better understanding of the fission system and multimodal interaction. We present our architecture and the description of the detection of optimal modalities. This is done by using an ontological model that contains different applicable scenarios and describes the environment where a multimodal system exists.

Keywords: multimodal system, multimodal fission, modality, ontology, interaction context, pattern.

3.1 Introduction

The communication plays an important role in our daily life as it allows people to interact with each other as individuals or groups. In fact, humans have a sophisticated ability to communicate and exchange information that is due to the division of the language, as well to the common comprehension of the operation of the things and an implicit understanding of the daily situations.

Since many years, the researchers want to find solutions to share common understanding of the structure of information among people or the machines and to re-use this knowledge in order to allow the creation of intelligent machines or systems able to understand the users, to perceive his environment and to react in ways to maximize the success rate of understanding and responding. Among these systems the multimodal systems that permit to combine in input and/or in output several modalities (Oviatt, 2003) dynamically (Figure 3.1).

These systems receive their input from sensors and gadgets (camera, microphone, etc.) and they interpret and understand these inputs (Zaguia et al., 2010b), (Lalanne et al., 2009) and (Feiteira et Duarte, 2011). This is known as fusion process. The resulting command is then executed in the output gadgets (fission process) (screen, speakers, projector etc.). This is known as fission process. Combining these modalities in the input/output is called multimodality (Bernsen, 2008; Lauer, 2009).

A known example of these systems is the Bolt system "Put That There" (Bolt, 1980a), where the author used the gesture and speech to move objects.

These systems improve the recognition and the understanding of the environment command (user, robot, machine etc.) by the machine.

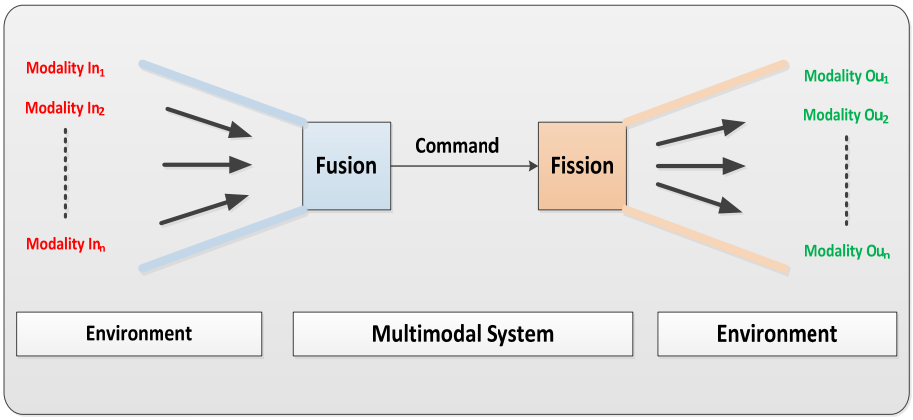


Figure 3.1 Multimodal system

But understanding is only possible if these systems or these machines are equipped with a knowledge base.

Knowledge includes all necessary components to achieve the annotation of the data, such as it can be performed by experts in a particular field.

Several studies have focused on the improvement of intelligent systems and especially the creation of systems that enable semantic interoperability, which means the systems will not only exchange data with each other in a given format (eg. the string "Canada") but also must have the same meaning for both parties (a Country).

So the main goal is to find a way to present data so that the machines, the users, the applications can understand. The most adequate solution found is ontology (Guarino, Oberle et Staab, 2009).
The ontology will have a good impact in multimodal systems and specially for fission module (Foster, 2002).

The fission module is a fundamental component of multimodal interactive system. It is mainly used at the output. Its role is to subdivide the requests/commands made by the user to elementary subtasks, then associate them to the appropriate modalities and to present them in

the available output media (Carnielli et al., 2008). The meaning of the command may vary according to the context, the task and the services. This paper is focused on the fission process. We propose a new methodological solution by modeling an architecture that facilitates the work of a fission module, by defining an ontology that contains different applicable scenarios and describes the environment where a multimodal system exists. The proposed architecture has three main characteristics:

Openness: handling a large number of modalities that prevents the restriction in its application to specific domains ;

Flexibility: the use of ontology makes the description of an environment and its scenarios easier;

Consistency*:* the description of the most potential objects and scenarios of the environment ;

This paper discusses these characteristics by explaining the architectural design of the proposed solution.

The rest of this paper is structured as follows. Section 3.2 presents the problems related to the fission process and highlights the novelty of our work. Section 3.3presents related researches. Sections 3.4 to 3.6 present the design of the architecture, the interaction context and the ontology which we will use to solve the problem "How to present fission rules and modality selection for multimodal systems?". Sections 3.7 and 3.8 describe our proposed fission algorithm and a scenario respectively. Conclusion is presented in Section 3.9.

3.2 Problematic and a proposed solution

A system can be called multimodal, if it provides input or output combining multiple modalities, so that the resulting communicative system is more efficient.

In our work, we focus specifically on 1) the services connected to the output: multimodal fission and 2) the creation of multimodal interaction system.

The first challenge is: what are the required modules to build the architecture of a fission system? Here, we will specify, define and develop all necessary components of the system. We will also show how they communicate.

The second challenge is how to select the output modalities considering that the state of the environment is dynamic? Here we will use the interaction context (Zaguia et al., 2010a) to resolve this problem.

The third challenge is how to subdivide a command to elementary subtasks? Here we will use a predefined pattern for all possible scenarios.

The fourth challenge is how to select the appropriate and available modality (ies) for a given sub-task using predefined patterns that describe a modality (ies) selection.

The fifth challenge is data modeling and how do we make it dynamic, flexible, easy to update and describe it easily? We will use ontology as a knowledge base.

The sixth challenge concerns validation of our proposed architecture (formalism)? We focus more closely on the design, specification, construction and evaluation of our fission architecture. We use the CPN-Tools (Colored Petri Net - Tools) to modulate and simulate our architecture.

To summarize, our goal is to develop a fission component for multimodal interaction. We also elaborate an efficient fission algorithm. To achieve our goal, we create a context sensitive architecture, able to manage multiple modules and modalities and automatically adapts to dynamic changes of the interaction context.

3.3 Related work

The multimodal interaction is a regular characteristic for each activity and human communication, in which we speak, listen, watch, etc., alternately or simultaneously.

The objective of the research in multimodality is to develop a flexible system capable to manipulate many modalities. We assume that the environment has a rich collection of different media/modality components. The fission module is a crucial component of multimodal system. But most research in multimodal systems (Jaimes et Sebe, 2007) focus more on the fusion than the fission. We can support our point by "There isn't much research done on fission of output modalities because most applications use few different output modalities therefore simple and direct output mechanism are often used" (Costa et Duarte, 2011). Also the allocation of output modalities was rather hard coded than based on intelligent system for the early multimodal system.

In general, the process of fission is the manner to segment the data which will be presented to the user depending on availability of modalities. For Foster, (Foster, 2005) multimodal fission is the process of realizing an abstract message through output on some combination of the available channels (Foster, 2002).

According to (Grifoni, 2009) and (Foster, 2005), the fission process goes throw three main stages: (1) Select and organize the content: this step consist of selecting and organizing the content that will be presented. (2) Modality (ies) selection: specifies the modalities that can display or present the command. (3) Coordination of outputs: coordinate outputs for each channel in order to create a coherent presentation.

In (Rousseau et al., 2006), Rousseau et al. describe a conceptual modal « WWHT (What-Which-How-Then) » for the design of multimodal system and the presentation of information in the output. This model is based on concepts What, Which, How and Then: 1) what is the information to present? 2) Which modalities should be used to present this information? 3)

How to present this information using these modalities? and 4) then how to handle the evolution of the resulting presentation?

In (Benoit et al., 2009), Benoit et al. present a very simple multimodal fission system that detects the state of the driver. They capture information from sensors installed in the car and then they test if the values entered are in some specific intervals and through this test, the system will generate alerts.

We propose a new solution for the fission module through modeling patterns that deal with different modalities and different possible scenarios, and by creating a knowledge base that contains these patterns. The adoption of this solution will facilitate the work of a fission module by giving it the most meaningful combinations of data.

3.4 Architectural Design

In this section, we will describe our proposed architectural design. The architecture is composed of five main modules ():

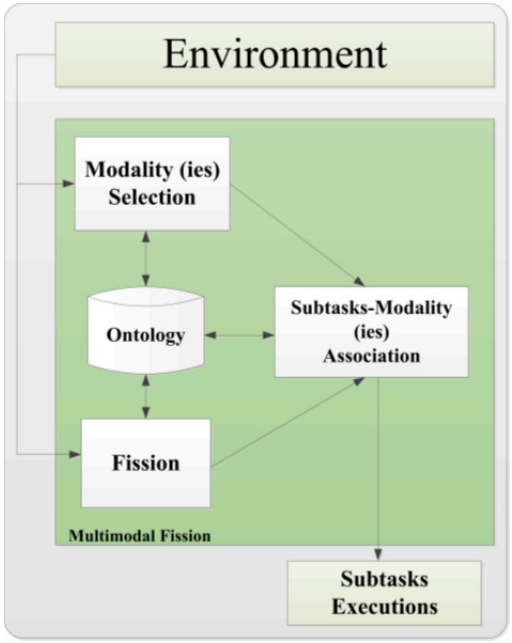


Figure 3.2 General approach of multimodal fission system

Environment: involves the physical geographical location where the multimodal system exists.

Modality (ies) Selection: the goal of this module is to select the adequate and available modality (ies) according to a specific context. This module is presented in section 5.

Fission: it is the most essential part of our architecture. This module permits to subdivide a command to elementary subtasks by finding the match with all patterns described in the ontology. These patterns are generally defined with two parts, namely, problem and solution. Therefore, we must define the problem and the solution so that we can talk about patterns.

In our case, the problem is the command parameters (the words that compose the command). The solution consists of all the possible subtasks for this command. Figure 3.3 shows an example of a pattern.

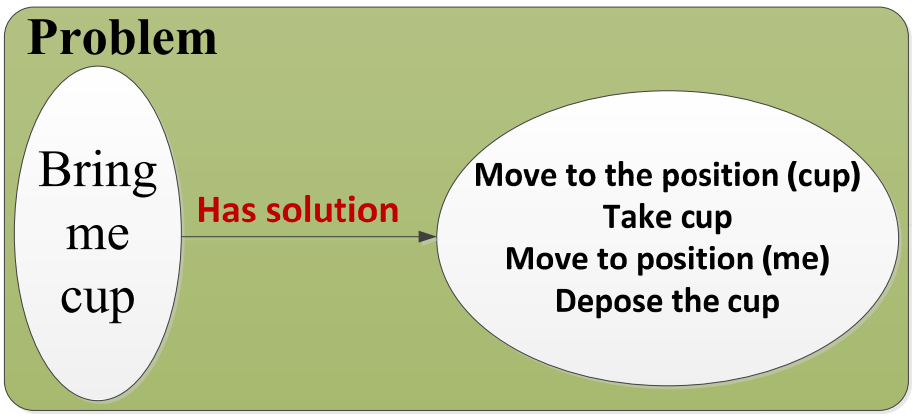


Figure 3.3 Example of pattern

The system sends a query with the problem parameters to find the matching pattern in the ontology.

Subtasks-Modality (ies) Association: its purpose is to associate for each subtask generated by the fission module the appropriate and available modality (ies). In this part, we also use patterns as predefined models that describe a modality (ies) selection. In our work, a modality pattern is composed of: a) *Problem* composed of the components: Application, Parameter, Priority, Combination, Scenario and Service and b) *Solution* composed of the chosen modality. For more details concerning scenarios selection and modalities selection, the readers can refer to (Zaguia et al., 2013b).

Ontology: is the knowledge base that describes every detail in the environment, the modality patterns and the patterns of fission that occurred previously.

3.5 Interaction context

In this work, a modality refers to the logical structure of man-machine interaction, specifically the mode by which data is presented in output as a result between a user and computer. Using natural language processing as basis for categorization, we classify output modalities into 3 different groups and for every group we present some media devices that support these modalities (Carnielli et al., 2008):

Vocal Output (VO): a sound is produced as data output: the user obtains the output by listening to it. For this modality, we can use different media such as speaker, headset and speech synthesis system ;

Manual Output (MO): the data output is presented in such a way that the user would use his hands to grasp the meaning of the presented output. This modality is commonly used in interaction with visually-impaired users. For this modality we can use Braille, overlay keyboard ;

Visual Output (VIO): data are produced and presented in a way that the user read them. For this modality we can use for instance a screen, printer, projector, TV ;

A modality is appropriate to a given interaction context if it is found to be suitable by checking the parameters of the user context, the environmental context and the system context.

3.5.1 User Context

User handicap: it affects the user's capacity to use a particular modality. We note four handicaps, namely Manual handicap, Muteness, Deafness, and Visual impairment.

User location: we differentiate between fixed / stationary locations, such as being at home or at work where user is in a controlled environment to that of a mobile location (on the go) where user generally has no control of what is going on in the environment.

3.5.2 Environmental Context

Noise level: the noise definitely affects our ability to use audio as data input or receiving audio data as output.

Brightness of workplace: The brightness or darkness of the place (i.e. to the point that it is hard to see things) also affects our ability to use manual input and modalities.

Darkness of workplace: The darkness of the place also affects our ability to use manual input and modalities.

3.5.3 System Context

The capacity and the type of the system that we use are factors that determine or limit the modalities that can be activated.
To recapitulate, a modality is considered adequate when it verifies all the parameters listed before. This is shown by a series of relationships given below:

$$VO = (user \neq deaf) \wedge (location \neq at\ work)$$

$$MO = user \neq manually\ handicapped \wedge location \neq on\ the\ go \wedge (computer \neq cellphone/PDA \vee computer \neq iPad)$$

$$VIO = user \neq visually\ impaired \wedge workplace \neq dark \vee (workplace\ very \neq dark)$$

These relationships are used in the simulation to select the appropriate modalities. For more details concerning selecting the suitable modalities, the reader can refer to (Zaguia et al., 2010a), (Zaguia, Tadj et cherif-ramadhan, 2012) and (Hina et al.).

3.6 Ontology

An ontology is the basis of the representation or the modeling of knowledge. This area is the brainchild of researchers representing various knowledge of today's world. This knowledge will be used by machines to perform reasoning. This knowledge is expressed in the form of symbols to which we give a “semantic” (meaning).
Ontology is a “formal and explicit specification of a shared conceptualization” (Gruber, 1993):

Formal specification: comprehensible by a machine ;

Explicit specification: the concepts, relations, functions, constraints, axioms are explicitly defined ;

Conceptualization: abstract model of a part of the world whom one wants to represent ;

Divided: knowledge represented is shared by a community.

Within ontology, the terms are gathered in the form of semantic concepts (or classes) as shown in the example of Figure 3.2.

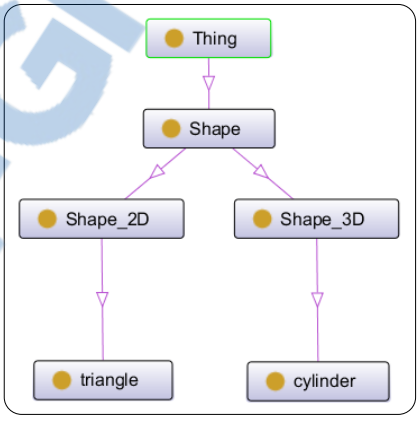


Figure 3.2 Example of ontology for the Shape

Ontology allows:

- To give a shared vocabulary to describe a field: for example (Figure 3.4) the Shape_2D concept represents the class of the shape 2 D ;

- To provide primitive typing classes and relations: Shape_2D is a subclass of Shape and triangle is an authority of the Shape_2D concept ;

- To reason: infer new facts from those we have already. It follows that triangle, in the previous example, is also an instance of concept Shape.

There exist several languages and tools to present ontology. Some of early languages are Ontolingua (University, 2013b) and OKBC (University, 2013a). Recent ones, based on xml, include RDF (Huajun et al., 2006), DAML+OIL (Horrocks, 2002) and OWL (Antoniou et Harmelen, 2009). We can also find ontology development tools such as the server in university of Stanford Knowledge Systems Laboratory, Protegé (Knublauch et al., 2004), OilEd (Bechhofer et al., 2001) and OntoEdit (Sure, Angele et Staab, 2002).

The purpose of ontology is to model a set of knowledge in a given field with a form usable by the machine. It provides a representative vocabulary for a given domain, a set of definitions and axioms that constrain the meaning of the terms of this vocabulary in a sufficient manner to allow a consistent interpretation of the data represented using this vocabulary. Ontology is used as a way to formalize information to obtain a knowledge base.

In this section we present the most essential of our ontology. It details the *home context*. This ontology can be upgraded to target other contexts such as hospitals, workplace, etc. Assuming that our system will help or assist a user in the house.

In our case, we will use Protegé (Huiqun, Shikan et Junbao, 2012) to develop our ontology. It is a free tool and the most widely used ontology editor. An open-source, it was developed by the university Stanford. It has evolved since its first versions (Protected-2000) to integrate from 2003 the standards of the semantic Web and in particular OWL. It offers many optional components: reasoners, graphical interfaces.

The general view of our ontology is presented by Figure 3.3. It is composed of several classes:

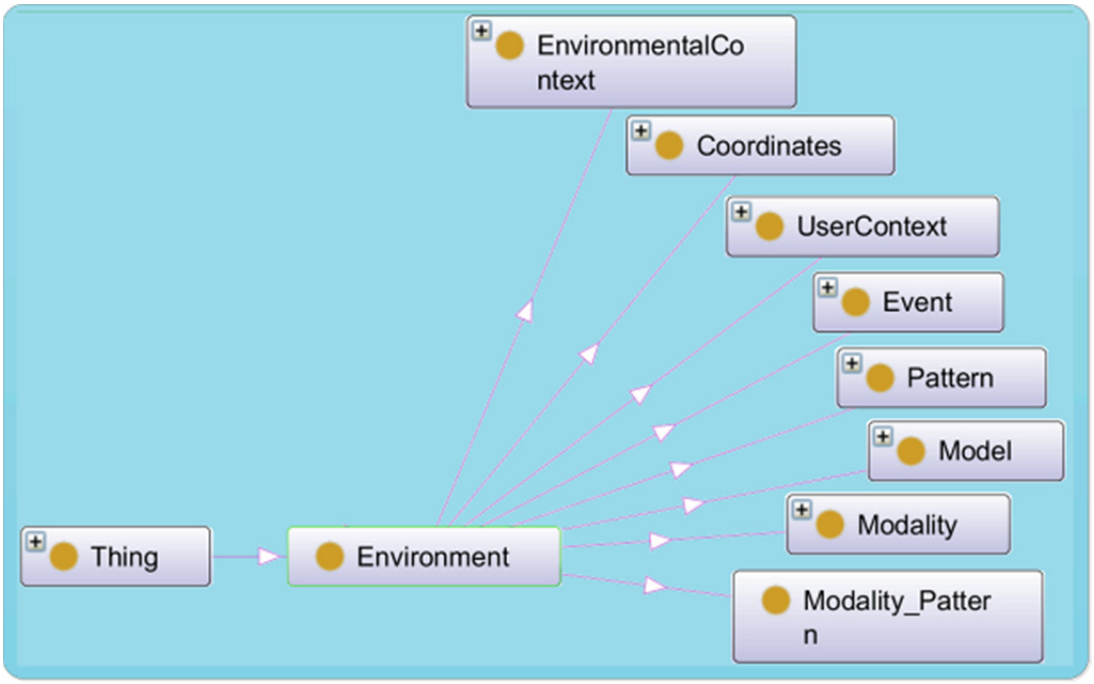


Figure 3.3 General view of ontology

Environmental Context: It describes the state of the environment, such as determination of the noise level. It is understood that the use of the audio modality is affected by this information. If the noise level is high, the audio modality will be disabled. As we can see in our ontology (Figure 3.4) the vocal modality is only active when the noise level is average or low.

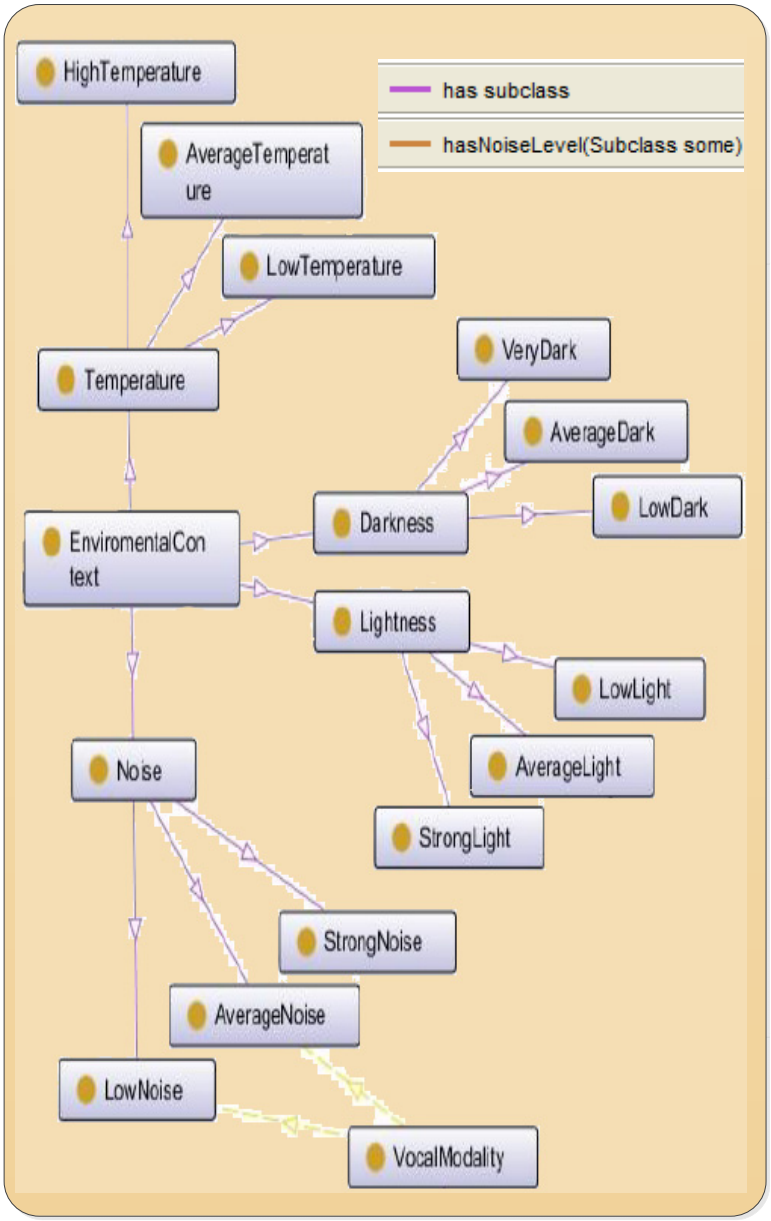


Figure 3.4 Environment context

Coordinates/Place: Used to locate various objects, places and people in the environment (Figure 3.5).

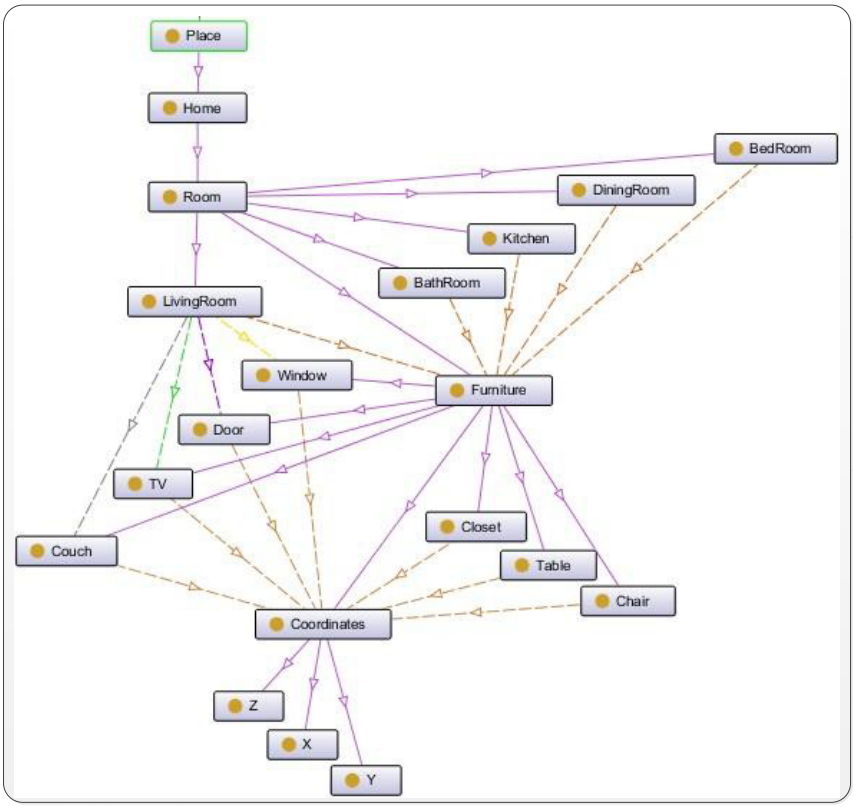


Figure 3.5 Place concept and its subclasses

UserContext: It describes the location and status of the user. It identifies the user's ability to use certain modalities. For example, the system disables the display modality if it detects that the user is visually impaired and disables the vocal modality if it detects that the user is in a library.

Modality: It contains various modalities that can be used by the system to present a given subtask.

Model: Contains a set of models that represent different combinations of events for different scenarios (user commands). This class will allow us to validate the meaning and the grammar of a command. Here, it presents the problem for a given pattern.

Pattern_Fission: Contains all possible sub-tasks for an order. It presents the solution for a given pattern.

Modality_Pattern: Allows us to select the appropriate modality for each subtask.

Event: It is the event that triggers the fission process. It presents our complex command.

The following describes in details the most important class developed for our system.

3.6.1 Modality class

As we can see in Figure 3.6, this class presents the different output modalities that we can use to present our data/information.

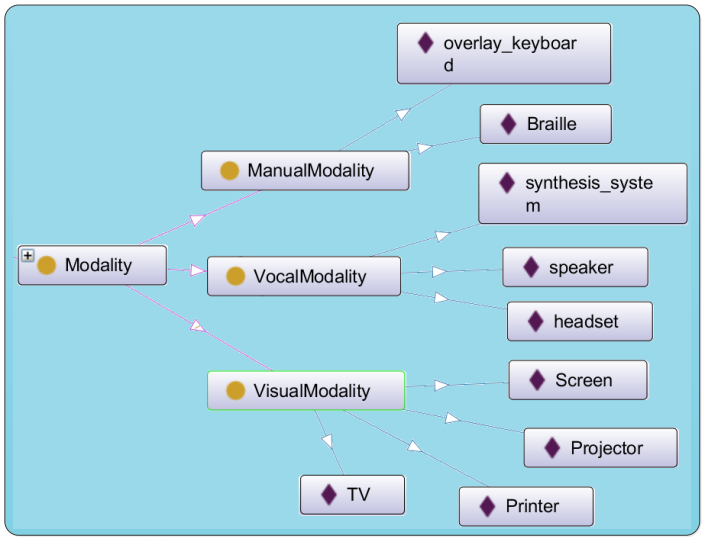


Figure 3.6 Modality class

It composed of three main subclasses {ManualModality, VocalModality, VisualModality} and every subclass has the adequate medias.

3.6.2 Event class

This class presents the classes that can form a command (Figure 3.7), it presents the possible combination of classes to form a command:

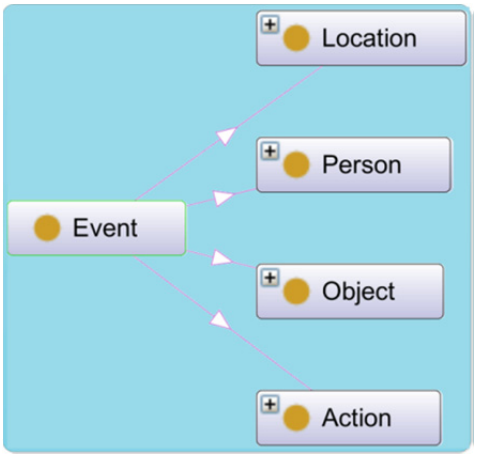


Figure 3.7 Event class

1) **Action**: It presents the verbs that the command can contain (vocabulary). We divided it on three categories:

$$Class\ Action\ =\ \{ActionForLocation,\ ActionForPerson, ActionForObject\}$$

a) **ActionForLocation**: It presents the verbs that refer to places. Here are some of these verbs:

$$Class\ ActionForLocation\ =\ \{locate, search, check, come, find, \dots\}$$

b) **ActionForPerson**: it presents the verbs for persons. For instance:

$$Class\ ActionForPerson$$
$$=\ \{replay, call, walk, demand, answer, ask, bring, help, \dots\}$$

c) **ActionForObject**: these are verbs that act on objects. It has been divided in tow classes to allow us to manage the meaning of a command, for example "move the wall" should be rejected.

$$Class\ ActionForObject = \{ActionForMovableObject, ActionForNonMovableObject\}$$

d) **ActionForMovableObject**: represents the verbs for objects that can be moved.

$$Class\ ActionForMovableObject = \{drag, keep, take, cut, give, move, \ldots\}$$

e) **ActionForNonMovableObject**: represents the verbs for objects that can be moved.

$$Class\ ActionForNonMovableObject = \{lock, clean, unlock, close, open\}$$

2) **Location***:* it presents different locations that we can find in a command. We subdivided it in three categories:
$Class\ Location = \{Indoor, Oudoor, IntendedLocation\}$

a) **Indoor**: we defined locations within the house, such as:

$$Class\ Indoor = \{dinnerroom, bathroom, livingroom, corner, bedroom, kitchen\}$$

b) **Outdoor**: we defined the locations outside of the house.
$Class\ Outdoor = \{backyard, road\}$

c) **IntendedLocation**: presents the prefix for locations. These permit to precise the position of an object. For example “put the pen **inside** the box.”

Here some these prefix:

$$Class\ IntendedLocation = \{after, outside, to, under, right, behind, on, before, inside\}$$

3) **Object**: it presents the different objects that we can use.

   a) **AverageObject**: it presents the medium objects in the environment, such as

   $$Class\ AverageObject = \{microwave, box, desk, computer, chair, television, table, \dots\}$$

   b) **ElectronicObject**: it presents the electronic objects. For instance:

   $$Class\ ElectronicObject = \{television, dvdplayer, microwave, \dots\}$$

   c) **HeavyObject**: it presents the heavy objects. For example:

   $$Class\ HeavyObject = \{oven, sofa, closet, window, refrigerator, \dots\}$$

   d) **Food**: it presents the foods. For example:

   $$Class\ Food = chicken, cheese, banana, fish, carrot, orange, bread, meat, apple, \dots\}$$

   e) **Liquid**: it presents the liquid for example:

   $$Class\ Liquid = \{café, jus, thee, soda, water, milk, \dots\}$$

   f) **MovableObject**: it presents the movable objects, such as:

$Class\ MovableObject = \{bed, sofa, oven, closet, \dots\}$

g) **NonMovableObject**: it presents the non-movable objects, for example:

$Class\ NonMovableObject = \{door, wall, window, \dots\}$

h) **ObjectForFood**: it presents the objects for food, for example:

$Class\ ObjectForFood = \{plate, glass, fork, knife, spoon, \dots\}$

i) **ObjectForLiquid**: it presents the objects for liquid, such as:

$Class\ ObjectForLiquid = \{bottle, cup, \dots\}$

j) **SmallObject**: it presents the small objects. For example:

$$Class\ SmallObject = \{watch, pants, tshirt, flower, glass, , mobile, ball, paper, \dots\}$$

4) **Person**: it presents the relations between persons. For instance:

$Class\ Person = \{uncle, son, friend, niece, brother, daughter, mother, cousin, \dots\}$

3.6.3 Grammar Model

This class will allow us to validate the meaning and the grammar of a command, we have defined several models. Each model includes two / several subclasses of the class "Event" in a predetermined order. For example Figure 3.8.

Model01 is composed of four subclasses: ActionForMovableObject, SmallObject, IntendedLocaion and AverageObject. These subclasses are defined in the model in a predefined order as seen in Figure 3.8:

$$ActionForMovableObject\ \mathbf{hasNext}\ SmallObject\ \mathbf{hasNext}\ IntendedLocation$$

$$\mathbf{hasNext}\ AverageObject$$

An example of this model: "Put the pen on the table". In this case:

$ActionForMovableObject = Put$

$SmallObject = pen$

$IntendedLocation = On$

$AverageObject = Table$

In the following, we show some other models we have modeled:

Model02:

$ActionForMovableObject\ \mathbf{hasNext}$

$MovableObject\ \mathbf{hasNext}\ IntendedLocation\ \mathbf{hasNext}\ NonMovableObject$

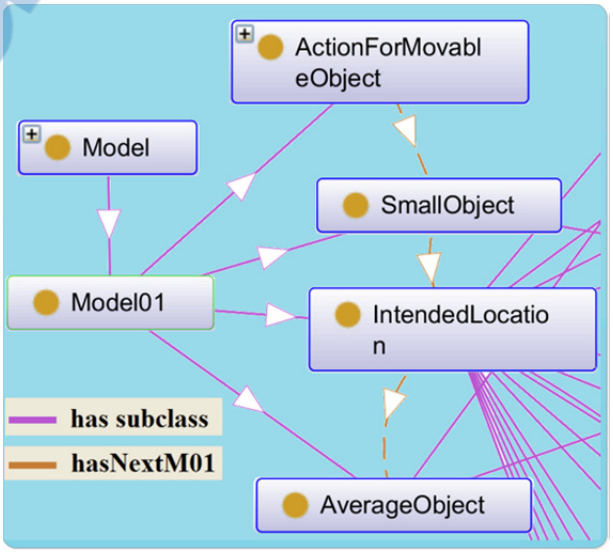


Figure 3.8 Example of a model

For example “Put the chair behind the wall”

Model03:

$$ActionForMovableObject\ \mathbf{hasNext}\ MovableObject$$

For example “drop the cup”.

Model04:

$$AFMO\ \mathbf{hasNext}\ MO\ \mathbf{hasNext}\ IL\ \mathbf{hasNext}\ P$$

For instance “give the pen to my father”

Model05:

$$AFNMO\ \mathbf{hasNext}\ NMO\ \mathbf{hasNext}\ LO$$

For instance “close the door of the kitchen”

These models enable us to process a large variety and complex commands.

3.6.4 Fission Pattern class

This class describes the different scenarios. These scenarios saved in patterns form which are mainly composed of two parts problem and solution, as detailed in (Zaguia et al., 2013b). Figure 3.9 shows an example of pattern.

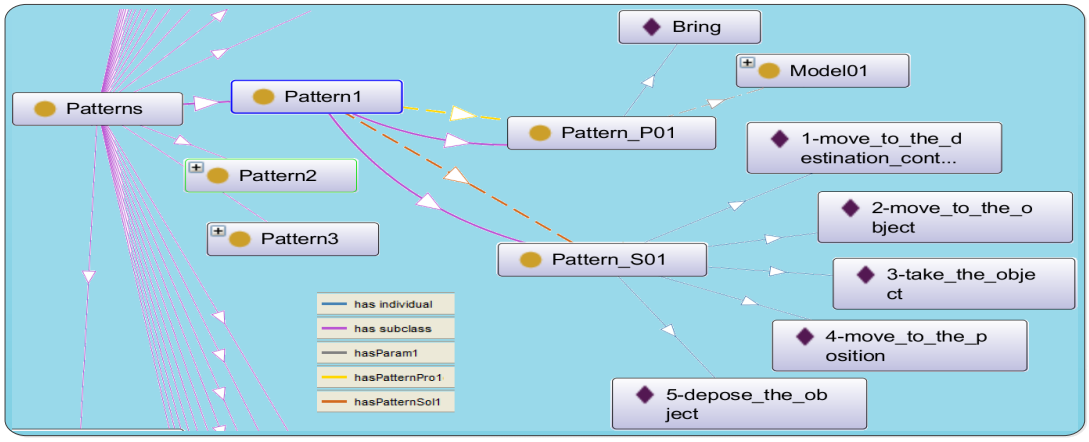


Figure 3.9 Example of Fission Pattern

As we can see class « *pattern1* » has two subclasses (Problem, Solution):

hasPatternPro1 (yellow Arrow) *class Pattern_P01* with parameters {Bring, Model01 (as shown in Figure 3.8)} ;

hasPatternSol1 (orange Arrow) *class Pattern_S01* with parameters {1-move to the destination context, 2-move to the object, 3-take the object, 4-move to the position, 5-depose the object"}.

3.6.5 Modality Pattern class

Here we present the class of Modality_Pattern (Figure 3.10). That permits, to select the adequate modalities for a given subtask. As we mention in the previous section, this pattern is quite similar to Fission *Pattern.*

Figure 3.10 shows an example of Modality_Pattern, the class *Modality_Pattern1* is composed of two subclasses:

Pat_Mod_Prob1 (Problem)

$$= [subclass = \{move\ to\ position\}, combination\ = \{serial, parallel\},$$
$$Application = \{Robot\}]$$

$$Pat\_Mod\_Sol1\ (Solution) = \{manual, visual, vocal, mobility\ mechanism\}$$

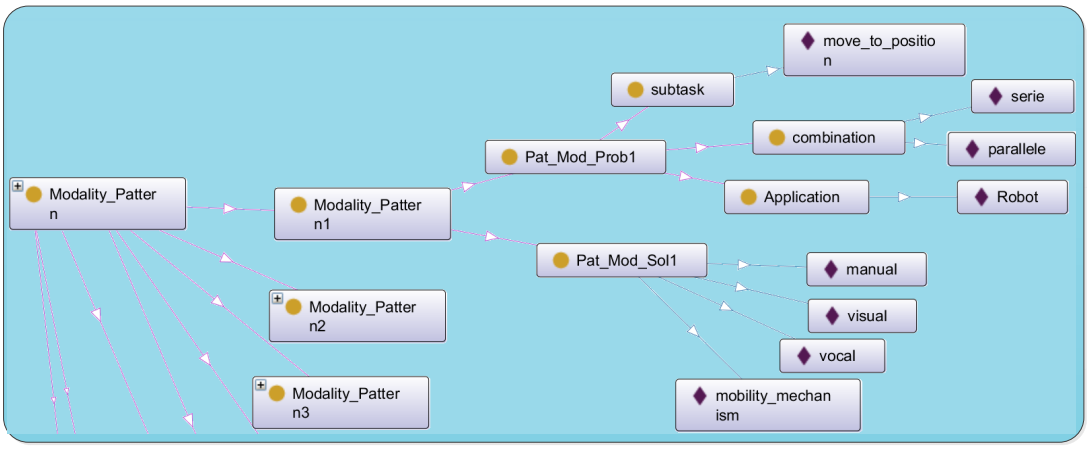


Figure 3.10 Example of Modality_Pattern

3.7 Fission Algorithm

In general, the fission rule is simple: if a complex command (CC) is presented, then a set of sub-tasks with the suitable modalities (and its parameters) are deducted.

Multimodal fission can be represented by the function:

$$f: F \rightarrow ExK$$
$$\forall cc\ \in\ F, \exists\ STi\ \in K\ and\ MOj\ \in\ E,$$
$$f(STi, MOj) = cc$$
With: $i \in [1..n]\ et\ j\ \in [1..m]$

$$f: CC = \sum_{i=1}^{n} ST_i \left(\left\{ \bigcup_{j=1}^{k} MO_j \right\}, \left\{ \bigcap_{j=1}^{l} MO_j \right\} \right) \quad (3.1)$$

With: ST_i = sub-task i.

MO_j= output modality i.

cc = complex command.

l and k are different from m and n because it depends on the sub-tasks. For example, for some sub-task we will use just two terms even if we have three modalities available.

In equation (1), the symbol ⋃ indicates that we can use either one or several modalities to present a sub-task. For example, if we present a text to the user, we use audio or display. The symbol ⋂ indicates that we use the available modalities together to present a sub-task.

Figure 3.11 describes the steps involved in the fission process. In this diagram, *n* numbers of commands serve as an input to the system. The steps undertaken are as follow:

Step-1: the system extracts every word from the command ;

Step-2: for every word, the vocabulary stored in the *Vocab* ontology is checked ;

Step-3: from each $word_i$ is extracted $vocab_i$ and is concatenated in the same order as in the original command. The model of the command is therefore obtained ;

Step-4: a query is sent to the ontology *Grammar Model* to look if the model is in the ontology;

Step-5: if the model is found, we proceed with step 7 otherwise we proceed with step 6 ;

Step-6: the command is not valid and a feedback is sent to the user ;

Step-7: a query is sent to find matching pattern fission from predefined patterns stored in the *Pattern Fission* ontology ;

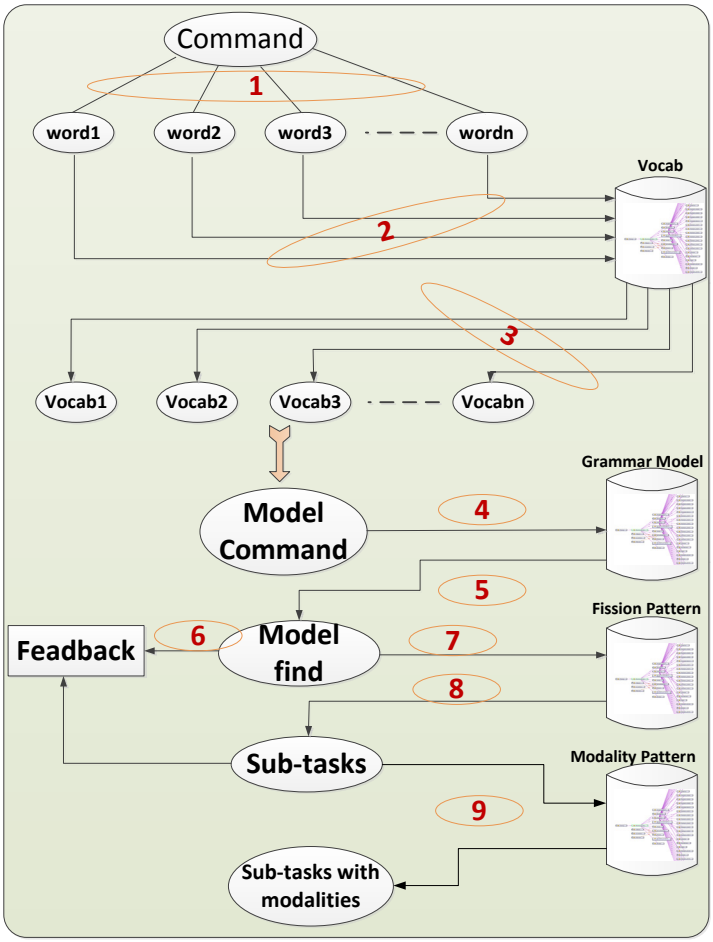


Figure 3.11 Stages of fission process

Step-8: if no matching pattern is found, a feedback is sent to the user ;

Step-9: for every subtask we associate the adequate and the available modality (ies). This is done by sending a query to find the matching pattern modality.

This algorithm is divided in two parts: the purpose of the first part (Figure 3.12) is to check if the command is valid. The second part concerns the fission process (Figure 3.13).

The inputs of the fission algorithm are the command CC and the model created with concatenation of all $vocab_i$ of the command (Figure 3.13).

We create the pattern problem; composed of the {model and the action verb of the command}; and then we search the matching of the pattern problem in the ontology (getPatternSolution(pattern-problem)).

If a solution is found, subtasks are created and for each one, the appropriate modality (ies) is determined (getSubtasksModalities (subtask)).

If we find a similar matching (PartMatchingModel (pattern-problem)), we try to find the missing subtask (createSubtasksMissing()) and we send feedback to the user.

Grammar and meaning

```
Input: command complex: cc
Begin
        For  each word w in cc do
                if w in Ontology Then
                   check = check+ getclass(w)
                End if
        End For

        if MatchingModel (check) Then
                FissionProcess(cc, check)
        else
                Feedback
        End if
End.
```

Figure 3.12 Fission Algorithm (Grammar and meaning)

```
FissionProcess

Input: cc, model
Create query : pattern-problem
If MatchingModel (pattern-problem) then
        getPatternSolution (pattern-problem)
        For each subtask do
          getSubtasksModalities (subtask)
        End For
Else if PartMatchingModel (pattern-problem)
then
        getPatternSolution (pattern-problem)
        createSubtasksMissing()
        Feedback
Else
        Feedback
End if
End
```

Figure 3.13 Fission Algorithm (Fission Process)

We will follow these steps for our simulation in the next section.

3.8 Application Scenario and simulation

To understand the mechanism of fission we will demonstrate a simulation of a scenario.
In this scenario, we assume that in our multimodal system, a robot moves an object from its initial position to another location. For instance “Bring me the cup”.

We first model and simulate the architecture defined in . The steps of each strategy are modeled using the colored Petri Net (Jensen, 1987) formalism and simulated using the CPN Tools (Zhu, Tong et Cheng, 2011).

Figure 3.16 shows the general view of the architecture. It is composed mainly by 8 modules:

Generator*:* this module generates events as random numbers to select a command in the *place* "command". As shown in Figure 3.16, the system processes the command "Bring me the cup".

T_parser: this module decomposes the command into words.

T_Fission: its role is to divide the command to elementary subtasks.

T_Grammar: permits to verify the meaning and the grammar of the command.

Modality(ies) Selection: this module allows to select the available modalities depending on the state of the environment.

T_Subtask-Modality_Association: as its name indicates, it associates for each subtask the appropriate modality (ies).

Ontology: it is a container that stores the patterns as ontology concepts, the models and the vocabulary.

The diagram in Figure 3.15 demonstrates a Petri net showing the parser processing. As we can see, this module has as an input, the command of type string and the output a list of words. In transition "*T_Parser",* the system creates a list of all the words in the command and then the transition "*Add indication end of command"* adds an indication at the end of the command to differentiate it with other commands.

The diagram in Figure 3.18 shows how the grammar module interacts with the ontology to verify the validity of the command. This module has as the first input a list of strings obtained from the parser and the second input, is a set of two elements of type string and Boolean, obtained from the ontology. The string type refers to the model of the command found in the ontology and the Boolean one determines if the command is valid or not.

We also obtain two outputs: the first one is a feedback module to prevent the user in case of an invalid command. The second output is connected to the fission module when the command is valid.

As we can see, the system receives the list of words of the command and sends them, one by one, to the ontology (transition “send word”) by checking the length of the list and the number of the words that are already sent (places “com” and “command lengh”).

When this module receives the answer from the ontology (place “Model of command”), it verifies the value of the Boolean element “modelGram”: a) if true: sends the model “concatGramF” to the fission module (place “to Fission”) otherwise 2) sends to the feedback module (place “to feedback”).

The diagram in Figure 3.17 is a continuation of Figure 3.18. This module sends the words to “ontology_Vocab” and then concatenates the vocab for every word to create the model of the command. As we can see in the place “word from grammar”, we check the equivalence of vocabulary of all the words of the command.

Figure 3.18 shows the steps to get the vocab of every word. All the vocabularies are stored in the place “vocabulary”. The system searches for every word the equivalent vocab, and sends the result. As we can see in Figure 3.18, the result sent for “Bring” is “AFMO”.

After the model of the command is created, the system sends the model of the command to the ontology to look if this model exists in the ontology (Figure 3.19). The model of the example “Bring me the cup” is “AFMO P MO”. In the place “Model Test” we can see the result of searching (“AFMO P MO”, true). The command is therefore verified grammatically and has a correct meaning.

After the verification is done, the fission process begins. The fission module sends a query to the ontology to find the matching pattern stored in the ontology. As we can see in

Figure 3.20, in the place “pattern”, the parameters of pattern problem are “modelP” and “actionP”. These parameter are compared with the parameters of command “(queryPatA,queryPatM)” to find the matching pattern. The solution pattern is “supList”.

The result of our command is ["1-move to the destination context","2-move to the object", "3-take the object","4-move to the position", "5-depose the object"] as shown in
Figure 3.20. This result represents the subtasks for our command. Adding the first subtasks “move to the destination context” will allow as managing orders for other contexts position. For our example the destination context is “kitchen” since the cup is usually in the kitchen.
After we get the subtasks, the fission module send the results to the T_Subtasks-Modality(ies)_association. Selection of the available modality(ies) is performed according to state of the environment.

Figure 3.23 shows the process of selection of modality(ies). There are three types of modalities: “audio”, “visual” and “manual”. The system starts getting the information from the sensors in the environment, in Figure 3.23 we simulate it by the places:

- “environmental context for light” is set to a value from 1 to 10 ;
- “environmental context for noise” is set to a value from 1 to 10 ;
- “Manual Modality”.

The system chooses randomly the value of noise and the brightness in the environment. If the noise is under 5 we activate the audio modality (ies) otherwise we disable it (Transition: “*audio Contexts Checking*”) and if the location of the user is for example in the library, we deactivate the audio automatically (place “User location”). If the user is mute or deaf, the audio modality is deactivated (place “User Profile”). If the light is higher than 5, we activate the visual modality (ies), otherwise we disable it (transition: “Visual conext test”).

In our case, we can see that the only activated modality is (“visual”, “ ”, “ ”). This is performed by checking all the active modalities by the transition “context verification”.

Figure 3.24 shows the final processing of our simulation: modality (ies) and subtasks associations. This module receives as an input, the list of subtasks presented by the place “list Subtasks” and the list of modalities presented by the place “List Modalities”.
In our case, the first input is the first subtask “move to the destination context” and the second input is (“visual”,“ ”, “ ”).

Figure 3.25 is the continuation of Figure 3.24 showing the result of the association of modalities. For the first subtask, the result is ("1-move to the destination context",["mobility mechanism", "TV(on)", "printer(on)"]) since the system detects only visual modality, it can show the track on TV and printer as they are ON and the system will use "mobility mechanism" as services related to the output modalities to move.

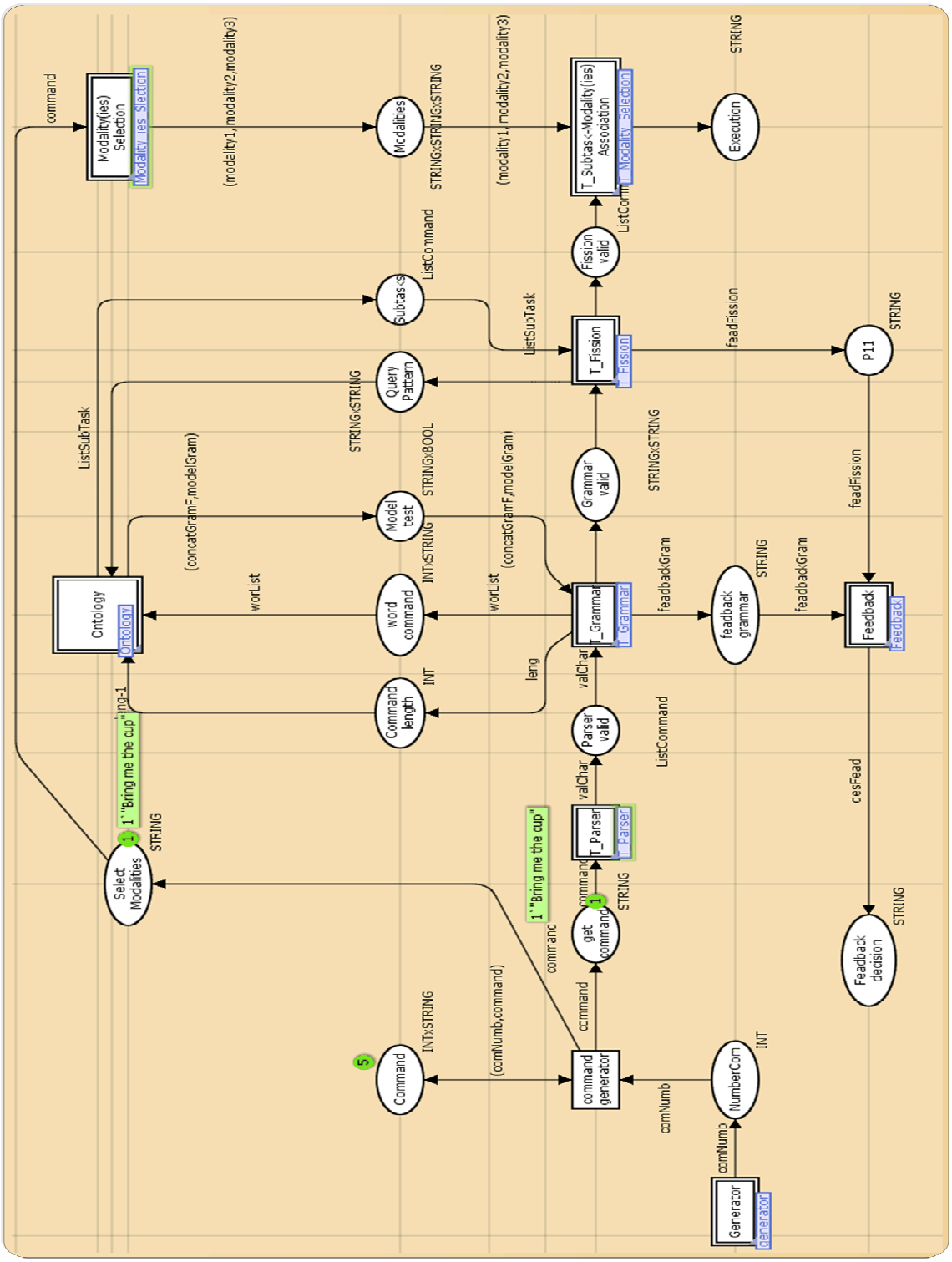


Figure 3.14 General view of our

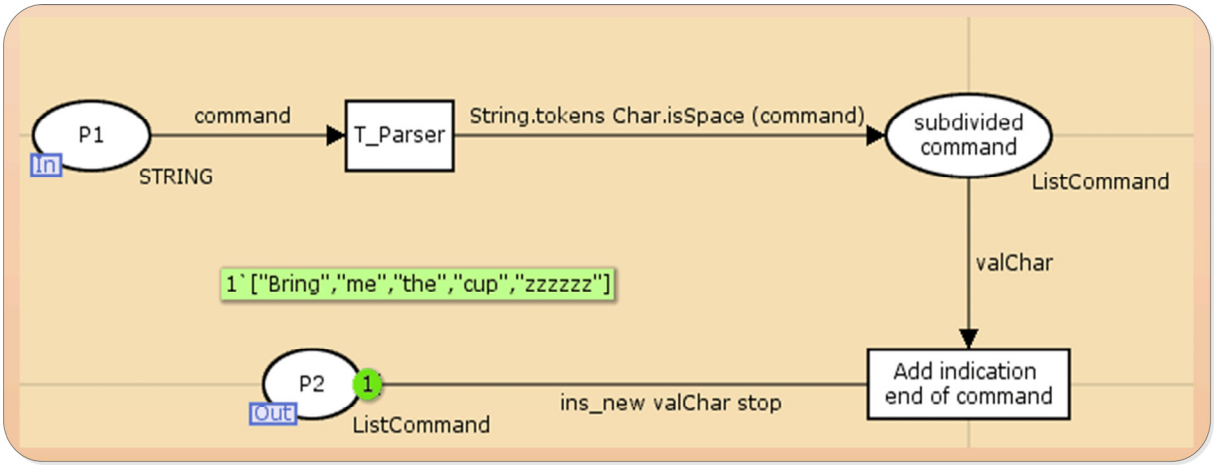


Figure 3.15 Parser processing

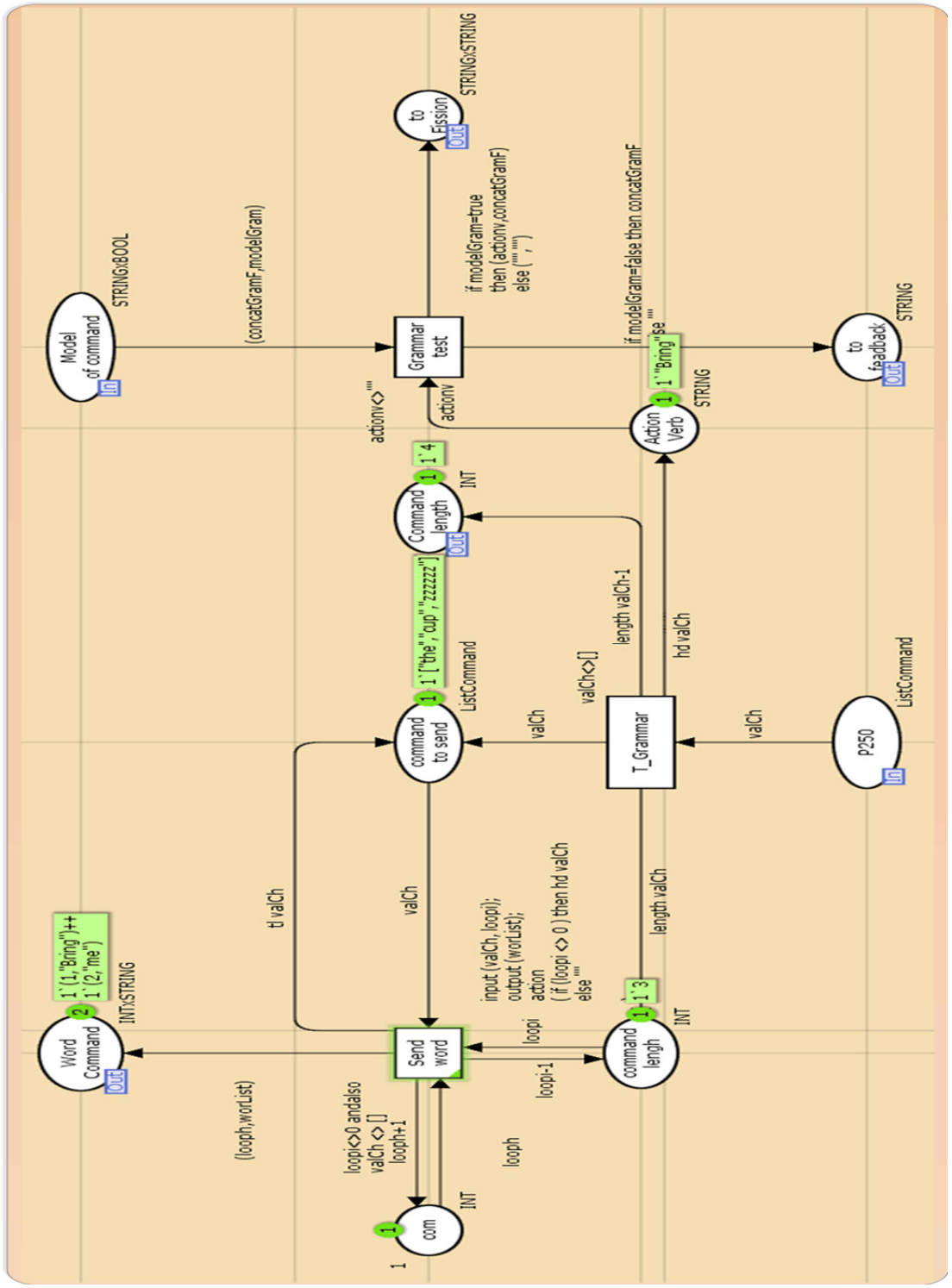


Figure 3.16 Grammar

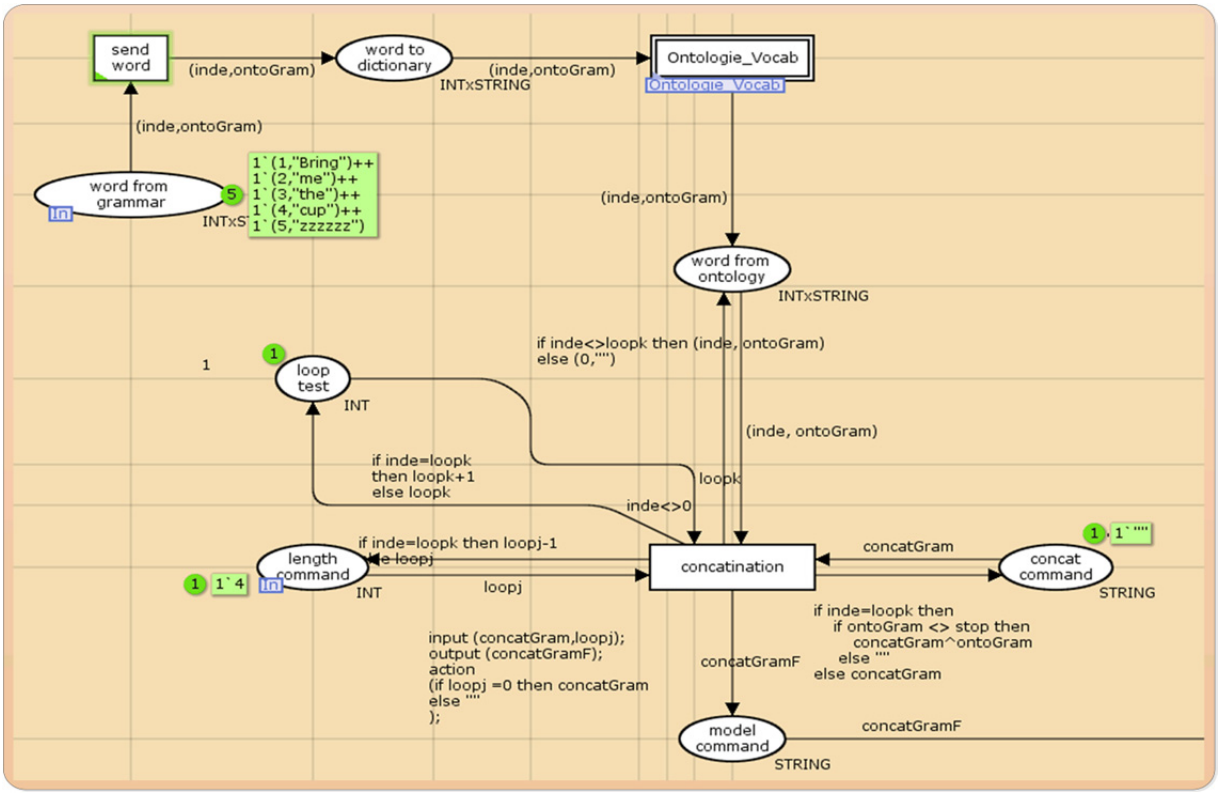


Figure 3.17 Ontology-Processing

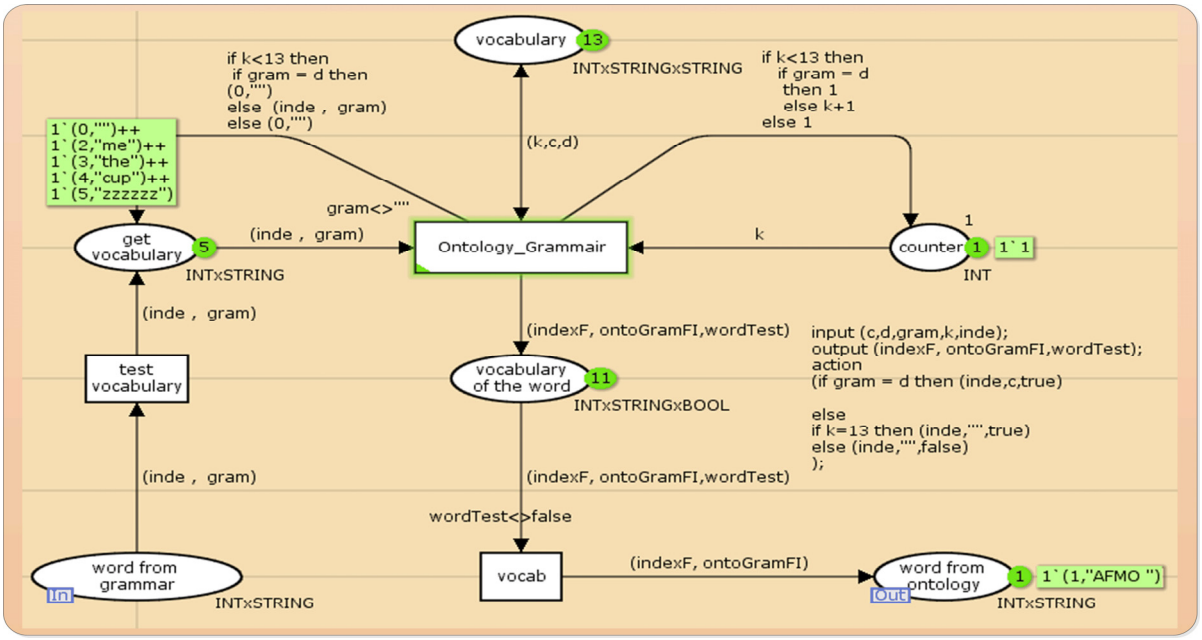


Figure 3.18 Vocabulary processing

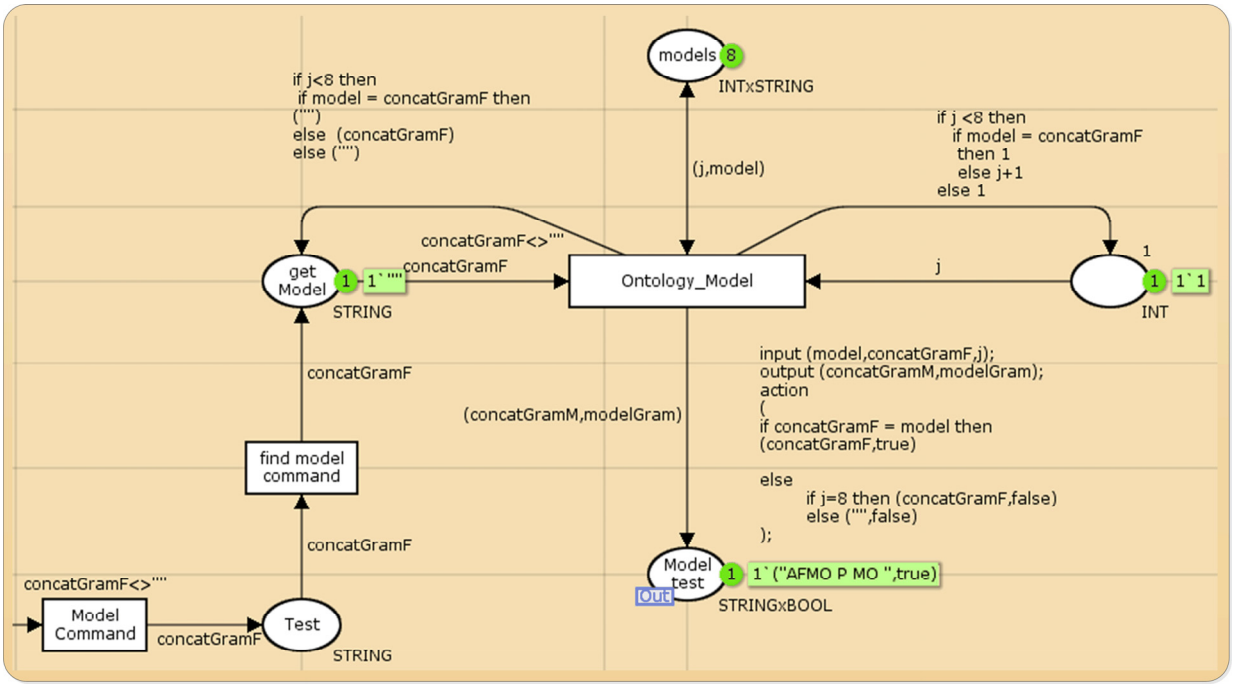


Figure 3.19 Model Processing

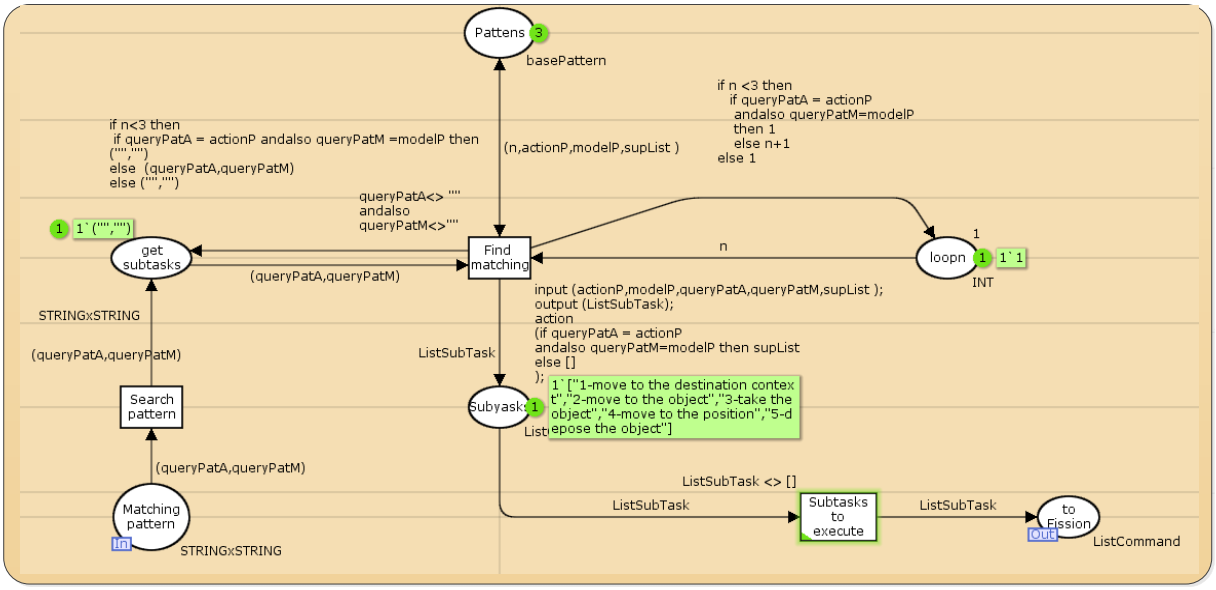


Figure 3.20 Matching Process

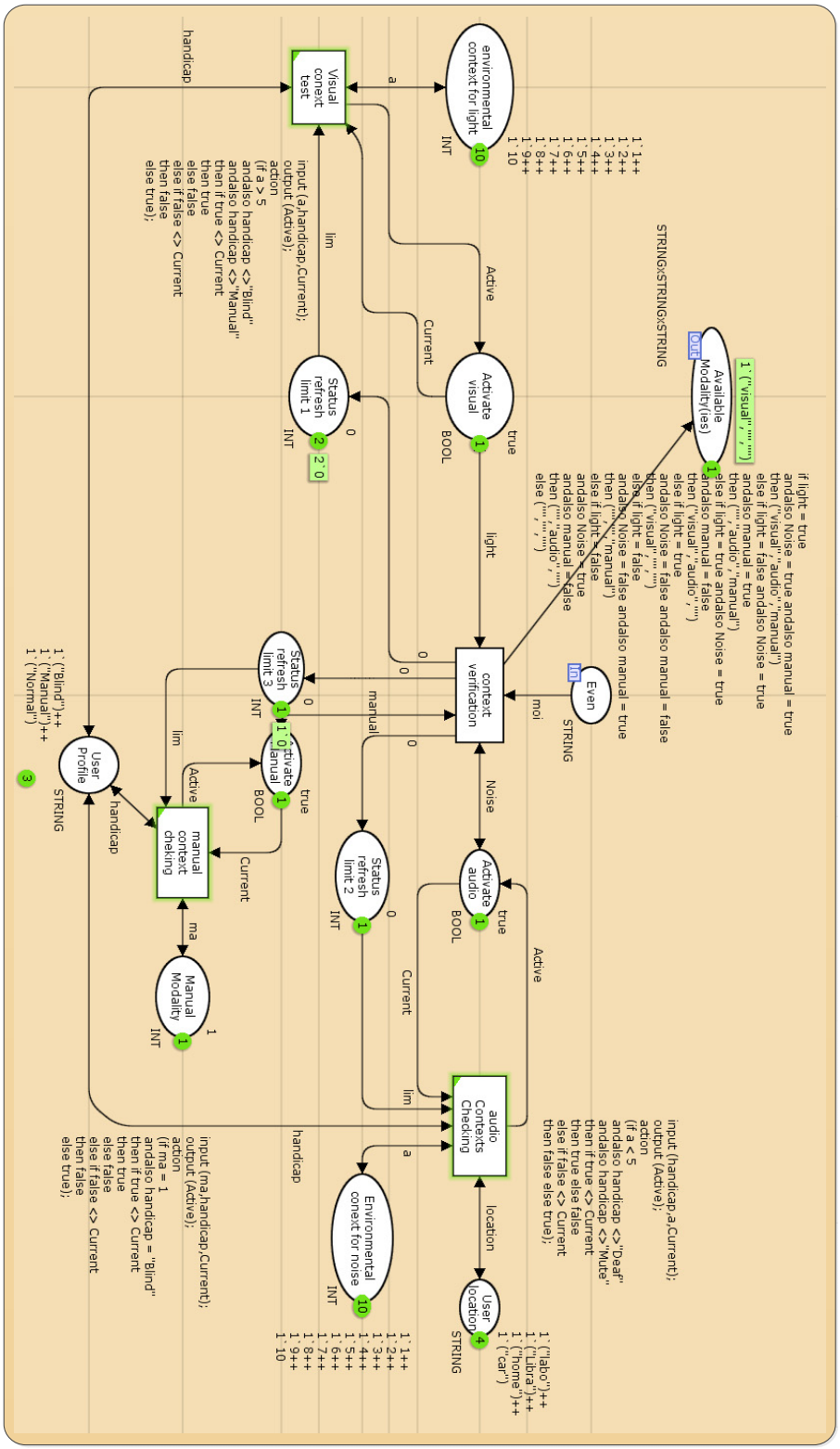


Figure 3.21 Modality (ies) selection process

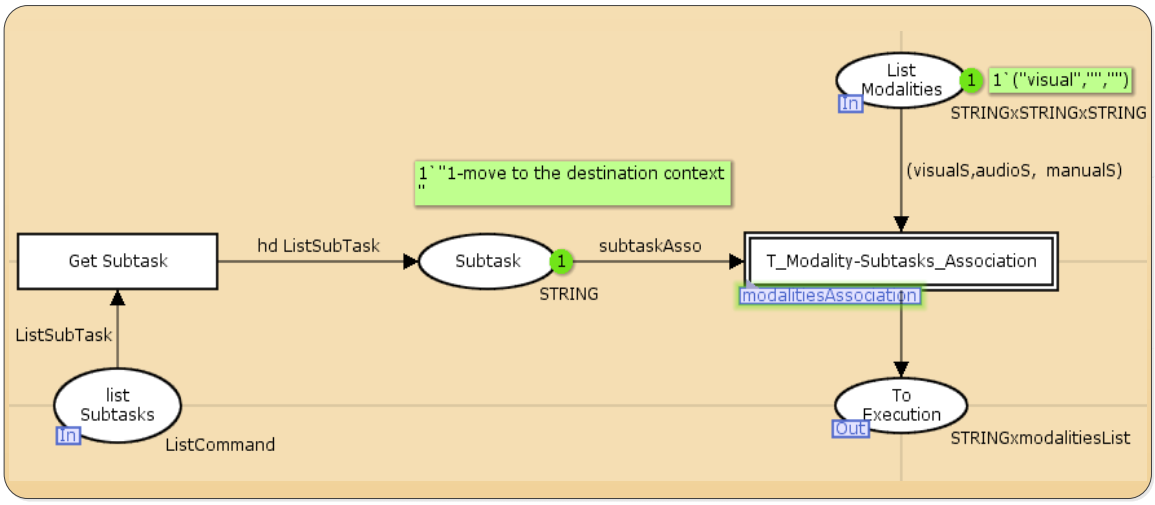


Figure 3.22 Modality (ies) -Subtasks association processing 1

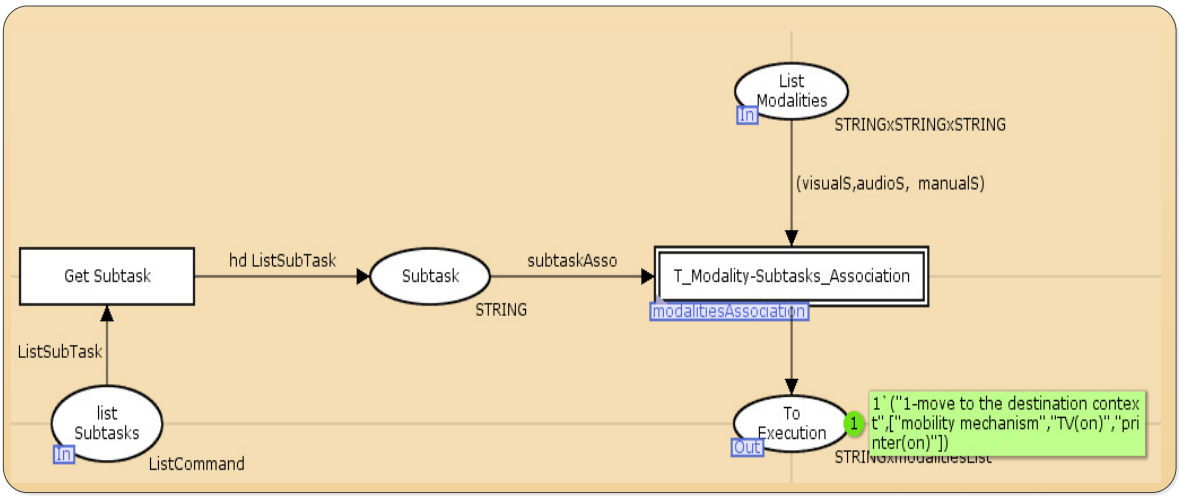


Figure 3.23 Modality (ies) -Subtasks association processing 2

3.9 Conclusion

The role of ontologies is to improve the communication between human, but also between human and machines and finally between machines.

In this paper we demonstrate the reasons of the use of ontology such as sharing common understanding of the structure of information, enabling reuse of domain knowledge and

making domain assumptions explicit. And we present our ontology to solve our problems that consists to model our data, make it dynamic, flexible and easy to update.
We presented architecture able to identify different modalities of output and split data on these modalities using patterns stored in the knowledge base.

The architecture was modeled using colored Petri net formalism and simulated with the CPN-Tools.

There are many fields where improvements of the human-system interaction are open to exploration. In fact, a system which is able to take advantage of the environment can improve interaction. This can reach an extent at which the system (robot / machine) is able to use human beings' natural language. We believe that this work contributes to the advancement of fission research in the field of multimodal human-machine interaction.
This system provides a good level of autonomy and a good capacity for decision-making can be an effective way to help or assist user and especially in inaccessible places, or considered as a danger to humans.

CHAPITRE 4

CONTEXT-BASED METHOD USING BAYESIAN NETWORK IN MULTIMODAL FISSION SYSTEM

Atef Zaguia[1], Chakib Tadj[1], Amar Ramdane-Cherif[2]

[1]MMS Laboratory, Université du Québec, École de technologie supérieure

1100, rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3 Canada

[2]LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France

This article is submitted to the Journal of Journal on Multimodal User Interfaces. Janvier 2014

Résumé

L'avancement technologique actuel a créé la nécessité de produire des machines plus puissantes, faciles à utiliser et permet de répondre aux besoins des utilisateurs. Pour atteindre cet objectif, les systèmes multimodaux ont été développés pour combiner plusieurs modalités en fonction de la tâche, les préférences et les intentions communicatives des utilisateurs.

Dans cet article, nous présentons une nouvelle approche pour surmonter le problème d'incertitude ou d'ambigüité durant le processus de fission dans un système multimodal. La méthode utilisée est basée sur le contexte en utilisant un réseau bayésien. Nous présentons également une architecture modulaire et distribuée, ce qui est très utile dans les systèmes multimodaux.

Mots clés : fission multimodal, réseau bayésien, modalité, pattern, contexte d'interaction.

Abstract

The current technological advancement has created the need to produce machines more powerful, easy to use and to meet the needs of users. To achieve this goal, multimodal systems have been developed to combine multiple modalities depending on the task, preferences and communicative intentions of the users.

In this paper, we present a new approach to overcome uncertain or ambiguous knowledge during the fission process in multimodal system. This approach is context-based method using Bayesian network. We also present a modular and distributed architecture, which is very useful in multimodal systems.

Keywords: Multimodal fission; Bayesian network; Modality; Pattern; Interaction context;

4.1 Introduction

Inspired by human communication, researchers in computer science and computer engineering devote a significant part of their efforts on communication and interaction between man and machine. They aim to create more efficient machines, easier to use and which can meet the needs of users. To achieve this goal, the machines must be able to recognize the user's commands and communicate similarly to human with the use of different ways of communication such as words, gestures, voice, etc.

Multimodal systems are an effective solution to this problematic. They allow the use of different modalities (gesture, vocal, etc.) to interact with machine, system, applications, etc. Nowadays, we are seeing a movement towards convergence of human-machine / machine-machine interactive systems to multimodal systems that enhance the interaction. These systems use the adequate modalities according to the user's preferences, his/her skill level and the nature of the task.

Generally, these systems have a process of understanding. They must also have the ability to interpret data from multiple input modalities generated from sensors and devices (camera, microphone, etc.). This is known as fusion process (Zaguia et al., 2010b) and (Portillo, García et Carredano, 2006). The resulting command is then executed in the output device (screen, speakers, projector, etc.). This is known as fission process (Zaguia et al., 2013b), (Zaguia et al., 2013a) and (Costa et Duarte, 2011).

A well-known example of these systems is the Bolt system "Put That There" (Bolt, 1980a), where the author used gesture and speech to move objects. Another simple example of multimodal system is the use of the ringer and brightness to indicate a call or GPS which provides visual and audible indications.

These systems improve the recognition and the understanding of the environment command (user, robot, machine, etc.) by the machine.

This paper focuses on the fission process. We present a new methodological solution by modeling an architecture that facilitates the work of a fission module, by defining an ontology that contains different applicable scenarios and describes the environment, where a multimodal system exists. We detail in this article the use of a probability method, namely the Bayesian network (BN), to perform multimodal fission. The BN is helpful in the case of ambiguity or uncertainty.

This paper discusses these solutions by highlighting the architectural design of the proposed solution. The rest of this paper is organized as follows. Section 4.2 presents the most important research works related to ours. Section 4.3 presents the proposed architecture. Section 4.4 describes the fission algorithm. Section 4.5 discusses the BN. Section 4.6 presents a concrete simulation example using CPN-Tools. The paper is concluded in section 4.7.

4.2 Related work

Multimodal systems represent a remarkable deviation from the use of conventional systems, such as windows and icons, to a man / machine interaction, providing to the user more naturalness, flexibility and portability.
The first multimodal system was created in 1980 by Richard Bolt "Put-That-There" (Bolt, 1980a). The system is equipped with a microphone and a screen. It allows moving or changing the display of objects on the screen, using the voice accompanied by pointing on the screen.
Since Bolt's work , the academic world has provided models and designs offering multimodal interaction techniques (Jacob, Li et Wachs, 2012), (Nordahl et al., 2012), (Oviatt et al., 2000a) and (Meng et al., 2009).

The multimodal system is mainly composed of two indispensable modules: fission module (Poller et Tschernomas, 2006) and fusion module (Atrey et al., 2010).
When two or more modalities are invoked at the same time (e.g. speech and clicking a mouse button), the user invokes complementarities of these modalities. This provides a rationale for the invocation of multimodal fusion and the fusion engine which is responsible for determining the meaning of such complementarities.

The fission is the way to segment the data that will be presented to the user according to the available modalities and context (Nguyen, Odobez et Gatica-Perez, 2012). The fission module is a crucial component of multimodal systems. However, most research in multimodal systems focus more on the fusion than the fission. This point is supported by "There isn't much research done on fission of output modalities because most applications use few different output modalities therefore simple and direct output mechanisms are often used" (Costa et Duarte, 2011) and by "Multimodal fission is a research topic that is not often addressed in the scientific community." (Perroud et al., 2012).

This module is the main subject in our work. We focus specially on 1) the services connected to the output: multimodal fission and, 2) the creation of a multimodal interaction system.

The few examples presented in the literature about the fission process are very simple or question-response based process, that provide limited choices to the user. The user is therefore restricted to some particular command (predefined fission). For example, in the system presented in (Benoit et al., 2009), the fission process is very simple. It detects the state of the driver. It captures information from sensors installed in the car and then it tests if the values entered are in a specific range, and through this test the system will generate alerts like sonar alerts or vibration of the wheel.

Most of the architectures presented for multimodal systems are dedicated to very specific modalities (Palanque et Schyn, 2003), (Oviatt et Incaa Designs, 2007) and (Henry, Hudson et Newell, 1990). Also, most of the multimodal systems presented in the literature don't mention solutions to overcome the problem of uncertainty or ambiguity during the fission or fusion process. In (Benoit et al., 2009), they use the BN in the fusion process to model human fatigue and to predict fatigue based on the visual cues obtained when a driver is driving a car. The BN presented by the authors don't deal directly with fusion process but just to determine a parameter (level of fatigue) that affect the decision in the fusion process
In this work we introduce a probabilistic reasoning to integrate uncertain and ambiguous knowledge. We use BN that we adapted using the contextual information such as temperature, history, user status, etc. to resolve the problem of ambiguity and uncertainty in the fission system.

4.3 Architectural design

In this section, we present our proposed architectural design as a solution to the described problems of previous architectures.

A general overview of the architectural design is shown in Figure 4.1. The proposed system architecture is modular and distributed. All the components can be installed in different places in the network (robot, computer, server, etc.). The architecture is composed mainly of 7 modules/components. These components are as follows:

Command Extraction: this module has in input different XML files from applications. The output corresponds to the command that our system will process. The system interacts with many applications. It exchanges data (information about environment, command, user preference, etc.) using XML files. The purpose of this module is to extract the command from the XML files and send it to the fission module ;

Context Acquisition Module: the goal of this module is to select the adequate modality (ies) according to the user's preference, status of media and the status of the environment.

This module has in input:

- **The state of the environment obtained from the captors**. It selects the appropriate modalities depending on information collected from the captors such as the noise level. It is understood that the use of the audio modality is affected by this information. For example, if the noise level is high, the use of the audio modality is disabled. This is called environment context (Zaguia et al., 2010a) ;

- **The user location and the status of the user**. It defines the user's ability to use certain modalities. For example if the user is blind the display modality is disabled and if the user is in a library the audio modality is disabled. This is called user context (Zaguia et al., 2010a) ;

- **The state of the media**. This allows to determine the capacity and the type of system that we use. For instance if the battery of the cell phone is low then the system disables the audio modality and decreases the level of brightness. This is called system context (Zaguia et al., 2010a).

Modality(ies)-subtasks Association: the goal of this module is to select the adequate modality (ies) for a given subtask. Their inputs are the selected modality (ies) and the list of subtasks obtained from Context Acquisition Module and the Fission Module, respectively. The output contains the list of subtasks with the adequate modality (ies). This module sends queries to the ontology to find the matching pattern (Alexander, Ishikawa et Silverstein, 1977a) of subtask selection. In our case, pattern is defined by two parts: problem that presents the model of the command and solution that contains the suitable subtasks ;

Ontology: is the knowledge base that describes every detail in the environment (objects and events, etc.). It also contains all possible patterns (Zaguia et al., 2013a) of the subtask selection and the modalities selection ;

Fission module: is the crucial module in our architecture. It decomposes a command to elementary subtasks. This module has in input the command obtained from Command Extraction. The output contains:

- the list of subtasks which are then sent to the subtasks executions module,
- or a feedback message if the command is incorrect,
- or the command to the BN module in the uncertain case.

This module is detailed in section 4.4.
For more details about the previous modules, the reader can refer to (Zaguia et al., 2013a) and (Zaguia et al., 2013b).

Bayesian Network Module: in case of ambiguity or uncertainty, this module allows interaction with the ontology to make a decision using a probabilistic method. The input of this module is the ambiguous command. The output provides the decision to the feedback module.

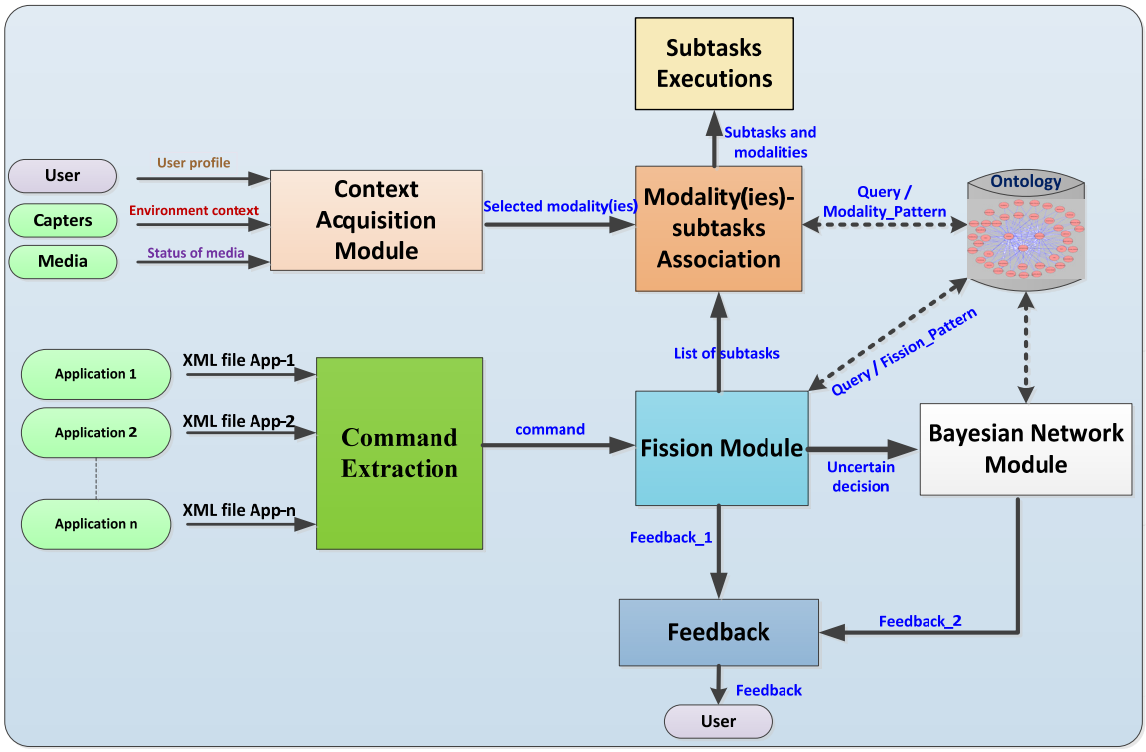


Figure 4.1 General approach of multimodal fission system

4.4 Fission algorithm

The fission process is the crucial part of the multimodal system. The fission is known as the way to subdivide commands to elementary subtasks and present them to the user according to the available modalities and context. Consequently, we can say that the goal of multimodal fission is to pass from a presentation independent of modalities to a coordinated and coherent multimodal presentation.

In general, the fission rule is simple (Figure 4.2): if a complex command (CC) is presented, then a list of subtasks with suitable modalities (and its parameters) is deduced.

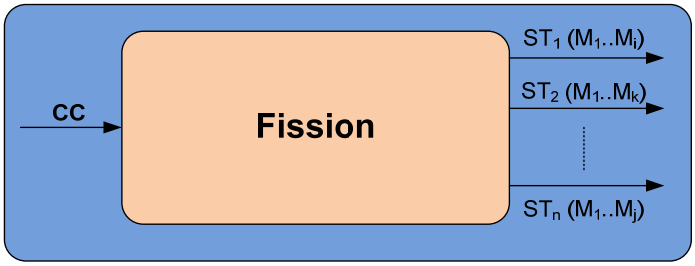


Figure 4.2 Fission process

Multimodal fission can be represented by the function:

$$f: F \rightarrow ExK$$
$$\forall cc \in F, \exists STi \in K \; and \; MOj \in E,$$
$$f(STi, MOj) = cc$$
$$\text{With: } i \in [1..n] \; et \; j \in [1..m]$$
$$f: CC = \sum_{i=1}^{n} ST_i \left(\left\{ \bigcup_{j=1}^{k} MO_j \right\}, \left\{ \bigcap_{j=1}^{l} MO_j \right\} \right) \quad (4.1)$$

With:

ST_i = sub-task i.

MO_j= output modality i.

cc = complex command.

l and k are different from m and n because it depends on the sub-tasks. For example, for some sub-tasks we will use only two terms even if we have three available modalities.

In equation (1), symbol ⋃ indicates that we can use either one or several modalities to present a sub-task. For example, if we present a text to the user, we use audio or display. The symbol ⋂ indicates that we use the available modalities together to present a sub-task.

Stages of fission process are described in Figure 4.3. The fission process starts when the system receives an XML file from a given application.

Then it extracts the command from the XML file and the system interacts with the ontology to get the meaning for every word (this part is detailed in (Zaguia et al., 2013b)). In the case of ambiguity or uncertainty decision the system uses BN to resolve the problem (this part is detailed in section 4.5) or else it gets the model of the command from the ontology. Then the system sends a query (containing the problem parameters) to the ontology to find the matching pattern. Pattern is defined with two parts: problem and solution. In our case the pattern problem presents the model of command and the pattern solution the list of subtasks.
If the pattern is found the system gets the list of subtasks and selects for every subtask the adequate modality (ies) (detailed in (Zaguia et al., 2013a)) or else the system will try to find a similar command in our ontology. If not found, the system creates elementary subtasks and asks a feedback from the user, or else the system finds the missing subtask using a BN (detailed in section 5.3) and asks a feedback from the user.

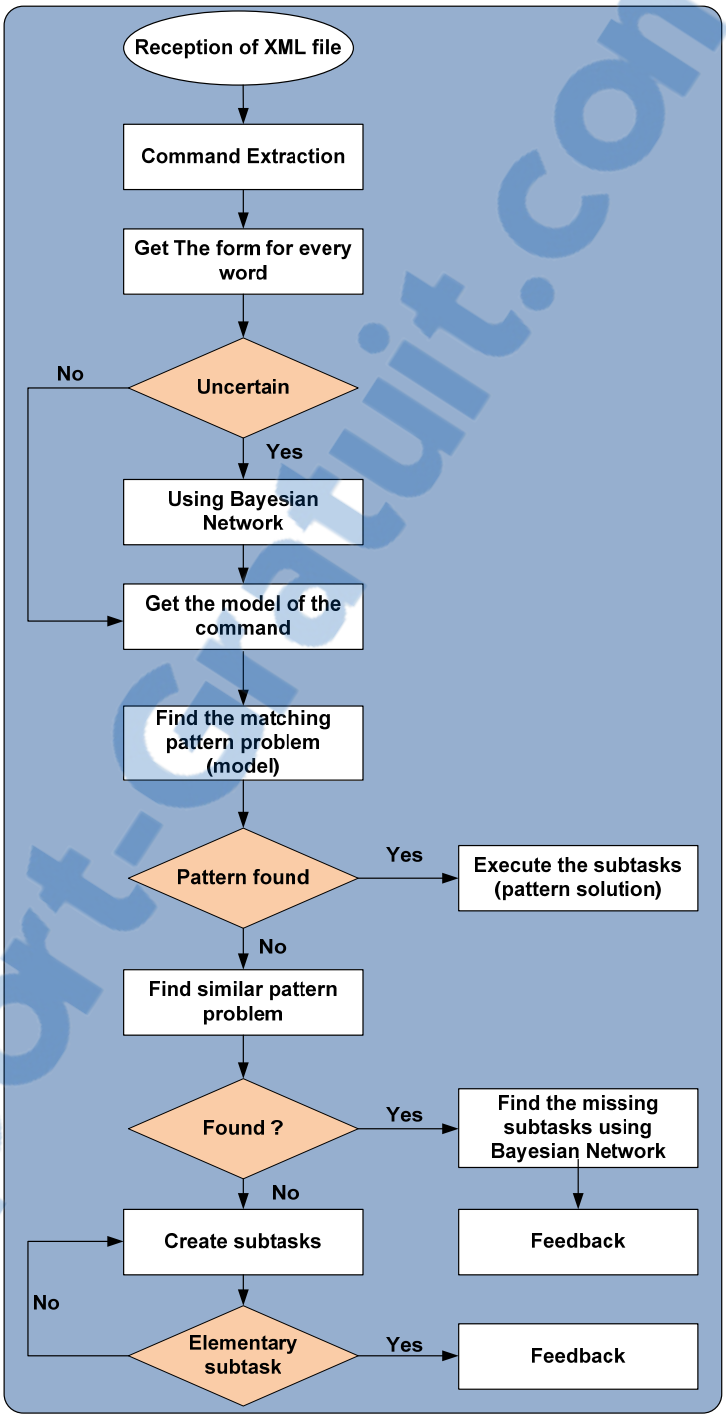


Figure 4.3 Fission algorithm

4.5 Bayesian network module

As we mentioned in the previous section, during the fission process, the system should make a decision between several equal choices. In uncertain or ambiguous cases, the BN is an effective solution to solve this problem.

In this section, we present the definition of the BN, then we present an example of how it is used in data mining and then how we adapt this method to use it within the context to overcome the problem of uncertainty in multimodal fission.

4.5.1 Definition

A BN provides a mechanism for graphical representation of uncertain knowledge and for inferring high-level activities from the observed data.

A BN is a graph in which nodes represent random variables and the links are influences between variables. The graph is acyclic: it contains no loop. (Friedman, Nachman et Peér, 1999).

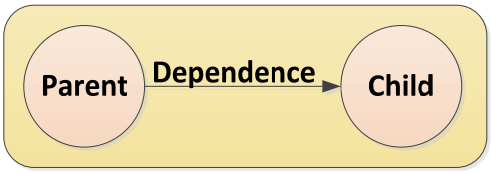


Figure 4.4 Relationship between two nodes in a BN

The arc produces a probability distribution where the parent has the a priori probability P (p), and the child conditional probability P (c |p) (Figure 4.4).

BN are used in many fields for diagnosis (medical and industrial), risk analysis, spam detection and data mining.

4.5.2 BN in Data Mining

For an effective research, we use a semantic Bayesian graph that combines a semantic inference and probability.

As we can see in Figure 4.5, a semantic Bayesian graph consists of several layers of nodes: queries, keywords, concepts and target resources.

At the first level of abstraction, keywords are used in user queries. Each keyword is connected to one or more concepts with a certain probability. Finally, each concept refers to a resource. The relationship between concepts and resources are semantic links.

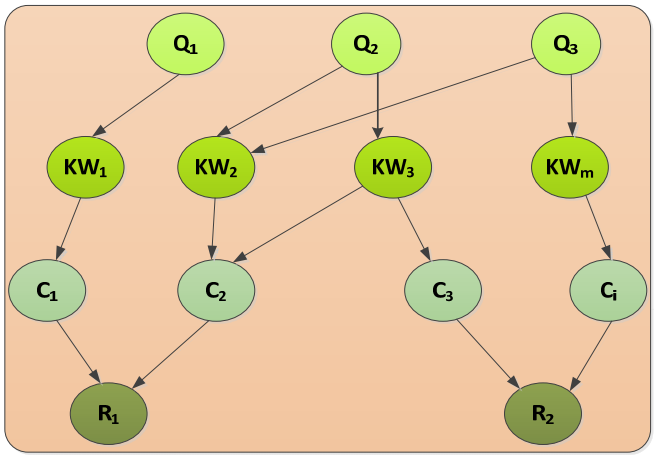


Figure 4.5 Semantic Bayesian network

The search of resources based on a semantic Bayesian graph is performed as follows:
First, the query is parsed and keywords are selected. Then the probability layer between keywords and concepts layer is calculated:

$$P(C|M) = \frac{\mathrm{P(M|C)P(C)}}{\mathrm{P(M)}} \begin{cases} C: concepts \\ M: keys\ words \end{cases} \tag{4.2}$$

$M = \mathrm{m}_1, \mathrm{m}_2 \ldots, \mathrm{m}_n \quad n: number\ of\ keys\ words$

Where:

- $P(C|M)$: a posteriori probability ;
- $\mathrm{P(M|C)}$: likelihood ;
- $\mathrm{P(M)}$: evidence ;
- $\mathrm{P(C)}$: a priori probability.

The idea in data mining is as follows: for each concept, its probability is calculated based on the conditional probabilities of the keywords of the user query P(M|c). Thereafter, it chooses the concept that has the highest probability. Since each concept's node points to a target resource, then the resource related to the chosen concept is returned as a response.

The conditional probability P (M|c) is assigned manually by consulting an expert in each field and P (c) generally equals 1 to give the same weight to each concept.
From this perspective, we present in the next paragraph the adaptation of this method to our case with the use of context.

4.5.3 BN applied in fission process

We can use the same concept to select data from the ontology in the case of uncertainly or ambiguity. We can represent the adaptation of the BN with context by the equation (4.3)

$$Con_i \underset{j=1}{\overset{m}{\rightarrow}} (C_j, P_j) \tag{4.3}$$

With $i \in [1..n]$

n presents the number of context.

m presents the number of ambiguous concepts.

Con = context, C= concept and P = probability.

Each context is connected to one or more concepts with a certain probability.
Here we explain our method using two different applications: GPS (by 1 example) and robot control (by 2 examples).

GPS example

Suppose the user asks the GPS: "I want to go to Montreal". As we can see in the ontology, the user may mean one of three possible concepts: C1 {City}, C2 {Restaurant} and C3 {Street} (Figure 4.6).

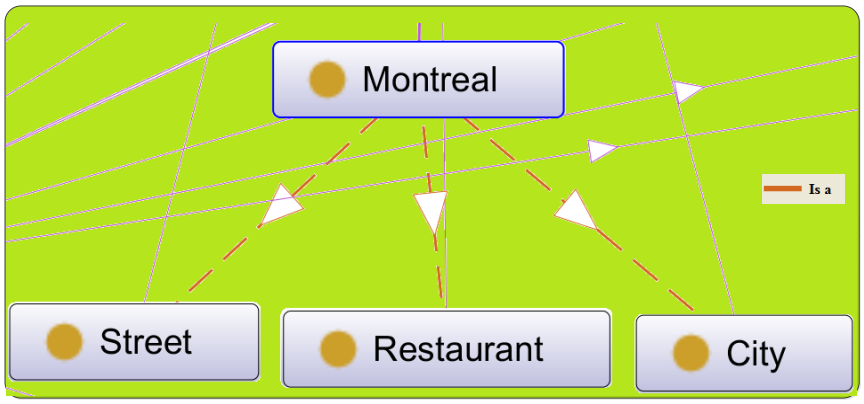


Figure 4.6 Montreal concept

In data mining, they use keywords of the command to select the appropriate resources. However, in this case the keywords {go, Montreal} are useless because all of them refer to destination. We can overcome this problem by taking into account the context (Miraoui, Tadj et ben Amar, 2008) instead of keywords such as in data mining. In this work we define context as: *user context*: user status, location, time and history. Context information is

collected from sensors or captors available in the user's environment. Suppose that the captured context is:

- the actual location is Paris ;
- the user is not hungry ;
- the actual time is 13:00.

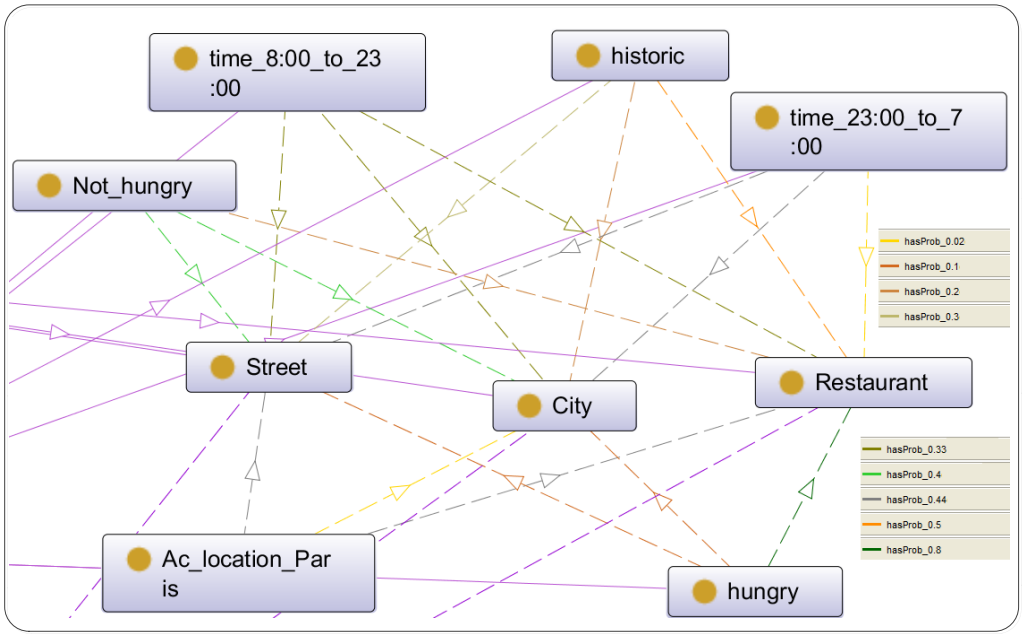


Figure 4.7 Bayesian Network for Montreal

Assume that the likelihood probabilities, usually estimated by an expert, are as follow (Figure 4.7):

$P(not\ Hungry|C1) = 0.4$

$P(not\ Hungry|C2) = 0.2$

$P(not\ Hungry|C3) = 0.4$

$P(Ac\_location\_Paris\ |C1) = 0.02$

$P(Ac\_location\_Paris|C2) = 0.44$

$P(Ac\_location\_Paris|C3) = 0.44$

$P(historic|C1) = 0.2$

$P(historic|C2) = 0.5$

$P(historic|C3) = 0.3$

$P(time\_8{:}00\_to\_23{:}00\ |C1) = 0.33$

$P(time\_8{:}00\_to\_23{:}00\ |C2) = 0.33$

$P(time\_8{:}00\_to\_23{:}00\ |C3) = 0.33$

Here we integrated many dynamic variables such as time, history, actual location, status of user, which can affect the decision. We can add more variables if we judge that it may to affect the result.

Then we calculate the a posteriori probabilities for every concept using equation (4.2).

$P(C1|Ac\_location\_Paris, historic, time\_8{:}00\_to\_23{:}00, not\ hungry) =$

$$P(C1)x\frac{P(Ac\_location\_Paris|C1)xP(historic|C1)}{P(Co)}\ \ x$$

$P(not\ hungry|C1)xP(time\_8{:}00\_to\_23{:}00|C1) =$

$$1x\frac{0.4x0.02x0.2x0.33}{P(Co)} = \frac{5x10^{-4}}{P(Co)}$$

$P(C2/Ac\_location\_Paris, historic, time\_8{:}00\_to\_23{:}00, not\ hungry) =$

$$P(C2)x\frac{P(Ac\_location\_Paris|C2)xP(historic|C2)}{P(Co)}\ \ x$$

$P(not\ hungry|C2)xP(time\_8{:}00\_to\_23{:}00|C2) =$

$$1x\frac{0.2x0.44*0.5*0.33}{P(Co)}=\frac{0.014}{P(Co)}$$

$$P(C3|Ac\_location\_Paris, historic, time\_8{:}00\_to\_23{:}00, not\ hungry)=$$

$$P(C3)x\frac{P(Ac\_location\_Paris|C3)xP(historic|C3)}{P(Co)}\ \ x$$

$$P(not\ hungry|C3)xP(time\_8{:}00\_to\_23{:}00|C3)=$$

$$1x\frac{0.4x0.44*0.3*0.33}{P(Co)}=\frac{0.017}{P(Co)}$$

As mentioned previously, we set $P(C1)=P(C2)=P(C3)=1$. We do so to give the same weight to every concept.

As we can see, P(C3|Co) has the highest value. This means that the concept C3 = {Street} is the most probable request the user meant. The system will seek feedback from the user "what is the number of the building in the Montreal Street?"

Robot control example 1

Suppose that the user gives an order to the robot "give me water". As we can see in the ontology (Figure 4.8), we have three possible cases: add water in glass or add water in casserole or add water in pail (in real life we can have many other choices, but we limit it to what we have in our ontology). We are in the case of uncertainly.

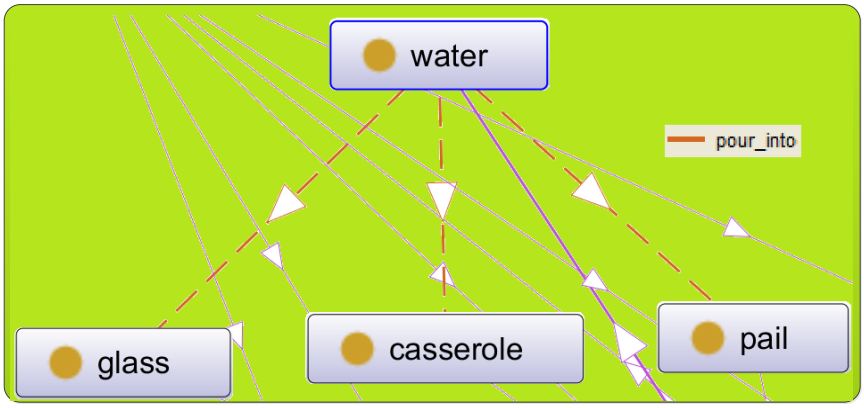


Figure 4.8 Example water ontology

We use the BN to resolve this problem. As shown in Figure 4.9, different contexts can influence the decision making. Each context is associated to all ambiguous concepts with a corresponding probability.

Using equation (1), we calculate the a posteriori probabilities for each concept {glass, casserole, pail}. Table 4.1 shows all likelihood probabilities.

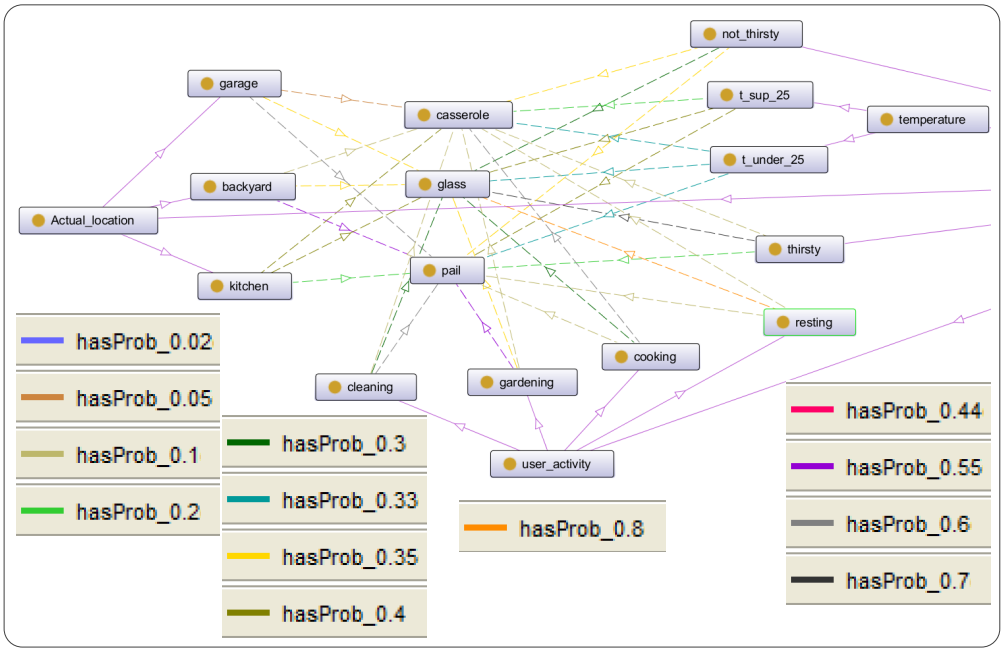


Figure 4.9 Likelihood probabilities

Table 4.1 Probabilities of likelihood of concepts {pail, casserole, glass}

	C1 {pail}	**C2{casserole}**	**C3 {glass}**
garage	0.6	0.05	0.35
kitchen	0.2	0.4	0.4
backyard	0.55	0.1	0.35
thirsty	0.2	0.1	0.7
not thirsty	0.35	0.35	0.3
t_under_25	0.33	0.33	0.33
t_sup_25	0.4	0.2	0.4
cleaning	0.6	0.1	0.3
gardening	0.55	0.1	0.35
cooking	0.1	0.6	0.3
resting	0.1	0.1	0.8

Suppose that captors, installed in the robots detect that the user is in the backyard, he is not thirsty, the temperature is higher than 25 degrees and he is gardening.

We calculate the a posteriori probabilities for every concept using equation (4.1) as in the previous example.

$$P(C1|backyard, Not\ thirsty,\ t\_sup\_25, gardening) =$$

$$1x\frac{0.55x0.35 * 0.4 * 0.55}{P(Co)} = \frac{0.042}{P(Co)}$$

$$P(C2|backyard, Not\ thirsty,\ t\_sup\_25, gardening) =$$

$$1x\frac{0.1x0.35 * 0.2 * 0.1}{P(Co)} = \frac{0.0007}{P(Co)}$$

$$P(C3|backyard, Not\ thirsty,\ t\_sup\_25, gardening) =$$

$$1x\frac{0.35x0.33 * 0.4 * 0.35}{P(Co)} = \frac{0.016}{P(Co)}$$

As we can see, $P(C1|backyard, Not\ thirsty,\ t\_sup\_25, gardening)$ has the highest probability. The system will choose the pail to bring the water.

In this stage, we presented two different ways to use the BN for two different applications. In the next example, we present the case of similar command and we show how the system finds the missing subtask as presented in section 4.4.

Robot control example 2

Assume that the user asks a robot "prepare a coffee with milk". Suppose that we already have in our ontology a pattern for "prepare a coffee" that contains all necessary subtasks.
Here we have similar commands (in red) and we have to find the missing subtasks ("prepare a coffee with milk").

In this case we select the adequate pattern for the command "prepare a coffee" and we find the missing subtask.

Milk presents a liquid in our ontology and we have already defined some possible elementary subtasks for liquid and objects as shown in Figure 4.10.

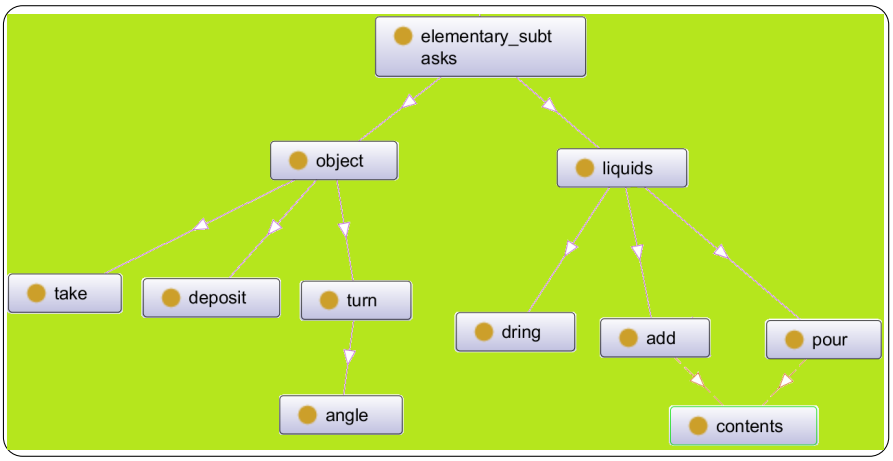


Figure 4.10 Some possible elementary subtasks for liquid and objects

We have three possible decisions: either *drink liquid* or *add liquid in content* or *pour liquid in content.* To make a decision, the system uses BN (Figure 4.11). We have three concepts: C1 {drink}, C2 {add} and C3 {pour}. We have two keywords related to these concepts *with* and *liquid.* In this case we use the keywords as used in data mining. As shown in Figure 4.11, the two keywords are related to the concepts with probabilities of likelihood:

$P(liquid|C1) = 0.5$
$P(liquid|C2) = 0.5$
$P(liquid|C3) = 0.5$

$P(with|C1) = 0.15$
$P(with|C2) = 0.7$
$P(with|C3) = 0.15$

$$P(C1|\, liquid, with) = P(C1)\ x\ P(liquid|C1)\ x\ P(with\ |C1)$$
$$= 1 \text{ x } 0.5 \text{ x} 0.15 = 0.075$$

$$P(C2|liquid, with) = P(C2)\ x\ P(liquid\ |C2)\ x\ P(with\ |C2)$$
$$= 1\ x\ 0.5\ x 0.7 = 0.35$$

$$P(C3|liquid, with) = P(C3)\ x\ P(liquid\ |C3)\ x\ P(with\ |C3)$$
$$= 1\ x\ 0.5\ x0.15 = 0.12$$

Note that:

$$P(C1|\ liquid, with) < P(C3|liquid, with) < P(C2|liquid, with)$$

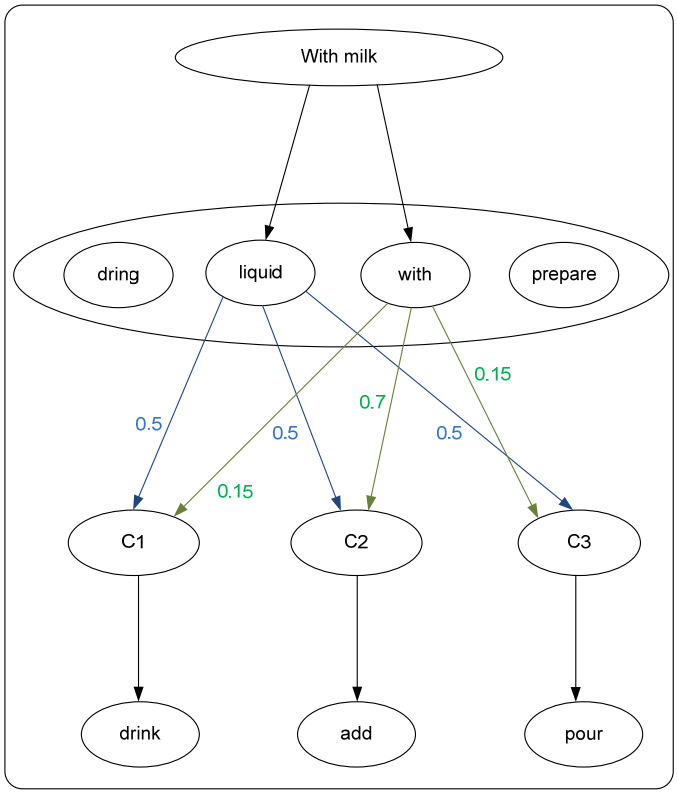


Figure 4.11 Bayesian network presenting

The concept *add* is therefore chosen as it has the lowest probability. The system adds the subtask "add milk" to the subtasks of the command "prepare a coffee" already existing in the ontology. The ontology is then updated with the new pattern.

4.6 Simulation

To understand the mechanism of fission using BN we demonstrate a simulation of a scenario.

In this scenario, we assume that in our multimodal system, a GPS indicates the directions to go to a specific location. For instance “I want to go to Washington”.

We first model and simulate the architecture defined in
Figure 4.1. The steps of each strategy are modeled using the colored Petri Net (Jensen, 1987) formalism and simulated using the CPN Tools (CPN-Tools, 2012).

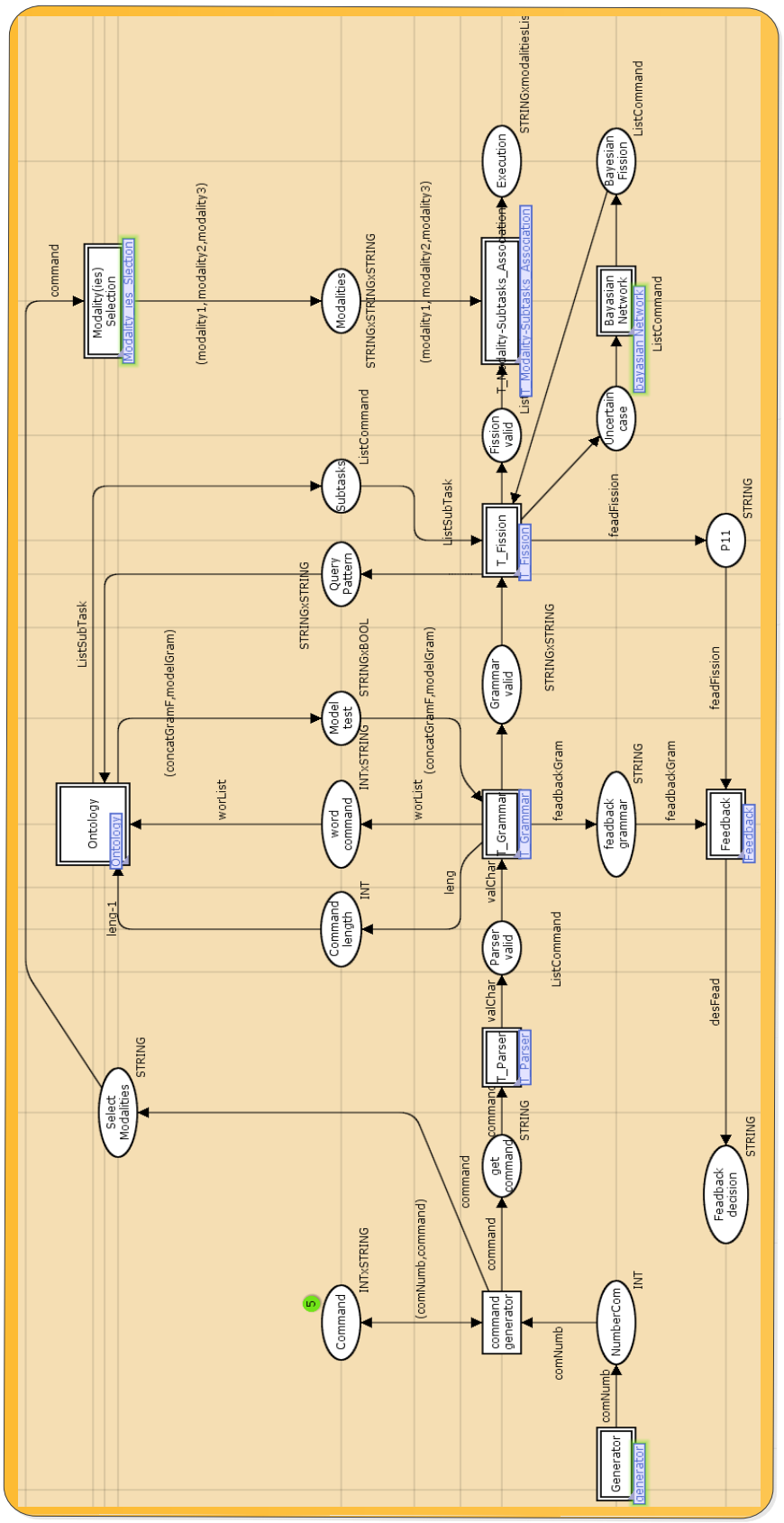


Figure 4.12 General view of our architecture

Figure 4.12 shows the general view of the architecture. It is composed mainly by 8 modules:

Generator*:* this module generates events as random numbers to select a command in the *place* "command".

T_parser: this module decomposes the command into words.

T_Fission: its role is to divide the command to elementary subtasks.

T_Grammar: permits to verify the meaning and the grammar of the command.

Modality(ies) Selection: this module allows to select the available modalities depending on the state of the environment.

T_Subtask-Modality_Association: it associates for each subtask the appropriate modality (ies).

Ontology: it is a container that stores the patterns as ontology concepts, the models and the vocabulary.

For more details about these modules, the reader can refer to (Zaguia et al., 2013a).

Bayesian Network: this is the main module presented in this paper. Its role is to handle **uncertainty.**

As we can see in Figure 4.13, the system receives the list of words of the command and the transition *list of the command*. The meaning for every word are retrieved from the ontology.

This is performed in the transitions *Decom List* and *meaning*. The role of *Decom List* is to send every word separately to the *meaning* transition. The *meaning* transition interacts with

the ontology to get the meaning. As shown in Figure 4.13, according to our example cited above, the place *meaning word, Washington* may have 7 possible solutions: {city, restaurant, basketball team, president, street, map}. In this case, the system won't be able to choose the right meaning.

Then in the transition *test of incertitude,* the system calculates the size of the meaning list. In our case, the size of *meaning* is greater than one. This is where the BN starts.

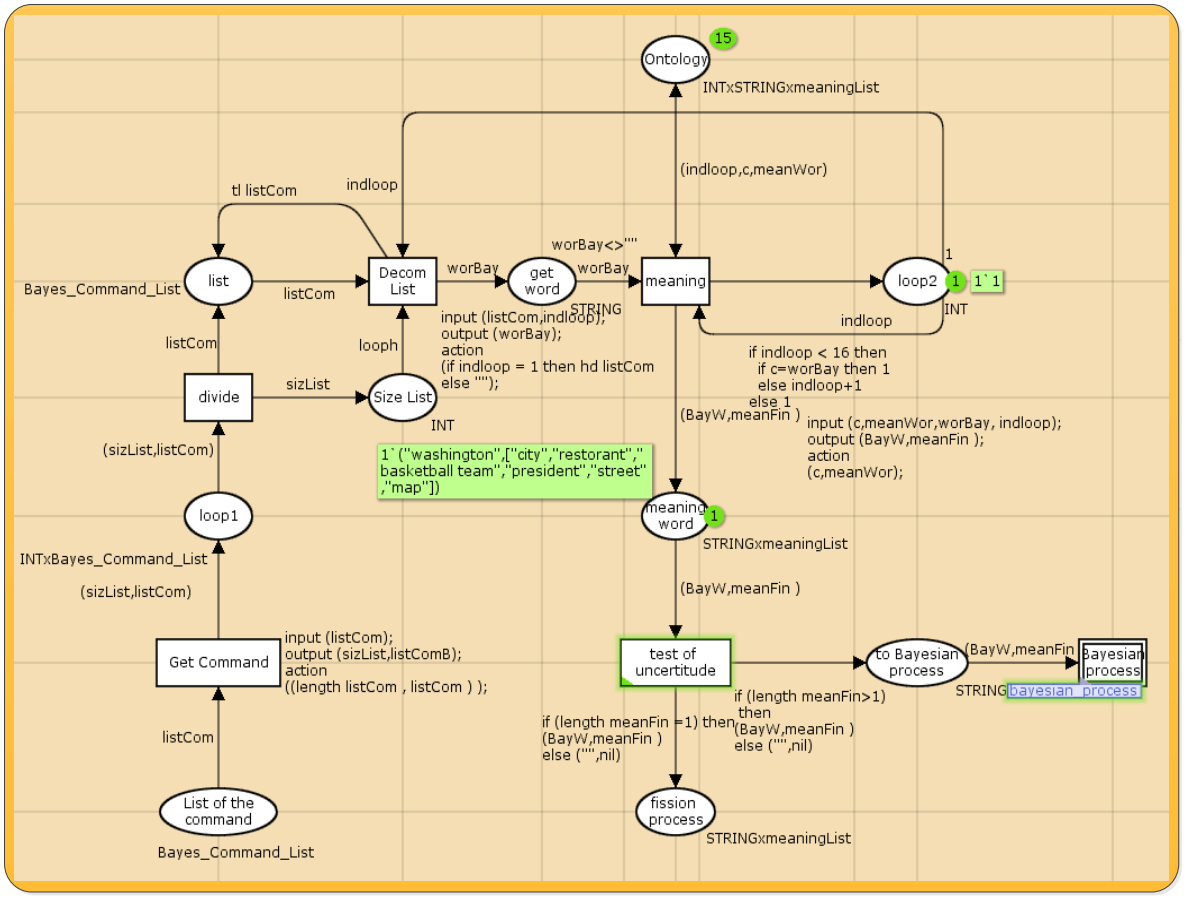


Figure 4.13 Process of uncertainty detection.

Figure **4**.15 shows the BN process. We have three main transitions: *Random context*, *probabilities of likelihood* and *calculate the a posteriori probabilities.*

The role of *Random context* is to simulate randomly information from captors. As shown in Figure 4.14, the transition "*collect information from captors"* collects information from places: *location captors*, *user state*, *Application* and *time.* In our example, the captured information is:

Table 4.2 The captured information

Application	GPS
Time	17:00
actual location	Washington
user state	not hungry

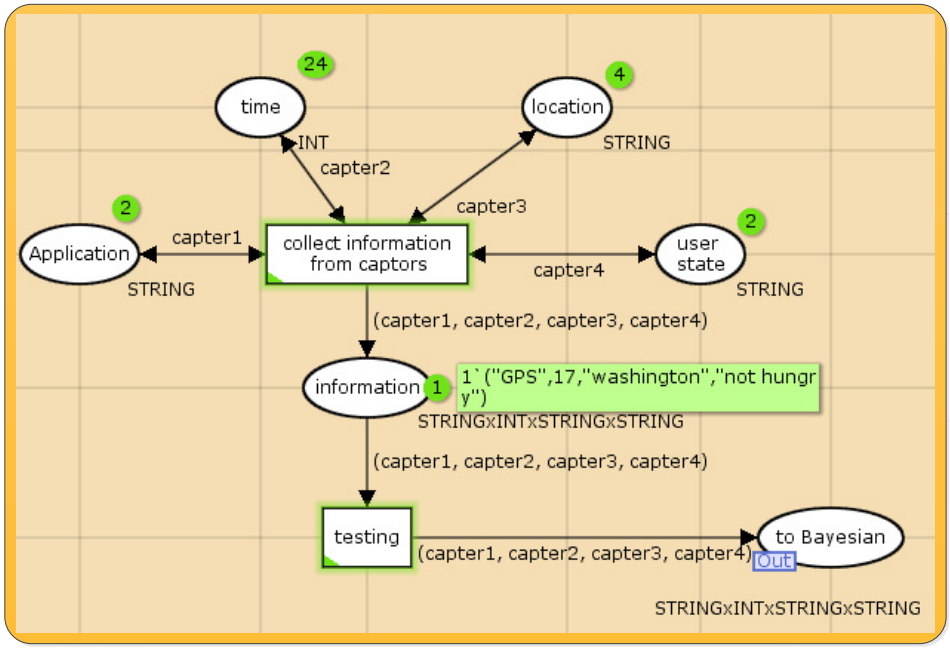


Figure 4.14 Collect of information from captors

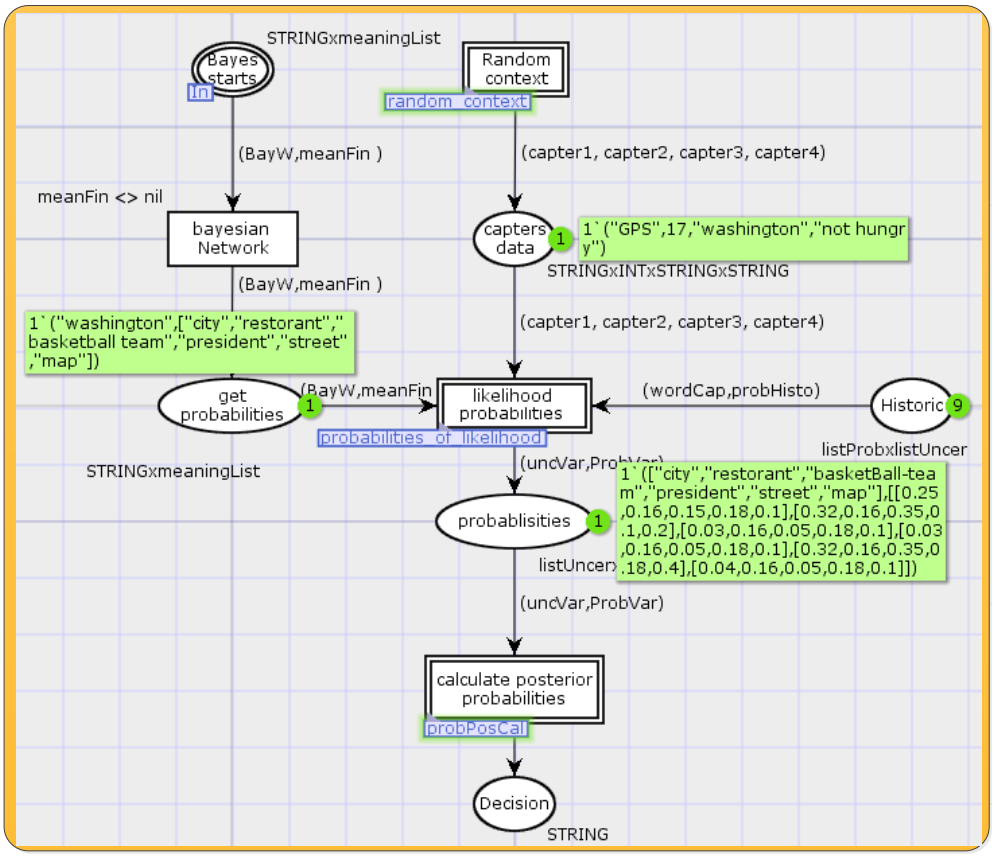


Figure 4.15 Bayesian process

The transition *likelihood probabilities* contains different likelihood probabilities. As shown in Figure 4.15, this transition has 2 inputs: from place *get probabilities* the word *Washington* and its meaning {city, restaurant, basketball-team, president, street, map} and from *captors data* place the context {GPS, 17, Washington, not hungry} and for the output the probabilities of likelihood, in our case the probabilities are shown in Table 4.3.

Table 4.3 Likelihood probabilities of concepts {city (Ci), restaurant (R), basketball-team (B-T), president (P), street (S), map (Ma)}

	GPS	**17**	**Washington**	**NoHungry**	**history**
Ci	0.25	0.16	0.15	0.18	0.1
R	0.32	0.16	0.35	0.1	0.2
BT	0.03	0.16	0.05	0.18	0.1
P	0.03	0.16	0.05	0.18	0.1
S	0.32	0.16	0.35	0.18	0.4
Ma	0.04	0.16	0.05	0.18	0.1

Then the transition calculates the a posteriori probabilities and takes a decision about the most probable concept.

In Figure 4.16, the system calculates the a posteriori probabilities for every concept using equation (1). The result obtained is (Figure 4.16):

(["city","restorant","basketBall-team", "president", "street", "map"],
[0.0001,0.0003,0.0,0.0,0.001,0.0]).

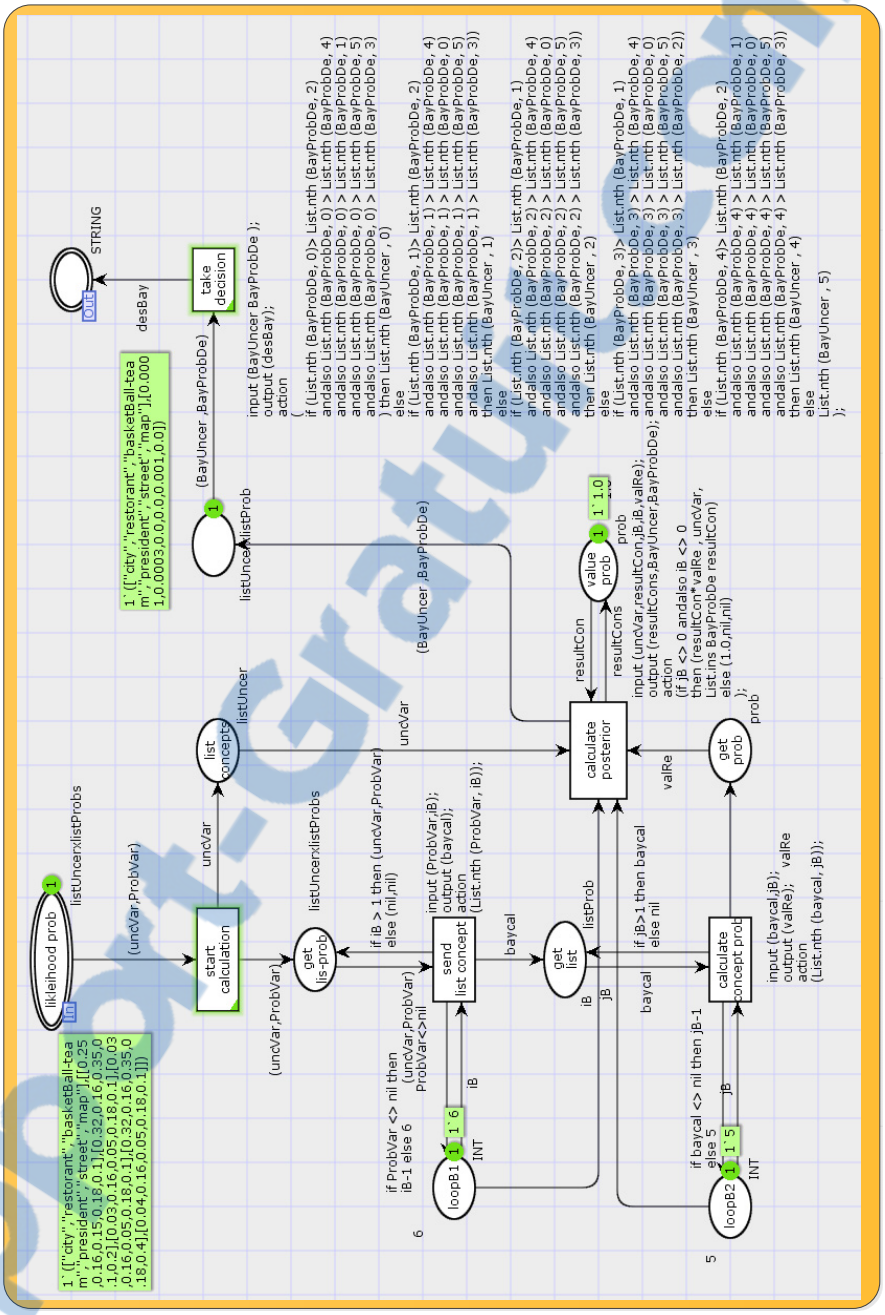


Figure 4.16 Probabilities calculation

4.7 Conclusion

The *fission* module allows segmenting the command generated by the machine to elementary subtasks and presents them to the available output modalities. However, most of the multimodal systems studied in the literature use predefined modalities. The available

architectures are targeted to specific applications. None of them seems to use any of the approaches belonging to probabilistic models when working with uncertainty, fuzzy or ambiguous real world data in multimodal systems.

In this paper we presented a new context-based method using Bayesian network to resolve the uncertainty problem during the fission process in a multimodal system. This method is useful, to overcome the problem of ambiguity or uncertainty. A concrete example was illustrated in order to show the effectiveness of the contribution, using CPN tool.

As a future work, we aim to develop a real application using methods presented in this paper.

CHAPITRE 5

PROTOTYPING USING A PATTERN TECHNIQUE AND A CONTEXT-BASED BAYESIAN NETWORK IN MULTIMODAL SYSTEMS

Atef Zaguia[1], Chakib Tadj[1], Amar Ramdane-Cherif[2]

[1]MMS Laboratory, Université du Québec, École de technologie supérieure

1100, rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3 Canada

[2]LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France

This article is submitted to the Journal of International Journal of Soft Computing and Engineering. Janvier 2014

Résumé

De nos jours, la technologie nous permet de produire des systèmes multimodaux bien développés et totalement contrôlés par l'homme. Ces systèmes sont équipés d'interfaces multimodales permettant une interaction plus naturelle et plus efficace entre l'homme et la machine. Les utilisateurs peuvent profiter de leurs modalités naturelles pour communiquer ou échanger des informations avec des applications. Dans ce travail, nous supposons que les différentes modalités de sortie (audio, écran, etc.) sont disponibles pour l'utilisateur. Dans cet article, nous présentons le prototype d'une architecture multimodale. Nous montrons en particulier comment la sélection des modalités et l'algorithme de fission sont mis en œuvre dans un tel système. Nous avons utilisé la technique de pattern pour 1) subdiviser une commande complexe en sous-tâches élémentaires et 2) sélectionner les modalités adéquates pour chaque sous-tâche. Nous intégrons une méthode basée sur le contexte en utilisant un réseau bayésien pour surmonter les situations ambiguës et incertaines.

Mots clés : multimodalité, ontologie, réseau bayésien, pattern, interface utilisateur, fission multimodale.

Abstract

Today, technology allows us to produce extensive multimodal systems which are totally under human control. These systems are equipped with multimodal interfaces, which enable more natural and more efficient interaction between man and machine. End users can take advantage of natural modalities (e.g. audio, eye gaze, speech, gestures, etc.) to communicate or exchange information with applications. In this work, we assume that a number of these modalities are available to the user. In this paper, we present a prototype of a multimodal architecture, and show how modality selection and fission algorithms are implemented in such a system. We use a pattern technique to divide a complex command into elementary subtasks and select suitable modalities for each of them. We integrate a context-based method using a Bayesian network to resolve ambiguous or uncertain situations.

Keywords: Multimodality, Ontology, Bayesian Network, Pattern, User Interface, Multimodal Fission.

5.1 Introduction

Computer systems are born of scientific needs, but they owe their success to general public use. This has motivated researchers to develop systems that meet the needs of users and to target use of these systems on a large scale. Current technological advances are leading to the design of ever more powerful machines which are increasingly easy to use. These machines should be capable of interacting with users in a harmonious way, but this is only possible if they are able to understand human communication in many of its natural modalities, such as speech, gestures, eye gaze, and facial expressions. Inspired by these communication modalities, multimodal systems are developed to combine a number of them, depending on the task at hand, on user preferences, and on the user's intentions.

These systems represent a remarkable deviation from conventional systems with their standard human-machine interface modalities, such as windows icons, for example, as they provide the user with more natural means of interaction, which are both flexible and portable. A multimodal system has two crucial components: the fusion process and the fission process. In fusion (Atrey and al., 2010) two distinct data modalities are combined; for example, data modalities, such as a mouse and speech (Djenidi and al., 2004). In fission, by contrast, a complex command is divided into elementary subtasks, which are presented as elementary output modalities (Zaguia and al., 2012).

In our work here, we focus on: 1) the services connected to the output, that is, multimodal fission; and 2) the creation of a multimodal interaction system. The objective is to develop a flexible multimodal system, capable of manipulating more than two modalities that can interact with more than one application. The approach consists of designing modules that detect the modalities involved, take into account the parameters of each modality, and perform the fission of the modalities in order to obtain the corresponding action to be undertaken. The rest of this paper is organized as follows. Section II discusses work that is related to ours. Section III describes our research issue. Section IV presents the various components of the proposed architecture. Section V discusses modality selection and the fission algorithm. Section VI presents the implementation of the prototype. The paper is concluded in section VII

5.2 Related work

In the context of human-machine and human-human communication, modality refers to the path or channel by which human and machine interact. Multimodality improves the recognition and comprehension of the commands emanating from the environment (user, robot, etc.) by the machines. A system that combines many modalities dynamically in both input and output is called a multimodal system. The first multimodal system was created in 1980 by Richard Bolt, “Put-That-There” (Bolt, 1980). This system, equipped with a

microphone and a touchscreen, made it possible to move or change the display of objects on the screen, and accommodate voice commands accompanied by pointing on a touchscreen. Since Bolt published his work, academia has provided prototypes and systems that offer a variety of multimodal interaction techniques. These systems are a particularly effective solution for users who can't use a keyboard or a mouse, visually impaired users, users equipped with mobile devices, handicapped users, etc.

Most current multimodal systems address a very specific technical problem, such as media synchronization (Little and al., 1991), or they are dedicated to very specific modalities (Oviatt and al., 2000), (Raisamo and al., 2006), (Debevc and al., 2009) and (Lai and al., 2007). The system presented in (Djenidi and al., 2004) is a multimodal system designed for blind users, which allows them to read through Braille mathematical formulas. This system combines a keyboard and voice for input and Braille and audio for the output.

In (Caschera and al., 2009), the authors present an efficient multimodal system (RISCOM) which can be used in the case of a disaster. The system sends information on a mobile device in the form of maps to indicate the location of safe havens in a natural disaster. It uses display, audio, and text message modalities, is easy to use, and provides instant access to the information. By integrating multiple modalities, this system is a very effective way to deliver emergency services in critical situations, which can help save lives.

The prototype presented in (Karpov and al., 2010) is equipped with a speaker, a video-camera, a microphone, a touchscreen, a graphical user interface, and a talking head. It is a multimodal kiosk with a user interface which accommodates touch, natural speech input, and head and hand gestures, and can also be used by those who are physically challenged (Karpov and al., 2010).

PIXELTONE (Laput and al., 2013) is a multimodal photo editing system. The user speaks to edit images, instead of hunting through menus.

However, most research in multimodal systems is focusing more on the fusion process (Zaguia and al., 2010b) than the fission process (Costa and Duarte, 2011), in spite of the fact that the fission module is a critical component of multimodal systems. As these authors put it, “There isn’t much research done on the fission of output modalities, because most applications use few different output modalities, therefore simple and direct output mechanisms are often used” (Costa and Duarte, 2011). Also supporting our viewpoint are (Perroud and al., 2012): “Multimodal fission is a research topic that is not often addressed in the scientific community.”. In the few cases where the fission process is presented, the systems are very simple (Benoit and al., 2009) and the user is limited to some predefined instructions.

In this paper, we propose a new methodological solution in which an architecture is modeled that facilitates the work of the fission process. We do this by defining an ontology that contains different applicable scenarios and describes the environment in which the multimodal system exists. We then implement this architecture in a real application.

5.3 Challenges and the proposed solution

Our objective is to develop an expert system capable of providing services to different multimodal applications. The main goal is to create a multimodal fission system capable of understanding a complex command from the environment, particularly from the user. In order to achieve this, a complete semantic modeling of the environment is required. The task of this system is to divide a complex command into elementary subtasks and present them to the output modalities.

To achieve this goal, we list the main challenges that need to be addressed in developing our system, along with our proposed solutions:

1. What are the modules required to design the architecture of a multimodal fission system? Here, we will specify, define, and develop all the necessary components of the system ;

2. How will we represent the multimodal information? We will model the environment semantically, and create a context-sensitive architecture that is able to: 1) manage multiple distributed modules; and 2) automatically adapt to the dynamic changes of the interaction context (user, environment, system) ;

3. How will we perform the fission process? We will introduce an algorithm that describes the fission mechanism, including the rules of fission and the rules for selecting the output modalities ;

4. How will the system manage uncertain or ambiguous data in the fission process? We will introduce a new, context-based method using a Bayesian network (BN) to resolve the uncertainty problem during the fission process in a multimodal system ;

5. What is the optimal representation of the environment in our architecture? The optimal representation is a solution based on an ontology, which we will adopt. The modalities, scenarios, and objects, along with their characteristics, are stored in the ontology that describes the relationship between them.

We present these challenges summary in this paper. For more details, please refer to (Zaguia and al., 2013a), (Zaguia and al., 2013b), and (Zaguia and al., 2013c).

5.4 Components of multimodal fission system

The objective of multimodal fission is to move from an independent presentation of modalities to a coordinated and coherent multimodal presentation. A general schema of a fission process is presented in Figure 5.1. The process consists of three main modules:

Modalities Selection, *Fission*, and *Subtasks-Modalities Association*. These modules are explained briefly in the next sections.

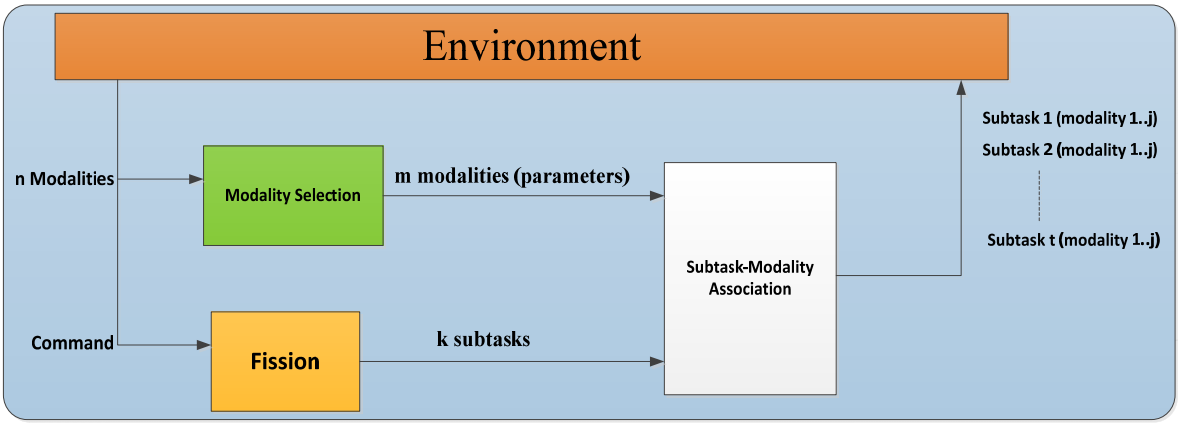


Figure 5.1 General view of the fission process

5.4.1 Modalities selection

This module plays an important role, as it is here that the appropriate modalities are selected according to the context. The context is defined by three essential components (Zaguia and al., 2010a):

User context: This component gives the user profile and its location. It makes it possible to determine the capability of the user to use certain modalities. For instance, if the user is blind, the display modality will be disabled.

Environmental context*:* This component obtains information from sensors installed in the user/machine environment. It detects changes in that environment and adjusts the selection of modalities based on these changes. For example, if the system detects that the environment is becoming too noisy, the audio modality is disabled.

System context: This component detects the computing device that the user is currently using, as well as the important parameters of the computing resource, such as the currently

available bandwidth, the network to which the computer is connected, the computer's available memory, the specifications of the battery, and the type of processor and its activities.

5.4.2 Fission

This module takes as input the complex command and produces as output the elementary subtasks. It is the crucial component in our architecture, with the role of dividing a complex command into elementary subtasks.

The fission process is based on the use of the pattern fission technique (Zaguia and al., 2013b). These patterns are generally defined as having two parts: problem and solution (Figure 5.2).

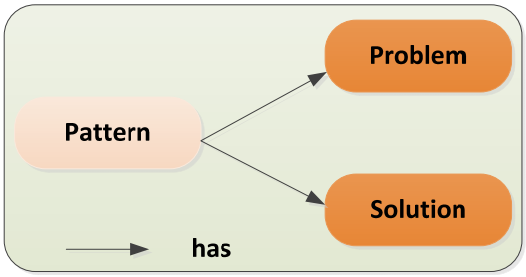


Figure 5.2 Pattern definition

The patterns are stored in an ontology. A simple example of a pattern is shown in Figure 5.3.

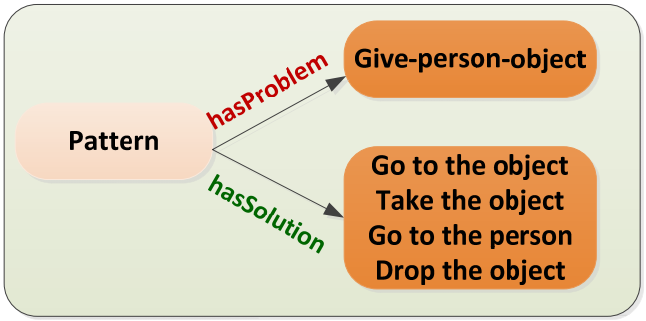


Figure 5.3 Example of a pattern

The system sends a query with the problem parameters to find the matching pattern fission in the ontology.

5.4.3 Subtasks-Modalities association

This module takes as input the elementary subtasks and the available modalities, and as output a set of subtasks associated with the appropriate modality or modalities. The goal with this module is to associate each subtask generated by the fission module with the appropriate and available modality or modalities. We also use patterns in this part as predefined models that describe the selected modality or modalities. In our work, a modality pattern is defined by: a) a *problem* composed of the components *application*, *parameter*, *priority*, *combination*, *scenario*, and *service*; and b) a *solution* composed of the selected modality or modalities. For more details regarding scenario selection and modality selection, see (Zaguia and al., 2013a).

5.4.4 Bayesian network

During the fission process, the system might face ambiguity or uncertainty. To overcome this problem, we use a context-based Bayesian network (BN). "A BN provides a mechanism for graphical representation of uncertain knowledge and for inferring high-level activities from the observed data. Specifically, a BN consists of nodes and arcs connected together forming a directed acyclic graph. Each node can be viewed as a domain variable that can take either a set of discrete values or a continuous value. An arc represents a probabilistic dependency between the parent node and the child node" (Ji, Zhu, and Lan, 2004).

We represent our adaptation of the BN with context information (time, user status, temperature, location, etc.) with the following equation:

$$Con_i \underset{j=1}{\overset{m}{\rightarrow}} (C_j, P_j),\ i = 1, \cdots, n \tag{5.1}$$

where n and m represent the number of contexts and the number of ambiguous concepts respectively, Con = context, C= concept, and P = probability. The arrow represents the relation between context and concept. Each context is connected to one or more concepts with a corresponding probability. We choose the most probable concept by calculating the probabilities of each concept using the following equation:

$$\mathrm{P(C|Con)} = \frac{\mathrm{P(Con|C)P(C)}}{\mathrm{P(Con)}} \tag{5.2}$$

$Con = \mathrm{con}_1, \mathrm{con}_2 \ldots, \mathrm{con}_n$
Where:

- $P(C|Con)$: a posteriori probability;
- $\mathrm{P(Con|C)}$: likelihood;
- $\mathrm{P(Con)}$: evidence;
- $\mathrm{P(C)}$: a priori probability.

For more details on the context-based method using the BN, see (Zaguia and al., 2013c).

5.4.5 Ontology

We use an ontology to model the environment. As shown in Figure 5.4, the environment is composed of the following classes (Zaguia and et al., 2013a):

The **Modality Class** represents the possible modalities present in the environment. It contains the subclasses *vocal*, *visual*, *gestural*, *tactile*, and *manual*.

The **Context Class** represents the interaction context containing the user context, environmental context, system context, and location.

The **Event Class** contains the four subclasses that can form a command, and their possible combinations:

– **Action subclass**: the verbs that the command can contain ;

– **Location subclass***:* the locations that we can find in a command ;

– **Object subclass**: the various objects that we can use ;

– **Person subclass**: the relations that exist between individuals.

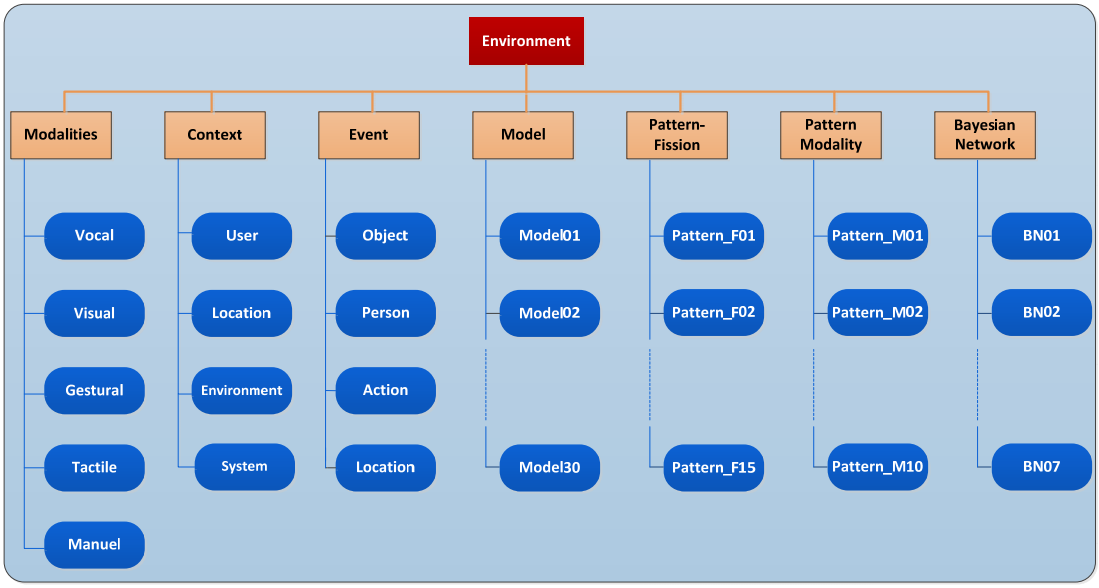


Figure 5.4. The classes of the ontology

The **Model Class** contains 30 subclasses (Model01 to model30), and allows us to validate the meaning and the grammar of a command. We have defined several models, each of which includes two or more subclasses of the Event class in a predefined order. For instance, the model of the command, “Put the ball on the table,” is Action→Object→Location.

The **Pattern Fission Class** contains 15 subclasses (Pattern_F01 to Pattern_F15), and describes various scenarios that are saved as patterns. They are mainly composed of two parts: problem and solution, as described briefly in section 4.2 and in detail in (Zaguia et al., 2013b).

The **Pattern Modality Class** contains 10 subclasses: Pattern_M01 to Pattern_M10, and allows the selection of the appropriate modalities for a given subtask.

The **Bayesian Network Class** contains 7 subclasses (BN01 to BN07), and defines the possible BNs in the case of ambiguity or an uncertain situation (Zaguia et al., 2013c). These subclasses are modeled in the ontology.

5.5 Modalities selection and fission algorithms

In this section, we present the algorithm for every module presented in Figure 5.1. Figure 5.5 shows modality selection algorithm. Once the environment has been modeled and a set of contexts defined, these contexts become the parameters that affect modality selection. There are four types of context: *environment*, *user*, *location*, and *system*. When an event is detected, the system receives information from sensors and looks for correspondence between the data received from the environment and those of the ontology. If contexts are verified, then the modality is enabled; otherwise it is disabled.

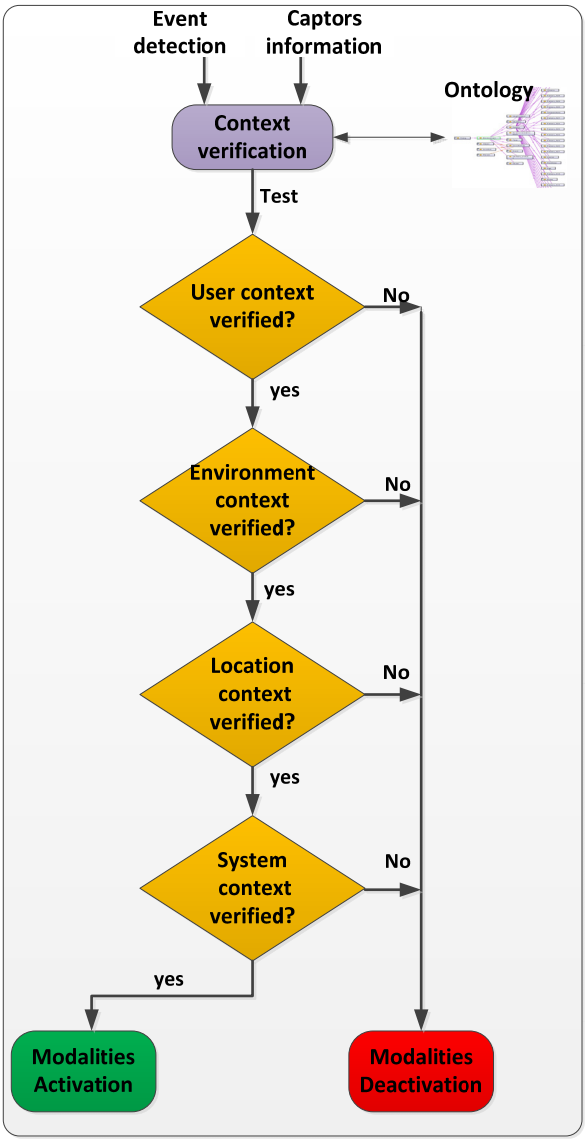


Figure 5.5. Modalities selection.

Figure 5.6 describes the steps involved in the fission process. In this diagram, a number of commands *n* serve as an input to the system. The steps undertaken are as follow:

Step-1: the system extracts every word from the command.

Step-2: for every word, the vocabulary stored in the Vocab ontology is checked.

Step-3: $vocab_i$ is extracted from each $word_i$. The extracted words are then concatenated in the same order as in the original command. In this way, the model of the command is obtained.

Step-4: a query is sent to the Grammar Model ontology to look for the model.

Step-5: if the model is found, we proceed with step 7 otherwise we proceed with step 6.

Step-6: the command is not valid and a feedback is sent to the user.

Step-7: a query is sent to find a matching pattern fission from predefined patterns stored in the Pattern Fission ontology. The system compares the query with all the pattern fission problems stored (in the ontology) until it finds a match. Then the pattern fission solution is returned.

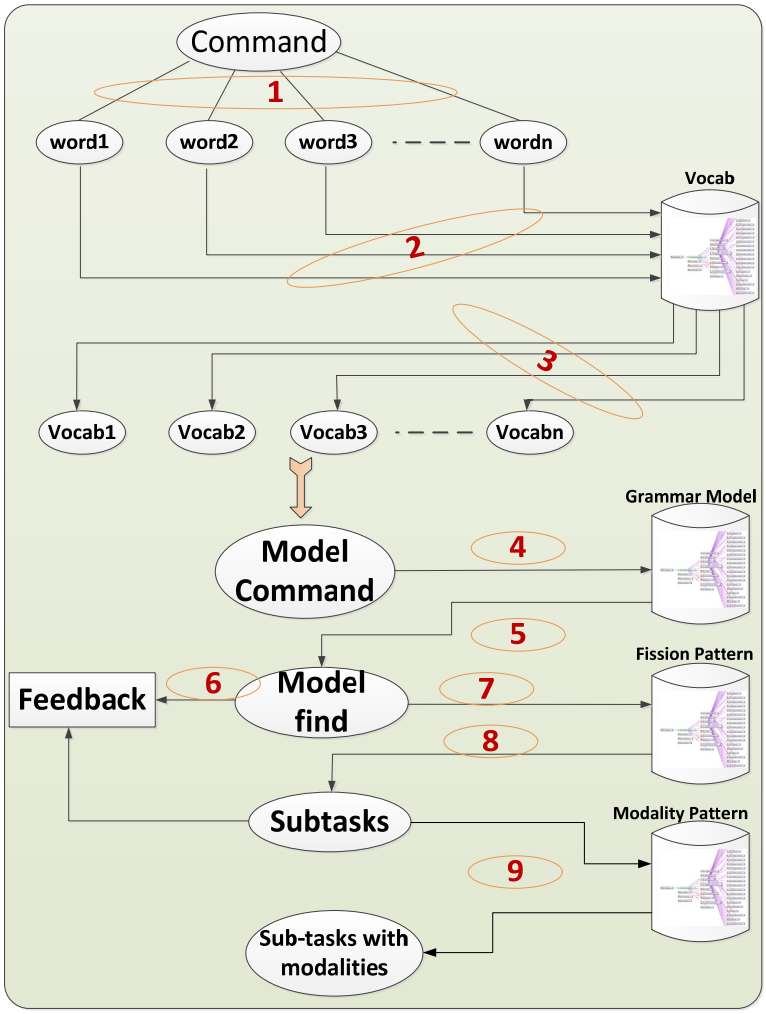


Figure 5.6 Stages of fission process

Step-8: if no matching pattern is found, a feedback is sent to the user.

Step-9: every subtask is associated with the appropriate and available modality or modalities. This is done by sending a query to find the matching pattern modality.

5.6 Prototype's implementation

The prototype focuses on multimodal fission, the main idea behind this work. We are interested in the selection of modalities and the fission process, so that the system will be

able to understand the commands sent by the user and adapt to any change in the environment.

We used the JAVA programming language to implement the four modules in our system (Figure 5.7), and we used four computers connected to the Internet to implement it. Table 5.1 describes each computer and the module it contains.

Table 5.1 Characteristics of the computers used

Computer	**specifications**	**Module**
A	Operating System: XP Internet Connection: WIFI IP Address: 192.168.2.100 Processor: Xeon (TM) CPU 3.40 GHz	GPS Application
B	Operating System: Win7 Internet Connection: file IP Address: 192.168.2.42 Processor: Intel® Core (TM) 2 CPU 1.67 GHz	Robot Application
C	Operating System: Win7 Internet Connection: file IP Address: 192.168.2.42 Processor: Intel® Core (TM) 2 CPU 1.67 GHz	Fission
D	Operating System: XP Internet Connection: file IP Address: 192.168.2.68 Processor: Intel®Core (TM) 2CPU 1.67 GHz	Ontology

As shown in Figure 5.7, stage (1) consists of sending xml files from user 1 (computer A) or user 2 (computer B) to the computer that contains the fission module, in our case computer D. This module receives files – stage (2) – and extracts the complex command. In stages (3), (4), (5), and (6), the fission module interacts with the ontology (computer C) to divide the complex command into elementary subtasks associated with the appropriate modalities. Once the fission process is completed, the system sends the result to the appropriate application – stage (7) – and the result is presented to the user in stage (8).

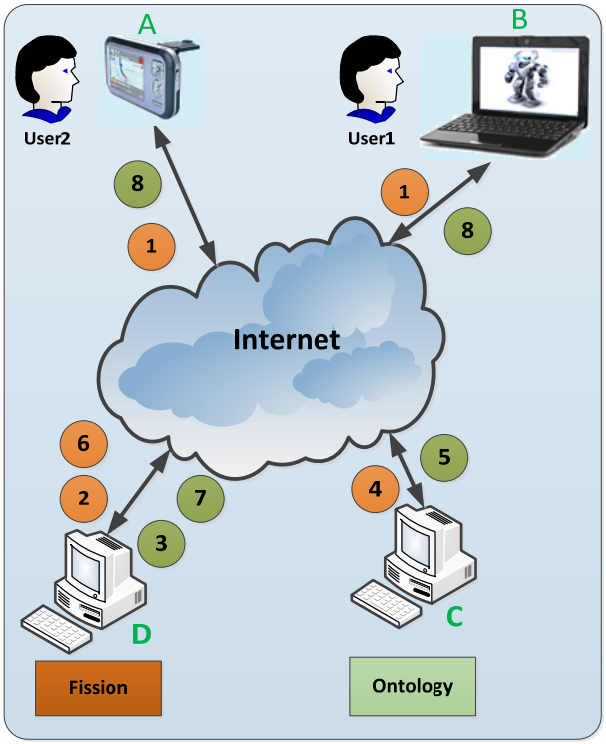


Figure 5.7 System's implementation

The files are transferred from one module to another using the Socket class implemented in Java (Java, 2013). Two graphic interfaces are created for the implementation of the prototype: robot control interface (Figure 5.8, corresponding to B in Figure 5.7) and GPS interface (Figure 5.9, corresponding to A in Figure 5.7). The robot interface is used to validate our fission module and the GPS interface is used to validate the BN with the context in the case of ambiguity or uncertainty.

5.6.1 Robot control interface

This interface is made up of five components: modalities, context, command, elementary subtasks, and actual execution.

The component context affects the selection of the modalities. Normally, contextual information is captured from sensors installed in the environment. In our simulations, the system randomly generates different values (Figure 5.8). Suppose we have the following information:

- the level of the noise is high ;
- the level of brightness is good ;
- the temperature is 19 °C ;
- the bandwidth is good ;
- the user is not physically challenged.

As the level of noise is high, the context will affect only the audio modalities. In the *command* component, the command chosen by the user to be executed is “Move the circle to (205, 175).” When the user clicks on “Start”, an XML file containing the complex command is created and sent to the Fission module, which is located in computer D. The system follows the steps of the algorithm presented in section 5. The result of the command is presented in the *elementary subtasks* component: {Move to the circle at position (152, 58), Take the circle, Move the circle to position (205, 175), and Drop the circle}. Finally, the *actual execution* component represents the subtask that the system is in the process of executing and the appropriate modalities associated with the subtask. In our example, the results are:

- Subtask: move the circle (152, 58) ;

- Modalities: {Mobility mechanism, Screen, printer}.

Figure 5.8 Robot control interface

5.6.2 GPS interface

This interface is implemented to illustrate the use of BN with the context. As shown in Figure 5.9 (a), the main interface is composed of two components: *Context* and *Command*. For the *Context* component, we used *time*, *actual location*, *user status*, and *temperature*. The values of these contexts are chosen randomly by the system (Table 5.2).

Let us say that the command entered by the user is "I want to go to Montreal." When the user clicks on "Start", the fission process begins. In this case, "Montreal" has many meanings in the ontology. This puts us in an uncertain situation.

Table 5.2 Context's parameters

Context	**Value**
Time	12:45
Actual location	Montreal
User status	Hungry
temperature	22

Figure 5.9 (b) shows the detection of the uncertain case {City, Restaurant, Street}. When we click on *Decision Process*, the system uses equation (2) and the algorithm presented in (Zaguia, Tadj et Ramdane-Cherif, 2013) to choose the best situation according to the context. The decision taken is shown in Figure 5.9 (c): Restaurant is the most probable request. The more contextual parameters we add, the more accurate the result will be.

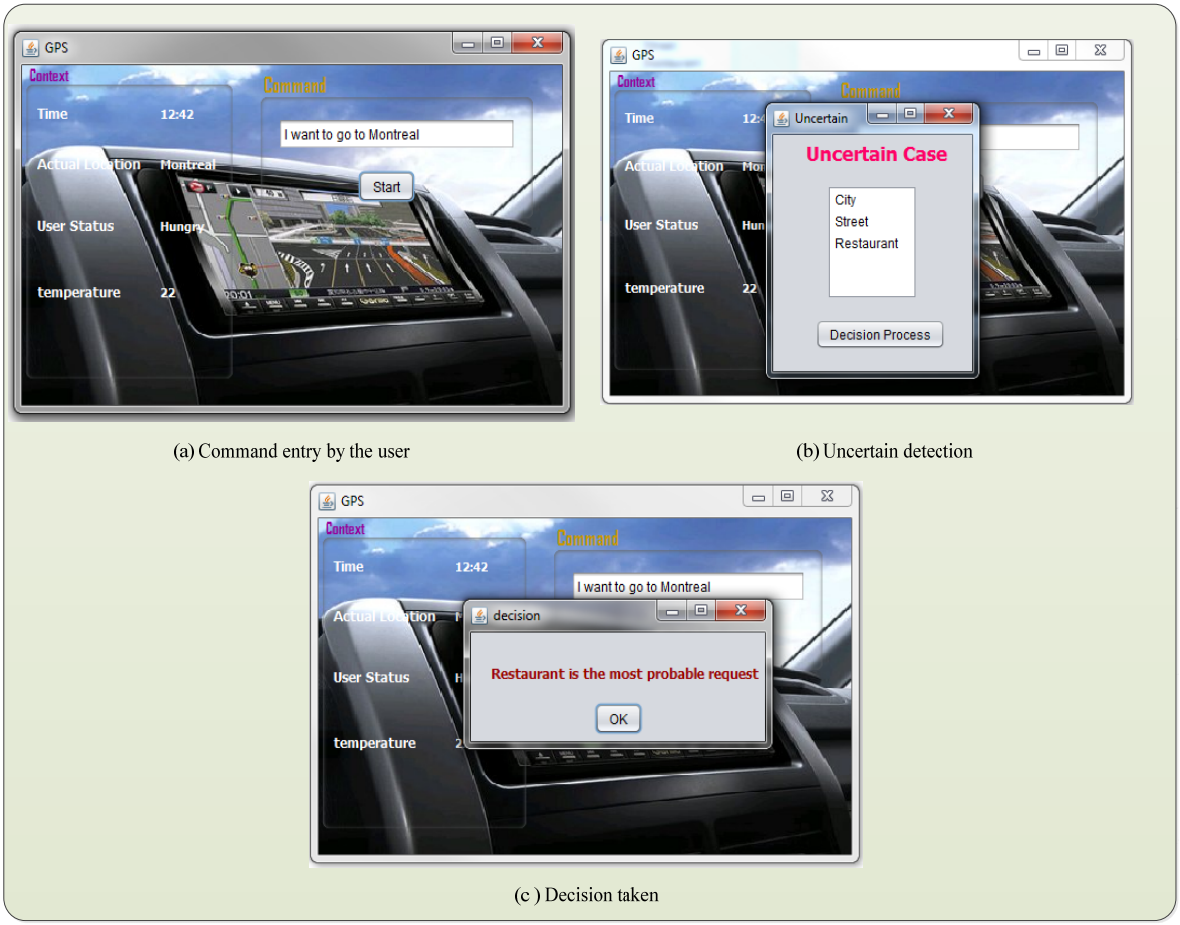


Figure 5.9 GPS interfaces

5.7 Conclusion

In In this paper, we presented a very useful architecture for a fission system which allows multimodal interaction. In this interaction architecture, several natural input/output modalities (speech, pen, touch, hand gesture, eye gaze, and head and body movements) can be considered.

We have shown how abstract concepts, such as: i) a fission process using a pattern technique; and ii) a context-based method using a Bayesian network, can be used to develop applications.

We implemented two real applications to validate our approach, and showed that the proposed solution is applicable in a real environment. We presented two interfaces: 1) *a robot control interface*, which is implemented to validate the fission process using the pattern technique stored in the ontology, and 2) *a GPS interface*, which is implemented to validate the context in a case of uncertainly or ambiguity.

CONCLUSION

De nos jours, la technologie nous permet de produire des systèmes extensibles et entièrement contrôlés par l'humain. Ces systèmes sont équipés par des interfaces multimodales permettant une interaction plus naturelle et plus efficace entre la personne et la machine. Les utilisateurs peuvent profiter des modalités naturelles (la parole, le geste, les yeux, etc.) pour communiquer ou échanger des informations avec les applications ou les systèmes.

Ce projet de recherche vise la conception d'un module de fission, se basant sur des techniques efficaces, comme les patterns, le contexte, les réseaux bayésiens et les ontologies. Dans ce rapport, nous avons présenté une revue de la littérature sur ce thème. Les nombreux travaux existants et les caractéristiques de ce type de module montrent l'importance de la conception d'un module de fission dans le domaine de l'interaction multimodale. Suite à notre lecture de l'état de l'art, nous déduisons que les systèmes actuels utilisent des modalités prédéfinies. Ainsi les architectures proposées sont spécifiques aux applications ciblées.

D'autre part, comme dans tout projet de recherche scientifique, le but essentiel reste le caractère humain et social. En effet, toutes les applications développées ont pour objectif de faciliter l'utilisation pour les usagers, et particulièrement pour ceux ayant des besoins particuliers tels que les personnes âgées, les handicapées ou les malades. Pour cela, nous avons développé un module de fission qui peut être implémenté dans un robot pour l'assistance d'utilisateurs avec des besoins spécifiques.

Dans nos travaux de recherche, nous avons modélisé et implémenté une architecture de fission multimodale qui permet l'interaction entre l'homme et la machine. Cette architecture est fondée sur la compréhension de l'environnement et l'établissement de la relation entre ses différents objets. Notre architecture se veut évolutive. Elle est apte à accueillir de nouvelles technologies et de nouvelles modalités et peut s'adapter dynamiquement aux différentes variations de l'environnement. Le choix des modalités de sortie appropriées repose sur un

mécanisme de sélection selon le contexte d'interaction. Celui-ci est défini par trois éléments : l'environnement, le système et l'utilisateur.

Le processus de fission se base sur l'utilisation de deux techniques : pattern et réseau bayésien. Les patterns, se composent généralement de deux parties : problème et solution. Ils sont utilisés pour : 1) sélectionner les sous-tâches adéquates pour une commande complexe donnée et 2) associer à chaque sous-tâche la ou les modalités de sorties adéquates. Ces patterns sont stockés dans l'ontologie, en utilisant le langage OWL, un standard du W3C. Les réseaux bayésiens sont utilisés avec le contexte pour résoudre le problème d'ambiguïté.

Nous avons réalisé une architecture capable d'identifier les différentes modalités de sorties et de fissionner les données sur ces modalités en utilisant les patterns stockés dans des bases de connaissances. L'architecture proposée a été modélisée en utilisant le formalisme de réseau de Petri coloré et simulée par l'outil CPN-Tools. La méthodologie présentée dans cette thèse a été appliquée pour le développement de deux applications réelles afin de valider le bon fonctionnement du processus de fission.

Nous pensons que nos travaux peuvent contribuer à l'avancement de la recherche sur la fission multimodale dans le domaine de l'interaction machine-personne-machine. Ce projet constitue un moyen pour une meilleure compréhension de l'interaction homme/machine (simulation et validation de différents scénarios d'interaction). L'architecture que nous avons conçue peut-être déployée sur des robots, des appareils mobiles, ordinateurs, etc. Les patterns stockés dans nos bases de connaissances pourront être utilisés par d'autres chercheurs dans leurs propres travaux.

Un travail futur consistera à créer un système multimodal complet contenant le moteur de fusion et le module de fission et de le tester dans un environnement réel sur un robot. L'ontologie, présentée dans notre travail, détaille le contexte maison. Cette ontologie peut être mise à niveau pour cibler d'autres contextes tels que les hôpitaux, les lieux de travail, etc. Un des défis sera de trouver une méthode intelligente d'apprentissage capable d'enrichir

notre base de connaissance dynamiquement. Aussi, il serait intéressant d'implémenter une autre méthode pour résoudre l'ambiguïté et la comparer avec la méthode bayésienne utilisée dans ce travail.

ANNEXE I

DÉCLARATIONS DU RÉSEAU DU PETRI COLORÉ

Ces pages représentent les déclarations des variables et des fonctions de monitor dans CPN-Tools

```
colset INT = int;
colset UNIT = unit;
colset BOOL = bool;
colset STRING = string;
colset Action_Verb_List = list STRING ;
colset Bayes_Command_List = list STRING ;
colset prob = real;
colset listProb = list prob;
colset listProbs = list listProb;
colset listUncer = list STRING;
colset listUncerxlistProbs = product listUncer*listProbs;
colset listUncerxlistProb = product listUncer*listProb;
colset listProbxlistUncer = product  listProb*listUncer ;
var uncVar, BayUncer : listUncer;
var ProbVar: listProbs;
var probHisto,baycal , BayProbDe: listProb;
var wordCap : listUncer;
var valRe, resultCon,resultCons : prob;
val moiff = [["",""],["",""]];
colset modalitiesList = list STRING ;
colset meaningList = list STRING;
var command, desBay : STRING;
colset ListCommand = list STRING;
var  valChar, valCh, listMot, ListSubTask,supList : ListCommand;
```

```
var x, inde,indexF, indloop: INT;
var test_grammair, testStr, worList,ontoGramFI, worBay :STRING;
var verb_Ac: Action_Verb_List;
var lim, a, ma, loopi,loopj, leng,looph,loopk, n,j,k,iB,jB: INT;
var light, Active, Current, Noise, manual : BOOL;
var modelGram, wordTest:BOOL;
var handicap,moi, location,concatGram, concatGramF,feadFission: STRING;
var model,model1,model2,model3,model4,model5,model6,model7: STRING;
colset STRINGxSTRINGxSTRING = product STRING*STRING*STRING;
colset INTxSTRINGxSTRING = product INT*STRING*STRING;
colset INTxSTRINGxmeaningList= product INT*STRING*meaningList;
colset STRINGxmeaningList = product STRING*meaningList;
colset INTxSTRINGxBOOL = product INT*STRING*BOOL;
colset STRINGxSTRING = product STRING*STRING;
colset INTxSTRING = product INT*STRING;
colset STRINGxmodalitiesList= product STRING*modalitiesList;
var valOnto, valPattern,actionP,modelP,audioS, visualS, manualS,subtaskAsso, BayW:
STRING;
var toPatGram, toFeadGram,actionv,actionV, concatGramM: STRING;
colset ListPatternSolu= list STRING;
colset  basePattern = product INT*STRING*STRING*ListPatternSolu;
var listCom, listComB: Bayes_Command_List;
var meanWor, meanFin : meaningList;
var pattern, subTask,sub, patternP, ontoGram, gram,feadbackGram,desFead : STRING;
colset STRINGxBOOL = product STRING*BOOL;
var modality1, modality2,modality3: STRING;
var sizList,capter2: INT;
colset INTxBayes_Command_List = product INT*Bayes_Command_List;
colset STRINGxINTxSTRINGxSTRING = product STRING*INT*STRING*STRING;
val ProPattern1 = " AFP P MO ";
```

```
val ProPattern2 = " AFMO MO IL AMO ";
val ProPattern3 = " AFMO MO IL NMO ";
val ProPattern4 = " AFMO MO IL P ";
val ProPattern5 = " AFMO MO IL LO ";
val ProPattern6 = " AFP P ";
val ProPattern7 = " AFNMO NMO LO ";
val ProPattern8 = " AFL MO IL LO ";
val SoPattern1 = ["1-move to the destination context","2-move to the object",
"3-take the object","4-move to the position", "5-depose the object"];
var comNumb:INT;
val suprim = ["the", "of"];
var queryPatA,queryPatM, capter1, capter5,capter3, capter4: STRING;
```

BIBLIOGRAPHIE

Alexander, Christopher, S Ishikawa et M Silverstein. 1977a. « Pattern languages ». *Center for Environmental Structure,* vol. 2.

Alexander, Christopher, Sara Ishikawa et Murray Silverstein (1216). 1977b. *A Pattern Language: Towns, Buildings, Construction*. Center for Environmental Structure Series.

Alm, Torbjorn, Jens Alfredson et Kjell Ohlsson. 2009. « Simulator-based human-machine interaction design ». *International Journal of Vehicle Systems Modelling and Testing,* vol. 4, n° 1/2, p. 1-16.

Antoniou, Grigoris, et Frankvan Harmelen. 2009. « Web Ontology Language: OWL ». In *Handbook on Ontologies*, sous la dir. de Staab, Steffen, et Rudi Studer. p. 91-110. Coll. « International Handbooks on Information Systems »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-540-92673-3_4 >.

Atrey, Pradeep K, M Anwar Hossain, Abdulmotaleb El Saddik et Mohan S Kankanhalli. 2010. « Multimodal fusion for multimedia analysis: a survey ». *Multimedia Systems,* vol. 16, n° 6, p. 345-379.

Bechhofer, Sean, Ian Horrocks, Carole Goble et Robert Stevens. 2001. « OilEd: A Reasonable Ontology Editor for the Semantic Web ». In *KI 2001: Advances in Artificial Intelligence*, sous la dir. de Baader, Franz, Gerhard Brewka et Thomas Eiter. Vol. 2174, p. 396-408. Coll. « Lecture Notes in Computer Science »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/3-540-45422-5_28 >.

Beinhauer, Wolfgang, et Cornelia Hipp. 2009. « Using Acoustic Landscapes for the Evaluation of Multimodal Mobile Applications ». In *Human-Computer Interaction. Novel Interaction Methods and Techniques*. (San Diego, CA, USA), sous la dir. de Jacko, Julie Vol. 5611, p. 3-11. Coll. « Lecture Notes in Computer Science »: Springer Berlin / Heidelberg. < http://dx.doi.org/10.1007/978-3-642-02577-8_1 >.

Benoit, Alexandre, Laurent Bonnaud, Alice Caplier, Phillipe Ngo, Lionel Lawson, Daniela G. Trevisan, Vjekoslav Levacic, Céline Mancas et Guillaume Chanel. 2009. « Multimodal focus attention and stress detection and feedback in an augmented driver simulator ». *Personal and Ubiquitous Computing,* vol. 13, n° 1.

Bernsen, Niels Ole. 2008. « Multimodality Theory ». In *Multimodal User Interfaces*, sous la dir. de Tzovaras, Dimitrios. p. 5-29. Coll. « Signals and Commmunication Technologies »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-540-78345-9_2 >.

Bolt, R. 1980a. « Put-that-there ». *Voice and gesture at the graphics interface ACMSIGGRAPH Computer Graphics,* vol. 14, n° 3, p. 262-270.

Bolt, Richard A. 1980b. « “Put-that-there”: Voice and gesture at the graphics interface ». In *SIGGRAPH '80 Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. (New York, USA) Vol. Volume 14. < http://portal.acm.org/citation.cfm?doid=965105.807503 >.

Buckley, James J, et Esfandiar Eslami. 2002. *An introduction to fuzzy logic and fuzzy sets*. springer.

Carnielli, Walter, Pizzi et Claudio. 2008. « Modalities and Multimodalities ». *Springer,* vol. 12, n° 1.

Caschera, Maria Chiara, Alessia D'Andrea, Arianna D'Ulizia, Fernando Ferri, Patrizia Grifoni et Tiziana Guzzo. 2009. « ME: Multimodal Environment Based on Web Services Architecture ». In *OTM 2009 Workshops*. (Vilamoura, Portugal), p. 514-512. Springer.

Costa, David, et Carlos Duarte. 2011. « Adapting Multimodal Fission to User’s Abilities ». In *Universal Access in Human-Computer Interaction. Design for All and eInclusion*, sous la dir. de Stephanidis, Constantine. Vol. 6765, p. 347-356. Coll. « Lecture Notes in Computer Science »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-642-21672-5_38 >.

Costa, Paulo C. G. da, Kathryn B. Laskey et Kenneth J. Laskey. 2005. « PR-OWL: A bayesian ontology language for the semantic web ». In *In Proceedings of ISWC-URSW'*.

Costa, Paulo C. G., Kathryn B. Laskey et Ghazi AlGhamdi. 2006. « Bayesian ontologies in AI systems ». In *Uncertainty in Artificial Intelligence*. (Cambridge, MA, USA).

CPN-Tools. 2012. « CPN-Tools ». < http://cpntools.org/ >.

Daouadji, Abdelhamid. 2011. « Techniques intelligentes de découverte de ressources web ». École de technologie supérieure.

Debevc, M., P. Kosec, M. Rotovnik et A. Holzinger. 2009. « Accessible Multimodal Web Pages with Sign Language Translations for Deaf and Hard of Hearing Users ». In *Database and Expert Systems Application, 2009. DEXA '09. 20th International Workshop on*. (Aug. 31 2009-Sept. 4 2009), p. 279-283.

Dey, Anind K. 2001. « Understanding and using context ». *Personal and Ubiquitous Computing,* vol. 5, n° 1, p. 4-7.

Ding, Zhongli, et Yun Peng. 2004. « probabilistic extension to ontology language owl ». In *In Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37)*. (Big Island, Hawaii).

Djendi, Hicham. 2007. « Architecture et modèles dynamique dediés aux applications multimodales ». École Technologie supérieur, 390 p.

Djenidi, H., S. Benarif, A. Ramdane-Cherif, C. Tadj et N. Levy. 2004. « Generic multimedia multimodal agents paradigms and their dynamic reconfiguration at the architectural level ». *EURASIP J. Appl. Signal Process.,* vol. 2004, p. 1688-1707.

Dumas, Bruno, Denis Lalanne et Sharon Oviatt. 2009. « Multimodal Interfaces: A Survey of Principles, Models and Frameworks ». *Human Machine Interaction* vol. 5440/2009.

Efstratiou, Christos, Keith Cheverst, Nigel Davies et Adrian Friday. 2001. « An architecture for the effective support of adaptive context-aware applications ». In *Mobile Data Management*. Vol. 2001, p. 15-26. Springer.

Ertl, Dominik, Jürgen Falb et Hermann Kaindl. 2010. « Semi-Automatically Configured Fission For Multimodal User Interfaces ». In *Third International Conference on Advances in Computer-Human Interactions*. (Saint Maarten, Netherlands, Antilles).

Feiteira, Pedro, et Carlos Duarte. 2011. « Adaptive Multimodal Fusion ». In *Universal Access in Human-Computer Interaction. Design for All and eInclusion*, sous la dir. de Stephanidis, Constantine. Vol. 6765, p. 373-380. Coll. « Lecture Notes in Computer Science »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-642-21672-5_41 >.

Foster, Mary Ellen. 2002. *State of the art review: Multimodal Fission*. University of Edinburgh. COMIC project.

Foster, Mary Ellen. 2005. « Interleaved Preparation and Output in the COMIC Fission Module ». In *Software '05 Proceedings of the Workshop on Software* (Stroudsburg, PA, USA).

Friedman, Nir, Iftach Nachman et Dana Peér. 1999. « Learning bayesian network structure from massive datasets: the «sparse candidate «algorithm ». In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. p. 206-215. Morgan Kaufmann Publishers Inc.

Giuliani, Manuel, et Alois Knoll. 2008. « MultiML: a general purpose representation language for multimodal human utterances ». In *International Conference on Multimodal Interfaces* (Chania, Crete, Greece), p. 165-172. ACM.

Grifoni, Patrizia. 2009. « Multimodal fission ». In *Multimodal Human Computer Interaction and Pervasive Services*, sous la dir. de Grifoni, Patrizia, VI: IGI Global.

Grone, Bernhard. 2006. « Conceptual Patterns ». In *ECBS '06 Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems* (Washington, DC, USA).

Gruber, T. 1991. « The Role of a Common Ontology in Achieving Sharable, Reusable Knowledge Bases ». In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*. (Cambridge).

Gruber, T. R. 1993. « Towards principles for the design of ontologies used for knowledge sharing ». In *Formal Ontology in Conceptual Analysis and Knowledge Representation*. (The Netherlands: Kluwer Academic).

Gu, T., H. K. Pung et D. Q. Zhang. 2004. « A bayesian approach for dealing with uncertain contexts ». In *Proceedings of the 2nd International Conference on Pervasive Computing*. (Austrian Computer Society).

Guarino, Nicola, Daniel Oberle et Steffen Staab. 2009. « What Is an Ontology? ». In *Handbook on Ontologies*, sous la dir. de Staab, Steffen, et Rudi Studer. p. 1-17. Coll. « International Handbooks on Information Systems »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-540-92673-3_0 >.

Henry, Tyson R., Scott E. Hudson et Gary L. Newell. 1990. « Integrating gesture and snapping into a user interface toolkit ». In *Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology*. (Snowbird, Utah, USA), p. 112-122. 97938: ACM.

Hibou, Mathieu. 2006. « Réseaux bayésiens pour la modélisation de l'apprenant en eiah: modèles multiples versus modèle unique ». *Actes 1res rencontres jeunes chercheurs en EIAH*.

Hina, Manolo Dulva, Chakib Tadj, Amar Ramdane-Cherif et Nicole Levy. « A Multi-Agent based Multimodal System Adaptive to the User's Interaction Context ». *Multi-Agent Systems–Modeling, Interactions, Simulations and Case Studies*, p. 29-56.

Hina, Manolo Dulva, Chakib Tadj, Amar Ramdane-Cherif et Nicole Levy. 2011. « A Multi-Agent based Multimodal System Adaptive to the User's Interaction Context ». In *Multi-Agent Systems*. INTECH.

Horrocks, Ian. 2002. « DAML+OIL: A Description Logic for the Semantic Web ». *IEEE Data Engineering Bulletin,* vol. 25, n° 1, p. 4-9.

Huajun, Chen, Wu Zhaohui, Wang Heng et Mao Yuxin. 2006. « RDF/RDFS-based Relational Database Integration ». In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*. (03-07 April 2006), p. 94-94.

Huiqun, Zhao, Zhang Shikan et Zhao Junbao. 2012. « Research of Using Protege to Build Ontology ». In *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on*. (May 30 2012-June 1 2012), p. 697-700.

Jacob, Mithun George, Yu-Ting Li et Juan P Wachs. 2012. « Gestonurse: a multimodal robotic scrub nurse ». In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. p. 153-154. ACM.

Jaimes, Alejandro, et Nicu Sebe. 2007. « Multimodal human–computer interaction: A survey ». *Computer vision and image understanding,* vol. 108, n° 1, p. 116-134.

Java. 2013. « Java Socket ».
< http://docs.oracle.com/javase/tutorial/networking/sockets/definition.html >.

Jensen, Kurt. 1987. « Coloured Petri nets
Petri Nets: Central Models and Their Properties ». In, sous la dir. de Brauer, W., W. Reisig et G. Rozenberg. Vol. 254, p. 248-299. Coll. « Lecture Notes in Computer Science »: Springer Berlin / Heidelberg. < http://dx.doi.org/10.1007/BFb0046842 >.

Jensen, Kurt. 1996. *Coloured Petri nets: basic concepts, analysis methods and practical use*, 1. Springer.

Ji, Qiang, Zhiwei Zhu et Peilin Lan. 2004. « Real-time nonintrusive monitoring and prediction of driver fatigue ». *Vehicular Technology, IEEE Transactions on,* vol. 53, n° 4, p. 1052-1068.

Karpov, A., A. Ronzhin, I. Kipyatkova et L. Akarun. 2010. « Multimodal Human Computer Interaction with MIDAS Intelligent Infokiosk ». In *Pattern Recognition (ICPR), 2010 20th International Conference on*. (23-26 Aug. 2010), p. 3862-3865.

Knublauch, Holger, RayW Fergerson, NatalyaF Noy et MarkA Musen. 2004. « The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications ». In *The Semantic Web – ISWC 2004*, sous la dir. de McIlraith, SheilaA, Dimitris Plexousakis et Frank Harmelen. Vol. 3298, p. 229-243. Coll. « Lecture Notes in Computer Science »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-540-30475-3_17 >.

Lai, Jennifer, Stella Mitchell et Christopher Pavlovski. 2007. « Examining modality usage in a conversational multimodal application for mobile e-mail access ». *International Journal of Speech Technology,* vol. 10, n° 1, p. 17-30.

Lalanne, Denis, Laurence Nigay, philippe Palanque, Peter Robinson, Jean Vanderdonckt et Jean-François Ladry. 2009. « Fusion Engines for Multimodal Input: A Survey ». In *International Conference on Multimodal Interfaces*. p. 153-160. ACM.

Landragin, Frédéric. 2007. « Physical, semantic and pragmatic levels for multimodal fusion and fission ». In *Seventh International Workshop on Computational Semantics* (Tilburg, The Netherlands).

Laput, Gierad P., Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder et Eytan Adar. 2013. « PixelTone: a multimodal interface for image editing ». In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (Paris, France), p. 2185-2194. 2481301: ACM.

Lauer, Claire. 2009. « Contending with Terms: "Multimodal" and "Multimedia" in the Academic and Public Spheres ». *Computers and Composition,* vol. 26, n° 4, p. 225-239.

Lehnert, Wendy G. 1978. *The Process of Question Answering: A Computer Simulation of Cognition*. L. Erlbaum Associates.

Little, Thomas D. C., C. Y. Roger Chen, C. S. Chang et P. Bruce Berra. 1991. « Multimedia Synchronization ». *EEE Data Eng. Bull.,* vol. 14, p. 26-35.

Meng, H., S. Oviatt, G. Potamianos et G. Rigoll. 2009. « Introduction to the Special Issue on Multimodal Processing in Speech-Based Interactions ». *Audio, Speech, and Language Processing, IEEE Transactions on* vol. 17, n° 3, p. 409 - 410

Miraoui, Moeiz, Chakib Tadj et Chokri ben Amar. 2008. « Context modeling and context-aware service adaptation for pervasive computing systems ». *International Journal of Computer and Information Science and Engineering,* vol. 2, n° 3, p. 148-157.

Mukaidono, Masao. 2001. *Fuzzy logic for beginners*. World Scientific.

Nguyen, Laurent, Jean-Marc Odobez et Daniel Gatica-Perez. 2012. « Using self-context for multimodal detection of head nods in face-to-face interactions ». In *Proceedings of the 14th ACM international conference on Multimodal interaction*. (Santa Monica, California, USA), p. 289-292. 2388734: ACM.

Nordahl, Rolf, Stefania Serafin, Luca Turchet et Niels Christian Nilsson. 2012. « A multimodal architecture for simulating natural interactive walking in virtual environments ». *PsychNology,* vol. 9, n° 3, p. 245-268.

O.Coplien, James, et Neil B. Harission. 2005. *Organizational Patterns of Agile Software Developpement*. Pearson Prentice Hall.

Oviatt, S. , et Seattle Incaa Designs. 2007. « Implicit user-adaptive system engagement in speech, pen and multimodal interfaces ». In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*. (Kyoto), p. 496-501. IEEE.

Oviatt, S., P. Cohen, Lizhong Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson et D. Ferro. 2000a. « Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions ». *Human-Computer Interaction,* vol. 15, n^{o} 4, p. 263-322.

Oviatt, Sharon. 1999. « Ten myths of multimodal interaction ». *Communications of the ACM CACM Homepage,* vol. 42, n^{o} 11.

Oviatt, Sharon. 2003. « Multimodal interfaces ». In *The human-computer interaction handbook*, sous la dir. de Julie, A. Jacko, et Sears Andrew. p. 286-304. L. Erlbaum Associates Inc.

Oviatt, Sharon, Phil Cohen, Lizhong Wu, John Vergo, Lisbeth Duncan, Bernhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson et David Ferro. 2000b. « Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions ». In *Human-Computer Interaction*. Vol. 15, p. 263–322. Lawrence Erlbaum Associates.

Palanque, Philippe, et Amélie Schyn. 2003. « A Model-Based Approach for Engineering Multimodal Interactive Systems ». In *9 th IFIP TC13 Int. Conf. on Human-Computer Interaction* IOS Press.

Perroud, Didier, Leonardo Angelini, Omar Abou Khaled et Elena Mugellini. 2012. « Context-Based Generation of Multimodal Feedbacks for Natural Interaction in Smart Environments ». In *AMBIENT 2012, The Second International Conference on Ambient Computing, Applications, Services and Technologies*. p. 19-25.

Pfleger, Norbert. 2004. « Context Based Multimodal Fusion ». In *ICMI 04*. (Pennsylvannia, USA), p. 265 - 272. ACM.

Poller, Peter, et Valentin Tschernomas. 2006. « Multimodal Fission and Media Design ». In *SmartKom: Foundations of Multimodal Dialogue Systems*, sous la dir. de Wahlster, Wolfgang. Springer Berlin Heidelberg.

Portillo, Pilar Manchón, Guillermo Pérez García et Gabriel Amores Carredano. 2006. « Multimodal fusion: a new hybrid strategy for dialogue systems ». In *Proceedings of the 8th international conference on Multimodal interfaces* (Banff, Alberta, Canada), p. 357 - 363.

Pous, M., et L. Ceccaroni. 2010. « Multimodal Interaction in Distributed and Ubiquitous Computing ». In *Fifth International Conference on Internet and Web Applications and Services* (Barcelona, Spain), p. 457-62.

Raisamo, Roope, Arto Hippula, Saija Patomaki, Eva Tuominen, Virpi Pasto et Matias Hasu. 2006. « Testing usability of multimodal applications with visually impaired children ». *MultiMedia, IEEE,* vol. 13, n° 3, p. 70-76.

Ringland, S. P A, et F. J. Scahill. 2002. « Multimodality — The Future of the Wireless User Interface ». *BT Technology Journal,* vol. 21, n° 3.

Robert, Steele, Khankan Khaled et Dillon Tharam. 2005. « Mobile Web Services Discovery and Invocation Through Auto-Generation of Abstract Multimodal Interface ». In *ITCC 2005 International conference on Information Technology*. (Las Vegas, NV). Vol. 35-43. IEEE.

Rousseau, Cyril, Yacine Bellik, Frédéric Vernier et Didier Bazalgette. 2006. « A Framework for the Intelligent Multimodal Presentation of Information ». *Signal Processing,* vol. 86, n° 12.

Saporta, Gilbert. 2006. *Probabilités, analyses des données et statistiques*. Editions Technip.

Sears, Andrew, et Julie A. Jacko ( ). 2007. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, Second Edition. CRC Press, 1,384 p.

Sowa, John F. 1995. « Top-level ontological categories ». *International journal of human-computer studies,* vol. 43, n° 5, p. 669-685.

Sure, York, Juergen Angele et Steffen Staab. 2002. « OntoEdit: Guiding Ontology Development by Methodology and Inferencing ». In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, sous la dir. de Meersman, Robert, et Zahir Tari. Vol. 2519, p. 1205-1222. Coll. « Lecture Notes in Computer Science »: Springer Berlin Heidelberg.
< http://dx.doi.org/10.1007/3-540-36124-3_76 >.

University, Standford. 2013a. « OKBC ». < http://www.ksl.stanford.edu/software/OKBC/ >.

University, Stanford. 2013b. « Ontolingua ».
< http://www.ksl.stanford.edu/software/ontolingua/ >.

Uschold, Mike, et Michael Gruninger. 1996. « Ontologies: Principles, methods and applications ». *Knowledge engineering review,* vol. 11, n° 2, p. 93-136.

Wahlster, Wolfgang. 2003. « Towards symmetric multimodality: Fusion and fission of speech, gesture, and facial expression ». In *KI 2003: Advances in Artificial Intelligence*. p. 1-18. Springer.

Wang, Danli, Jie Zhang et Guozhong. 2006. « A Multimodal Fusion Framework for Children's Storytelling Systems ». In *LNCS* Vol. 3942/2006, p. 585-588. Berlin / Heidelberg: Springer-Verlag.

wikipedia. 2011. « Ontology ». < http://en.wikipedia.org/wiki/Ontology_(information_science) >.

Wilensky, R.W. 1978. « Understanding Goal-Based Stories ». Yale University, 240 p.

Xu, Tianling, Kaiguo Yuan, Jingzhong Wang, Xinxin Niu et Yixian Yang. 2009. « A real-time information hiding algorithm based on HTTP protocol ». In *Conference on Network Infrastructure and Digital Content, IEEE IC-NIDC2009*. (Beijing, China), p. 618-622. IEEE.

Yuen, P C, Y Y Tang et P S P Wang. 2002. *MULTIMODAL INTERFACE FOR HUMAN-MACHINE COMMUNICATION*, 48. World Scientific Publishing.

Zadeh, Lotfi A. 1965. « Fuzzy sets ». *Information and control,* vol. 8, n° 3, p. 338-353.

Zaguia, Atef, Manolo Dulva Hina, Chakib Tadj et Amar Ramdane-Cherif. 2010a. « Interaction context-aware modalities and multimodal fusion for acessing web services ». *Ubiquitous Computing and Communication Journal,* vol. 5, n° 4.

Zaguia, Atef, Manolo Dulva Hina, Chakib Tadj et Amar Ramdane-Cherif. 2010b. « Using Multimodal Fusion in Accessing Web Services ». *Journal of Emerging Trends in Computing and Information Sciences,* vol. 1, n° 2, p. 121-138.

Zaguia, Atef, Manolo Hina, Chakib Tadj et Amar Ramdane-Cherif. 2010c. « Interaction Context-Aware Modalities and Multimodal Fusion for Accessing Web Services ». *Ubiquitous Computing and Communication Journal,* vol. 5.

Zaguia, Atef, Chakib Tadj et Amar cherif-ramadhan. 2012. « Architecture générique pour le processus de la fission multimodale ». In *25th Canadian Conference on Electrical and Computer Engineering*. (Montreal). IEEE.

Zaguia, Atef, Chakib Tadj et Amar Ramdane-Cherif. 2013. « Context-Based method using Bayesian Network in multimodal fission system ». *Submitted*.

Zaguia, Atef, Ahmad Wahbi, Moeiz Miraoui, Chakib Tadj et Amar Ramdane-Cherif. 2013a. « Modeling Rules Fission and Modality Selection Using Ontology ». *Journal of Software Engineering and Applications,* vol. 7, n° 6, p. 354-371.

Zaguia, Atef, Ahmad Wahbi, Chakib Tadj et Amar Ramdane-Cherif. 2013b. « Multimodal Fission For Interaction Architecture ». *Journal of Emerging Trends in Computing and Information Sciences,* vol. 4, n° 1.

Zhu, Lulu, Weiqin Tong et Bin Cheng. 2011. « CPN Tools' Application in Verification of Parallel Programs Information Computing and Applications ». In *Information Computing and Applications*. Vol. 105, p. 137-143. Coll. « Communications in Computer and Information Science »: Springer Berlin Heidelberg. < http://dx.doi.org/10.1007/978-3-642-16336-4_19 >.