

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M. Ing.

PAR
Atef ZAGUIA

FUSION MULTIMODALE ET SERVICES WEB

MONTRÉAL, LE 25 OCTOBRE 2010

© Tous droits réservés, Atef Zaguia, 2010

PRÉSENTATION DU JURY
CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

M. Chakib Tadj, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Amar Ramdane-Cherif, codirecteur de mémoire
Laboratoire LISV à l'Université de Versailles-Saint-Quentin-en-Yvelines, France

M. Tony Wong, président du jury
Département de génie de la production automatisée à l'École de technologie supérieure

M. Roch Glitho, membre du jury
Département de génie informatique et des technologies de l'information à l'Université Concordia.

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 19 OCTOBRE 2010

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

À mon père et ma mère,

Qui m'ont soutenu tout le long de ma scolarité et qui m'ont toujours encouragé.

REMERCIEMENTS

Ce travail de recherche, ce mémoire, ne pourraient exister sans un certain nombre de personnes auxquelles je tiens à exprimer ici ma sincère gratitude.

Tout d'abord, je remercie vivement mon directeur de recherche Monsieur Tadj Chakib et mon co-directeur Monsieur Amar Ramdane-Cherif, pour leurs directives, leurs aide précieuse, pour avoir pris de leurs temps afin de m'aider et de m'encourager et je les remercie surtout pour leurs bons conseils et leurs appui tout au long de l'élaboration de ce projet.

Je tiens à remercier vivement tous les membres du jury de ma soutenance, pour l'attention qu'ils ont apportée à mon travail et pour leurs précieuses remarque grâce auxquelles j'ai pu améliorer ce mémoire. Merci aussi à M. Tony Wong de m'avoir fait l'honneur de présider ce jury. Merci aussi à M. Roch Glitho d'avoir accepté de faire partie du jury de mémoire et de m'avoir fait profiter de ses conseils éclairés.

Je remercie également mes très chers parents, mes deux frères (Imed, Issam) et ma sœur (Dorsaf) qui m'ont toujours soutenu même si parfois l'inquiétude était perceptible. Je remercie également tous ceux qui m'ont apporté leur soutien et aide (Oncle Nejib et sa famille) et ceux qui ont contribué à l'élaboration de ce mémoire.

Je tiens à dire un merci tout particulier à mon collègue de laboratoire M. Hina Manolo, mais aussi à mes amis personnels, qui ont tous su, chacun à sa façon, apporter un indispensable souffle à ce travail : Randa, Bassem, Heithem et Heykel.

FUSION MULTIMODALE ET SERVICES WEB

Atef ZAGUIA

RÉSUMÉ

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 (le parole, le geste, les yeux, ..., etc.) pour communiquer ou échanger des informations avec les applications ou les systèmes. L'utilisation des applications multimodales avec les services Web, intégrée avec des modalités naturelles, est une solution efficace pour les utilisateurs, en particulier pour ceux qui ne peuvent pas utiliser un clavier ou une souris, les utilisateurs malvoyants, les utilisateurs qui sont équipés d'appareils mobiles, les utilisateurs qui sont infirmes, ..., etc. Notre travail présente une approche dans laquelle les modalités d'entrées variantes (parole, écran tactile, clavier, etc.) sont mises à la disposition de l'utilisateur dans le but d'accéder aux services Web. La technique utilisée dans l'état d'art présente une approche avec deux modalités prédéfinie. Notre approche permet l'utilisation d'un nombre infini de modalités (d'un point de vue conceptuel) en utilisant un niveau sémantique appelé « Multimodal Fusion ». Une telle approche donne à l'utilisateur la possibilité d'utiliser les modalités qu'il juge appropriée à sa situation. La description détaillée de l'approche proposée, ainsi que l'application qui a été élaboré, sont présentées dans ce mémoire.

Mots-clés : fusion multimodal, services Web, interface homme-machine, interaction homme-machine.

MULTIMODAL FUSION AND WEB SERVICES

Atef ZAGUIA

ABSTRACT

In our days, the technology allows us to produce extended and totally human-controlled multimodal systems. These systems are equipped with multimodal interfaces allowing a more natural and more efficient interaction between man and machine. End users could take advantage of natural modalities (e.g. eye gaze, speech, gesture, etc.) to communicate or exchange information with applications. The use of multimodal applications in Web services, integrated with natural modalities, is an effective solution for users who cannot use a keyboard or a mouse, on users who have visual handicap, on mobile users equipped with wireless telephone/mobile devices, on weakened users, etc. Our work presents an approach in which various input modalities (speech, touch screen, keyboard, eye gaze, etc.) are available at user's disposition in order to access Web services. While current state-of-the-art uses two pre-defined modalities, our approach allows a maximum number of concurrent modalities using semantic level called "multimodal fusion". Such approach gives user the flexibility to use the modalities as he sees fit for his situation. The detailed description of the proposed approach as well as the application that has been developed that uses these multimodalities is presented in this work.

Keywords: multimodal fusion, web service, human-computer interface, human-computer interaction.

TABLE DES MATIÈRES

[illegible]

1.5.3	Les langages multimodaux :	38
1.5.3.1	EMMA (Extended MultiModal Annotation markup language).....	38
1.5.3.2	X + V (XHTML plus Voice).....	39
1.5.4	Les langages pour les interactions gestuelles.....	39
1.5.4.1	InkML (Ink Markup Language Tags)	39
1.5.5	Conclusion	39
1.6	Application multimodales.....	40
1.6.1	Introduction.....	40
1.6.2	Applications multimodales utilisant le service Web.....	41
1.6.2.1	Les évènements catastrophiques	41
	Scenario 1	41
	Scenario 2	42
	Scenario 3	42
1.6.2.2	Réservation de voitures de location.....	43
1.6.2.3	Accès aux services.....	44
1.6.3	Applications multimodales qui peuvent être implémentée avec service Web...	44
1.6.3.1	Système de conception de salle de bain	44
1.6.3.2	Système multimodal utilise dans le domaine robotique.....	46
1.6.3.3	Enseignement assisté par ordinateurs.....	47
1.6.4	Conclusion	47
	CHAPITRE 2 FUSION MULTIMODALE DE L'INTERACTION : CONTRIBUTION.....	49
2.1	Architecture et approche	49
2.2	Fusion multimodale	52
2.3	Élément temps dans le processus de fusion	57
2.4	Spécification et composants de fusion multimodale.....	60
2.4.1	L'interface utilisateur.....	61
2.4.2	L'analyseur	62
2.4.3	La base de données	65
2.4.4	Exemple et simulation à l'aide de réseaux de Petri.....	73
2.4.4.1	Scénario	75
2.4.4.2	Grammaire.....	76
2.4.4.3	Simulation 1	76
2.4.4.4	Simulation 2	77
2.4.4.5	Simulation 3	77
2.5	Implémentation	84
2.6	Conclusion	86
	CONCLUSION.....	87
	LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES	89

LISTE DES TABLEAUX

	Page
Tableau 1.1 Comparaison entre SAX et DOM	36
Tableau 1.2 Récapitulatif des différents langages	40
Tableau 2.1 Modèle d'un tableau de « MODALITY »	68
Tableau 2.2 Modèle d'un tableau « Modality_Main_Parameters ».....	69
Tableau 2.3 Modèle d'un tableau « Union_Modality_Main_Parameters ».....	70
Tableau 2.4 Modèle d'un tableau de « fusion ».....	70
Tableau 2.5 Modèle d'un tableau de « Union_Fusion_Main_Parameters ».....	71
Tableau 2.6 Modèle d'un tableau de «Fusion_Main_Parameters ».....	72
Tableau 2.7 Paramètres ajoutés de la modalité.....	72
Tableau 2.8 Caractéristiques des ordinateurs utilisés	85

LISTE DES FIGURES

	Page
Figure 1.1 Deux Stratégies pour la fusion multimodale.	19
Figure 1.2 Protocoles de services Web.	21
Figure 1.3 Principe général de fonctionnement.	23
Figure 1.4 Couplage des applications.	26
Figure 1.5 Offre de fonctionnalité via les services Web.....	26
Figure 1.6 Structure de code XML.	30
Figure 1.7 SAX analyseur XML.	33
Figure 1.8 DOM analyseur XML.....	34
Figure 1.9 Fichier XML.....	34
Figure 1.10 Exemple d'arbre DOM.....	35
Figure 1.11 Démonstration du système.	45
Figure 1.12 Exemple de capture d'écran de fonctionnement du système.	46
Figure 1.13 Exemple d'interaction multimodal entre un robot et un humain.	47
Figure 2.1 Architecture du système de fusion multimodale pour accéder aux services Web.	50
Figure 2.2 Framework de la fusion multimodale.....	51
Figure 2.3 Fusion de deux modalités.	52
Figure 2.4 Le processus détaillé de la fusion.	54
Figure 2.5 Agent de Fusion.....	55
Figure 2.6 Algorithme de Fusion.	57
Figure 2.7 Le facteur temps impliqué dans deux modalités.	58
Figure 2.8 Le facteur temps impliqué dans trois modalités.	59

Figure 2.9 L'interface utilisateur du système.	62
Figure 2.10 Modèle de fichier XML (EMMA) contenant des modalités (parole et écran tactile).	64
Figure 2.11 (gauche) le fichier XML combiné résultant et (droite) le fichier XML résultant après la fusion.	65
Figure 2.12 Diagramme de classe de la base de Données.	67
Figure 2.13 Modèle d'un fichier XML.	67
Figure 2.14 Symboles et notations des réseaux de Petri.	74
Figure 2.15 Exemple de réseau de Petri.	74
Figure 2.16 (A) Menu principal de la location de voiture (B) Modalités disponibles (C) Informations concernant date de locations (D) Informations concernant locateur.	78
Figure 2.17 Réseau de Petri montre les entrées pour le mois , le jour et le temps en utilisant des modalités différentes.	79
Figure 2.18 Un simple scénario montre les interactions multimodales entre l'utilisateur et la machine.	80
Figure 2.19 Grammaire pour la période de location de voiture.	80
Figure 2.20 Les activités du système quand l'utilisateur choisi "Rent" en utilisant la voix.	81
Figure 2.21 Réseaux de Petri montre toutes les variations possibles pour les données entrées pour mois, jour et temps en utilisant parole, clavier et écran tactile pour la sélection de "Rent".	82
Figure 2.22 Réseau de Petri présente toutes les opérations impliquant différentes modalités que l'utilisateur choisit pour "Start Month ", " Start Day "et" Start Time "	83
Figure 2.23 Réseau de Petri pour la confirmation des transactions.	84
Figure 2.24 Implémentation du Système.	86

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	Application Programming Interface
B2B	business to business
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
DOM	Document Object Model
EAI	Entreprise Application Intégration
EDI	échange de données informatisées
EMMA	Extended MultiModal Annotation markup language
FTP	File Transport Protocole
HTML	HyperText Markup Language
http	HyperText Transfert Protocol
InkML	Ink Markup Language Tags
MPML	Multimodal Presentation Markup Language
PLS	Pronunciation Lexicon Specification
RdP	Réseau de Petri
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SAX	Simple API for XML
SOAP	Simple Object Access Protocol
SMTP	Simple Mail Transfer Protocol
SRGS	Speech Recognition Grammar Specification

SSML	Speech Synthesis Markup Language
VoiceXML	Voice eXtensible Markup Language
UDDI	Universal Description, Discovery and Integration
W3C	est un groupement d'entreprises reconnues dans le domaine informatique, qui a comme principal objectif de développer et de promouvoir des technologies propre à internet
WSDL	Web Services Description Language
XML	Extensible Markup language

INTRODUCTION

Problématique

Ce mémoire 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 d'entrée, le système doit être capable de choisir parmi les modalités disponibles, celles qui satisfont au mieux les contraintes environnementales, les besoins fonctionnels de la tâche à exécuter et les préférences de l'utilisateur. En intégrant les événements multimodaux de l'entrée à un niveau sémantique appelé « fusion multimodale », le système pourra réaliser une interprétation concomitante de plusieurs modalités qui va aider à lever des ambiguïtés de l'interprétation. L'interface offre plusieurs fonctionnalités multimodales (par exemple audio et visuelle) dans la présentation de l'information à l'utilisateur (interaction personne-machine) ou à la machine (interaction machine-machine). Dans ce cas, nous parlons de processus de fusion. L'interaction personne-machine / machine-machine peut être bidirectionnelle, symétrique et entrelacée. Elle s'apparente à l'interaction et la communication humaine.

Contribution

Notre approche consiste à développer un système flexible capable de manipuler plusieurs modalités. L'approche consiste à développer un système composé de plusieurs modules. Ce système devrait permettre la détection des modalités utilisées, prendre en compte les paramètres de chacune d'elles et d'effectuer la fusion de celles-ci dans le but de déterminer et

d'entreprendre l'action désirée. Il est entendu que l'action désirée est une requête multimodale.

Cette approche est plus souple que celles trouvées dans l'état de l'art. En effet, toutes les approches décrites dans la littérature fonctionnent avec deux modalités prédéfinies (trois dans de très rares cas).

La fusion des modalités impliquées dans ce travail est fondée sur les médias et les modalités choisies par l'utilisateur. Ce choix dépend des préférences de celui-ci, du matériel disponible et de la situation dans laquelle il se trouve.

Structure du document

La suite de ce mémoire est organisée de la manière suivante : dans le chapitre 1 nous présentons l'état d'art et les différentes recherches reliées à notre projet.

Le chapitre 2 décrit notre méthodologie envisagée pour résoudre notre problématique, en décrivant l'approche qui permet de réaliser la fusion en utilisant les services Web. Nous présentons à la fin du chapitre un exemple d'application.

Nous concluons ce mémoire par un résumé de notre contribution et nous présenterons quelques travaux que nous avons l'intention de réaliser dans le futur.

CHAPITRE 1

PRÉSENTATION ET ANALYSE CRITIQUE DE L'ÉTAT DE L'ART RELATIF À NOTRE PROBLÉMATIQUE DE RECHERCHE

1.1 Caractérisation des interactions multimodales

1.1.1 Définition et terminologie du media, de la modalité et de la multimodalité

1.1.1.1 Le media

Dans le dictionnaire de la langue française (Robert 2010), le media 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 media et chaque auteur possède sa propre conception de media :

« *Une finalité purement matérielle* : le media 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 media est considéré comme un procédé physique et/ou logiciel utilisable comme véhicule ou support d'information

Une finalité 'technico-humaine' à deux niveaux d'abstraction : le media est défini comme un couplage d'un dispositif physique, aux qualités sensorielles humaines, et aussi comme un système représentationnel » (Djendi 2007, p.12).

1.1.1.2 La modalité

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

«

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 dépendant :
 - 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 media les supports ou les véhicules de l'information :

- Média : microphone, écran, clavier, etc.
- Modalité : parole, vision, etc.

1.1.1.3 La multimodalité

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

Elle 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 and 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.

Quand un système multimodal (Djendi 2007) reçoit des informations en provenance de différentes entrées, il fait une analyse puis une interprétation des données. Les avantages des systèmes multimodaux se résument aux points suivants :

- La capacité d'interprétation;
- La capacité de compréhension des informations.

1.1.1.4 La Fusion

La *fusion* (Pfleger 2004; Pérez, Amores et al. 2005) est une combinaison logique de deux ou plusieurs entités, dans notre cas deux ou plusieurs modalités. Les signaux des modalités sont interceptés par l'agent de fusion, puis combinés ensemble en se basant sur certaines règles sémantiques.

Nous trouvons dans la littérature deux types de stratégies de fusion : *fusion précoce* (*early fusion*) (Wöllmer, Al-Hames et al. 2009) et *fusion tardive* (*late fusion*). La *fusion précoce* (Snoek, Worring et al. 2005) réfère à la fusion qui intègre les caractéristiques de chaque modalité avant le concept d'apprentissage comme le montre la Figure 1.1 b). Cette fusion se produit au niveau du signal (Oviatt, Cohen et al. 2000).

La *fusion tardive* (*late fusion*) (Mohan, Dhananjaya et al. 2008) est réalisée au niveau sémantique. Figure 1.1 a)

Plusieurs méthodes de fusion ont été traitées dans la littérature. Certaines utilisent le calcul probabiliste (Pitsikalis, Katsamanis et al. 2009) ou notion de score. Nous pouvons citer par exemple le théorème de Bayes (Zhang and Guan 2009) et SVM (Support Vector Machines) (Fierrez-Aguilar, Ortega-Garcia et al. 2003). Cependant, ces techniques sont limitées aux modalités images et vidéos. D'autres chercheurs ont utilisé des architectures combinées avec une grammaire (Portillo, García et al. 2006; Sun, Chen et al. 2006).

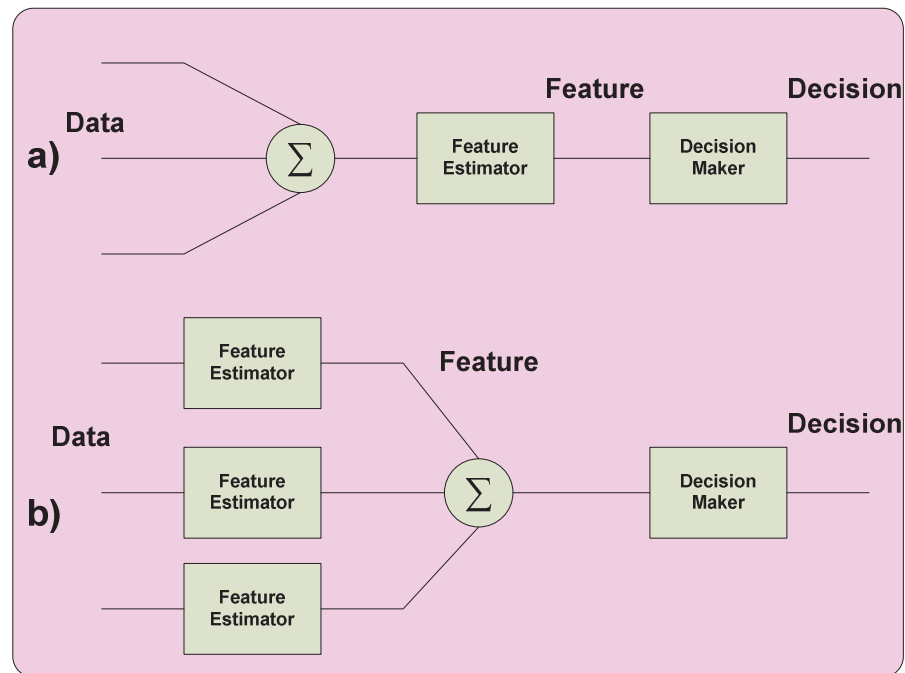


Figure 1.1 Deux Stratégies pour la fusion multimodale.

Tirée de Paleari and Antipolis (2006)

Les systèmes multimodaux donnent un sens à une requête suite à la combinaison multimodale et l'interprétation des modalités. La fusion multimodale est donc une étape cruciale dans la conception des systèmes multimodaux interactifs.

1.2 Service Web

1.2.1 Introduction

Inclus depuis 2001 aux environnements d'intégration du marché, les services Web (W3Cq 2010) ont donné naissance à de nouvelles problématiques d'administration : mise en œuvre de politiques d'accès, de supervision, de gestion des transactions, etc. Une fois un Service Web déployé, se pose en effet la question de son exploitation (Pagni, Hau et al. 2008).

Les services Web sont basés sur une collection de standards et de protocoles. Ils nous permettent de faire traiter les demandes des systèmes distants en utilisant un langage

commun et non à propriété industrielle, et des protocoles de transport comme http (Xu, Yuan et al. 2009) et SMTP (Sureswaran, Al Bazar et al. 2009).

1.2.2 Définitions

Un service Web (Daniel 2003) est un composant logiciel représentant une fonction applicative (ou un service applicatif). Il peut être accessible depuis une autre application (un client, un serveur ou un autre service Web) à travers le réseau Internet en utilisant les protocoles de transports disponibles. Ce service applicatif peut être implémenté comme une application autonome ou comme un ensemble d'applications. Il s'agit d'une technologie permettant à des applications de dialoguer à distance via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent. Pour ce faire, les services Web s'appuient sur un ensemble de protocoles comme le montre la Figure 1.2 standardisant les modes d'invocation mutuels de composants applicatifs. Ces protocoles sont répartis selon quatre axes :

Couche de transport : cette couche s'occupe de transporter les messages entre les applications. Actuellement, ceci s'effectue par le biais de HTTP (HyperText Transfert Protocol) (W3Ce 2010), FTP (File Transport Protocole) (W3Cd 2010) ou SMTP (Simple Mail Transfer Protocol) (Tzerefos, Smythe et al. 1997). D'autres protocoles sont en cours d'intégration.

Messages XML : il s'agit de formaliser les messages à l'aide d'un vocabulaire XML commun. Actuellement, ceci s'effectue par le biais de SOAP (Simple Object Access Protocol) (W3Cf 2010).

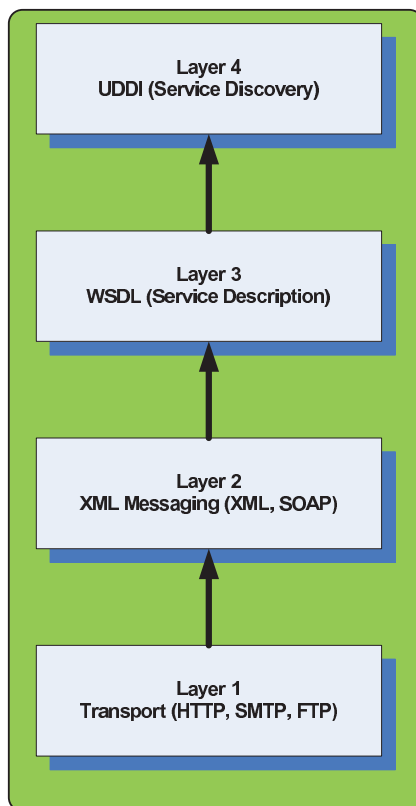


Figure 1.2 Protocoles de services Web.

Tirée de : http://www.softteam.fr/technologies_web_services.php

Description des services : il s'agit de la description de l'interface publique des services Web. Actuellement, ceci s'effectue par le biais de WSDL (Web Services Description Language) (W3Cg 2010).

Recherche de service : il s'agit de la centralisation des services Web (de leurs descriptions) dans un référentiel commun. Actuellement, ceci s'effectue par le biais d'UDDI (Universal Description, Discovery and Integration).

1.2.3 Standards

1.2.3.1 WSDL (Web Service Description Language)

Fournit un mode de description des composants applicatifs permettant d'invoquer leurs fonctions à distance. Cette description de l'interface publique d'utilisation des services Web repose sur le langage XML.

1.2.3.2 UDDI (Universal Description Discovery and Integration)

Fournit les références des connexions permettant d'invoquer dynamiquement et à distance les services Web proposés par les sociétés. En quelque sorte, UDDI propose d'industrialiser les services Web en automatisant aussi bien leur référencement que toute la procédure de recherche et de découverte de ces services.

1.2.3.3 SOAP (Simple Object Access Protocol)

Définit la structure des messages échangés par les applications via Internet. Cette définition repose également sur le langage XML.

1.2.3.4 REST (Representational State Transfer)

« REST n'est pas un protocole ou un format, c'est un style d'architecture, c'est le style architectural original du Web.

Ce style architectural n'est pas limité à la réalisation d'application pour un utilisateur humain. Il est de plus en plus utilisé pour la réalisation d'architectures orientées services utilisant des services Web destinés à la communication entre machines. REST se pose à la plupart des cas d'utilisation de SOAP, alternative censée être plus simple à mettre en œuvre. Les systèmes qui

suivent les principes REST de Fielding sont souvent appelés RESTful. » (Ngolo, Palma et al. 2009)

1.2.4 Fonctionnement

La Figure 1.3 présente les grandes phases de la mise en œuvre d'un service Web (Daniel 2003). En préambule, le fournisseur réalise un service Web ainsi que son interface au format WSDL.

Publication : le fournisseur publie (enregistre) son service Web auprès d'un distributeur (sur un annuaire UDDI). Cette opération se fait en envoyant directement à l'annuaire un message UDDI (encapsulé dans une enveloppe SOAP) via un protocole de transport. Les informations fournies regroupent la localisation du service, la méthode d'invocation (et les paramètres associés) ainsi que le format de réponse. Toutes ces informations seront par la suite formalisées à l'aide de WSDL.

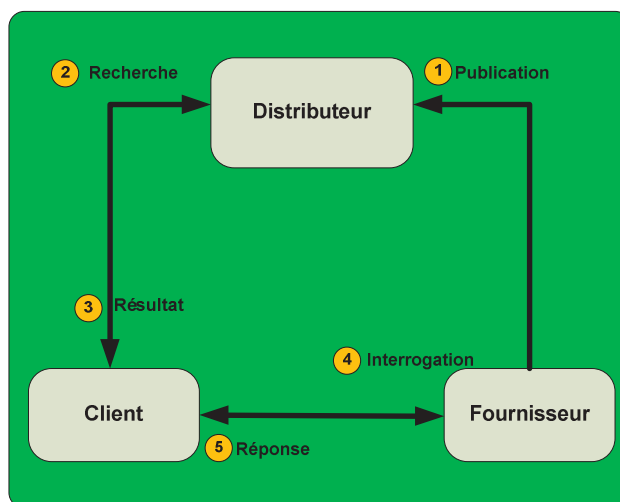


Figure 1.3 Principe général de fonctionnement.

Tirée de : http://www.softteam.fr/technologies_web_services.php

Recherche : le client effectue une recherche de service Web auprès du distributeur. Cette recherche se fait en envoyant un message UDDI (requête) encapsulé dans une enveloppe SOAP via un protocole de transport.

Résultat : le client reçoit en retour (via un protocole de transport) une réponse de l'annuaire. Cette réponse est un message WSDL encapsulé dans une enveloppe SOAP.

Interrogation : à l'aide de la réponse reçue, le client va pouvoir accéder directement au service Web chez le fournisseur. La demande de service s'effectue à l'aide d'un message SOAP via un protocole de transport.

Réponse : le client reçoit une réponse (via un protocole de transport) du service Web sous la forme d'un message SOAP (Englander 2003). Il ne reste alors qu'à exploiter cette réponse.

1.2.5 Point fort des services Web

1.2.5.1 Universalité

Les services Web visent à résoudre un problème qui a fait couler beaucoup d'encre et favorisé la création de nombreuses initiatives: l'interopérabilité entre les applications disparates. Ces dernières années, les entreprises se sont en effet rendues compte que la principale source de coûts dans le développement d'une nouvelle application est son intégration à l'existant. Les services Web sont l'approche la plus simple qui ait été proposée pour résoudre ce problème majeur. Par son mécanisme simpliste – messages XML transportés par HTTP – les services Web réussissent là où d'autres initiatives beaucoup plus complexes, telles que Corba, ont échoué.

Ainsi, que les applications tournent sous Windows ou Unix, qu'elles soient codées en Java, C++, VB ou Fortran, qu'elles soient déployées sur un serveur ou non, qu'elles soient derrière un firewall ou pas, elles sont capables de communiquer à partir du moment où elles comprennent le XML.

Un autre point fort des services Web est que les spécifications sont supportées par les acteurs forts du marché : IBM, Microsoft, Sun et Bea, ainsi que par un organisme de normalisation, le W3C¹ (World Wide Web Consortium).

1.2.5.2 Simplicité d'utilisation

Un autre facteur de réussite des services Web est leur simplicité et leur utilisation des standards acceptés par tous : HTTP et XML. La mise en place d'une communication inter-applicative simple ne doit pas prendre plus de deux jours de développement en utilisant cette technologie (Daniel 2003).

1.2.5.3 Couplage lâche des applications : simplification des contraintes d'urbanisation

Le troisième atout majeur des services Web est la possibilité de coupler les applications de manière beaucoup plus souple qu'avec Corba (Fay-Wolfe, DiPippo et al.), RMI (Ahuja and Quintao 2000) ou DCOM (Roy and Ewald 1997). En effet, ces RPC permettent de mettre en place des collaborations entre objets distants. Ainsi, pour mettre en place une collaboration entre applications, il est nécessaire de faire collaborer les objets qui les composent:

¹ Le W3C est un groupement d'entreprises reconnues dans le domaine informatique, qui a comme principal objectif de développer et de promouvoir des technologies propre à internet

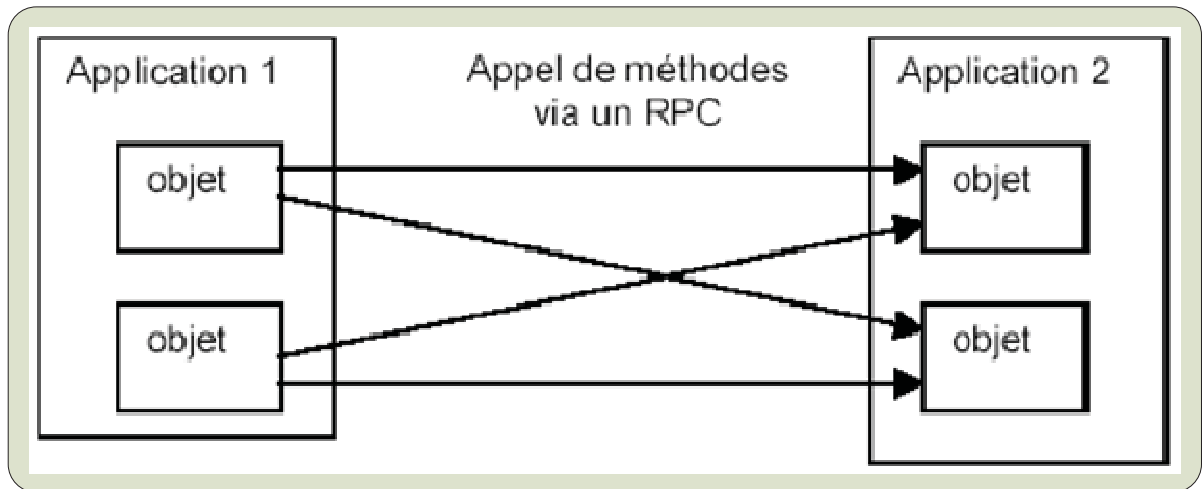


Figure 1.4 Couplage des applications.
Tirée de Daniel (2003)

Or ce mécanisme entraîne un couplage fort pour les applications qui collaborent. Il est souvent nécessaire que la structure des données soit la même des deux côtés de la connexion, etc. Les services Web quant à eux, introduisent la notion de « service » : une application offre des fonctionnalités à d'autres applications. Ainsi, nous n'avons plus à se soucier de la structure interne de l'application et des objets qui la composent. Les appels de méthodes au niveau objet sont la responsabilité du service, une sorte de « proxy » entre l'application cliente et les objets de l'application distante.

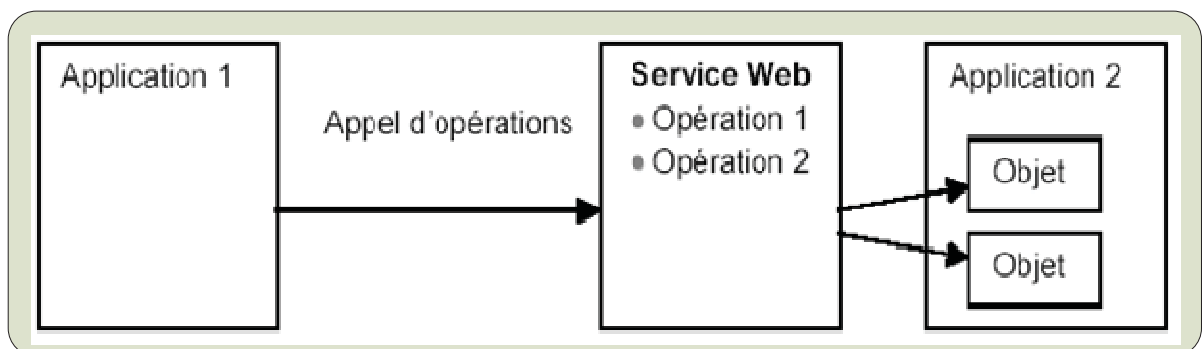


Figure 1.5 Offre de fonctionnalité via les services Web.

1.2.6 Inconvénients

- Les services Web souffrent de performances faibles comparées à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM;
- Par l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place au travers des pare-feu;
- La complexité de services Web (difficulté de déployer le services Web sur les appareils mobiles : Cellulaire, PDA etc.)

1.2.7 Conclusion

Dans cette partie nous avons effectué une brève analyse des services Web: les standards, le fonctionnement et enfin les points forts et les inconvénients de ces services.

Les services Web permettent surtout de standardiser l'accès aux applications, indépendamment du système d'exploitation, de l'infrastructure (serveur...) et du langage de programmation.

Les briques de base de services Web sont donc :

- WSDL : descriptions de services Web;
- SOAP : appel de services a distance par échange de message XML;
- UDDI : pour le référencement des Services.

Mais en ce qui concerne la syntaxe d'échange de données, c'est à ce niveau qu'intervient une technologie fondamentale pour les services Web : XML.

Dans la section suivante la technologie XML sera plus précisément étudiée.

1.3 XML (Extensible Markup language)

1.3.1 Introduction

L'évolution technologique ainsi que l'apparition de nouvelles formes de données ont mené à une transformation nécessaire des capacités et des manières d'utilisation de l'informatique et des réseaux.

En effet, ces dernières années, le nombre d'utilisateurs de l'ordinateur, des applications multimédia, des applications distribuées, etc. a augmenté clairement. Nous assistons de plus en plus à « la volonté de construire des ordinateurs toujours, plus petits, plus puissants et moins chers sans oublier la grande explosion de l'Internet » ((Wyke, Rehman et al. 2002) p.1). Ceci génère évidemment de grandes quantités de données. Ce qui a engendré un besoin croissant en matière de stockage, de traitement et d'analyse des données.

Pour résoudre ce problème, plusieurs outils ont été créés. La Technologie XML s'avère la mieux placée pour offrir ses solutions.

En effet, XML (W3Cp 2010), acronyme de « Extensible Markup language », est une façon de représenter des données structurées. Son objectif principal, lors de sa genèse par le W3C, est de créer une version de SGML (Standard Generalised Markup language) qui serait largement utilisée aussi bien sur Internet que sur HTML (Hyper Text Markup Language).

XML a été créé pour palier la faiblesse de ces deux technologies concurrentes. Le W3C a donc tenté de réunir la puissance de SGML et la simplicité de HTML pour créer un langage permettant l'échange de données structurées via Internet, c'est-à-dire un langage simple à utiliser.

Depuis sa création, il devient de plus en plus populaire. Il est perçu comme étant un outil puissant de manipulation, de sauvegarde et d'organisation des données.

Pour définir XML, nous disons que c'est un langage de description de données, une norme de structuration des données. Certains auteurs ont défini cette technologie comme étant « un véhicule pour les informations, qui livre sur votre ordinateur des données exploitables et propose un format de données universel » (Wyke, Rehman et al. 2002), p.7) ou encore « XML est un métalangage de description des langages à base de balises qui offrent une grande facilité dans la définition des balises et des relations structurelles qui existent entre elles » ((Wyke, Rehman et al. 2002), p.8).

Pour éclaircir et préciser d'avantage ses définitions, nous analysons le métalangage et la structuration des données.

Concernant le métalangage, XML permet de créer plusieurs auteurs. C'est précisément son avantage, puisque chacun peut créer son propre langage à partir des règles de syntaxe dictées pour cette norme. C'est-à-dire qu'il se repose sur un ensemble de règles syntaxiques pour une présentation structurée de l'information.

1.3.2 Structure

XML se présente comme HTML au niveau de la structuration, mais ses balises ne sont pas prédéfinies. La structure de code XML se présente sous la forme représentée à la Figure 1.6.

```

<? Xml Version='1.0' encoding= "UTF-8"?>
  <balise racine nom attribut= 'valeur'>
    Autres balises enfants et des contenus
  </ balise racine >

```

Figure 1.6 Structure de code XML.

Dans XML un élément (ou balise) doit contenir tous les autres. On appelle cet élément « élément racine ». Il ya deux sortes de balises XML :

- 1) Balise par paire <balise> contenu </balise>
- 2) Balise unique <balise/>

Nous remarquons dans le code, qu'il est possible de définir des attributs explicites à l'intérieur d'une balise.

Ils sont des paires *nom-valeur* qui interviennent dans les balises ouvertes, après le nom de l'élément. Il arrive qu'un élément (ou balise) ait besoin d'être complété par des informations additionnelles. Les attributs entrent dans l'une des trois catégories suivantes :

- 1) Un facteur distinctif du type élément;
- 2) Une information descriptive complémentaire;
- 3) Une information qui sera utilisée par l'application lors du traitement de données.

Donc, le document XML se présente sous forme de structure d'arbre où il y a des nœuds enfants et des nœuds parents. Chaque nœud possède un type précis. Cette structuration permet d'exprimer et de comprendre facilement des données complexes.

1.3.3 Domaine d'application

XML est une technologie très puissante capable de résoudre des problèmes complexes dans différents domaines. Cette technologie apporte un ensemble de solutions originales et performantes dans les domaines d'applications suivants :

- Le marketing one-to-one et la personnalisation de la relation client à travers le canal Internet;
- La gestion du contenu et les portails d'information d'entreprise ou EIP (Entreprise Information Portals), c'est-à-dire la production, l'archivage et la diffusion de données et d'informations;
- L'intégration d'application hétérogène ou EAI (Entreprise Application Intégration);
- Le commerce électronique B2B (business to business) et l'EDI (échange de données informatisées).

1.3.4 Conclusion

XML est une technologie très puissante ayant beaucoup d'avantages. Elle exploite un ensemble d'outils importants et « sans cesse croissant ». Elle est « libre de droits », ce qui permet la construction gratuite de logiciels, extensibles et portables. Elle forme des champs imbriqués hiérarchiquement. Ceci facilite la lecture des fichiers XML, la création et la maintenance des sites Web.

Le mérite du document XML réside dans le fichier texte. Donc l'utilisation d'un éditeur texte est suffisante pour la visualisation ou l'édition. Cependant, il n'est pas conçu pour être lu par l'utilisateur mais pour être analysé et manipulé automatiquement. D'où, le rôle d'analyseur (*parser*) qui devient important pour analyser le document XML et vérifier s'il est bien formé.

Dans la partie qui suit, nous présentons les types et les rôles des analyseurs XML.

1.4 Analyseur (*Parser*)

1.4.1 Introduction

La manipulation des fichiers XML est une tâche courante dans la phase de développement des applications. Cette manipulation s'effectue à travers des analyseurs.

Un analyseur XML est une bibliothèque de fonctions permettant la manipulation d'un document XML (Wang, Lia et al. 2007). Il est caractérisé par l'efficacité et la rapidité qui sont les critères essentiels d'un analyseur.

Il existe différents analyseurs implémentés dans différents langages. Les types les plus répandus sont :

- 1) Ceux qui utilisent un arbre pour représenter et exploiter le document;
- 2) Ceux qui utilisent les événements.

Présentement, il existe deux normes importantes de l'API (Application Programming Interface) : DOM et SAX. Ces dernières sont implémentées avec le langage JAVA (Bates and Sierra 2003). Elles permettent à l'utilisateur de manipuler facilement le document XML.

1.4.2 SAX (Simple API for XML)

C'est un analyseur rapide et ne nécessite pas une grande consommation de mémoire. Cette approche est basée sur un accès sérialisé et événementiel d'un document XML (Figure 1.7).

Le code SAX est souvent complexe. En effet, puisque ce analyseur n'utilise pas un arbre pour exploiter le document, l'utilisateur est confronté à une difficulté dans la manipulation, la sérialisation et la traversée du document XML.

L'analyseur lit le contenu du document XML et produit des événements en fonction des éléments qu'il rencontre.

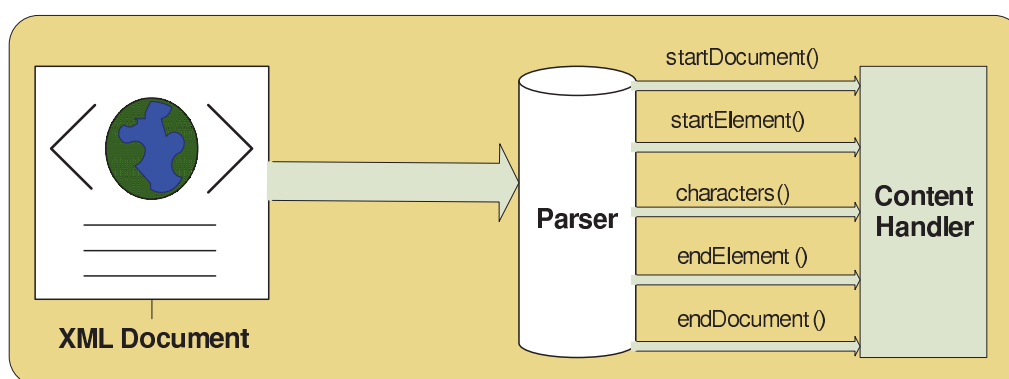


Figure 1.7 SAX analyseur XML.

Tirée de <http://www.isr.umd.edu/~austin/ence489c.d/xml.html>

Il existe plusieurs catégories d'évènements rencontrés connus par l'analyseur. Par exemple :

- début de document
- fin de document
- début de balise
- fin de balise
- donnée

1.4.3 DOM (Document Object Model)

C'est un système large (Wang, Li et al. 2007), très complexe et indépendant. Il utilise le modèle objet conçu pour soutenir n'importe quelles données XML (Figure 1.8).

Il crée des objets qui contiennent l'arbre avec les différents tags. Ces objets fournissent des méthodes permettant soit de parcourir l'arbre, soit de modifier ses contenus.

DOM fonctionne en deux étapes : la première étape consiste à charger le document XML (c'est ainsi que DOM consomme de la mémoire). La deuxième étape consiste à effectuer différentes opérations sur le document.

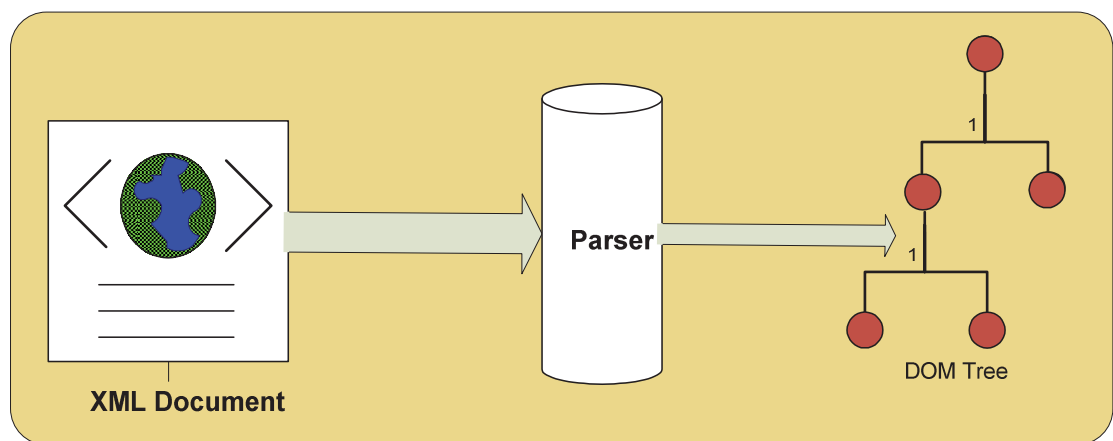


Figure 1.8 DOM analyseur XML.

Tirée de <http://www.isr.umd.edu/~austin/ence489c.d/xml.html>

A titre d'exemple, supposons que nous avons un fichier XML (Figure 1.9) :

```

<combine>
  <speech startTime= '500' endTime= '1000'>
    <action>move</action>
    <object> cube</objet>
    <aim>put it in position</aim>
  </speech>
  <touch startTimet= '1010' endTime= '1015'>
    <position>
      <X> a </X>
      <Y>b</Y>
    </position>
  </touch>
</combine>
  
```

Figure 1.9 Fichier XML.

L'arbre DOM obtenu après le processus d'analyse est présenté dans la Figure 1.10.

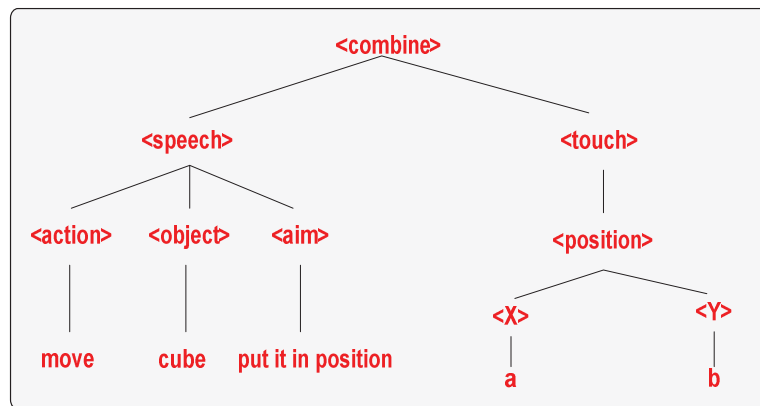


Figure 1.10 Exemple d'arbre DOM.

1.4.4 Conclusion

L'analyseur SAX est un analyseur qui traverse le fichier XML de haut en bas. Il analyse (*parse*) le fichier XML nœud par nœud et ne le stock pas dans la mémoire. Il n'est pas possible de supprimer ou d'insérer des nœuds.

DOM est un analyseur qui traverse le fichier XML dans n'importe quelle direction, Il occupe plus de mémoire. Il est possible d'insérer ou de supprimer des nœuds.

Le choix de l'un ou l'autre de ces deux analyseurs dépend de notre utilisation. Le Tableau 1.1 résume les caractéristiques de chaque analyseur.

Tableau 1.1. Comparaison entre SAX et DOM

	Avantage	Désavantage
SAX	-Rapide. -Analyse seulement les données essentielles. -Utilisation faible de la mémoire.	-Analyse séquentielle des documents. -Programmation difficile.
DOM	-Parcours libre de l'arbre. -Modification facile du contenu de l'arbre.	-Consommation de la mémoire. -Traitement de tout le document avant l'exploitation.

1.5 Les standards W3C XML relatives aux différentes modalités existantes :

En 2002, un groupe spécifique aux interactions multimodales a été créé par W3C (W3Cj 2010) dans le but de développer des langages basés sur XML permettant d'intégrer les différentes modalités existantes.

Et depuis, plusieurs langages ont été implémentés concernant l'interaction visuelle, vocale, gestuelle et multimodale.

1.5.1 Les langages pour les interactions visuelles

Dans cette section nous présentons le langage MPML (Multimodal Presentation Markup Language).

1.5.1.1 MPML (Multimodal Presentation Markup Language)

MPML (Tsutsui, Saeyor et al. 2000) est un langage à base de balises, conforme à XML. Il est conçu et mis au point pour faciliter la présentation multimodale. Il est basé sur le langage SMIL (Hoschka 2002) (Synchronized Multimedia Integration Language) qui supporte les fonctions de contrôle verbal et les agissements d'un agent intelligent (comme les agents Microsoft Office).

MPML permet aux utilisateurs d'écrire facilement des présentations multimodales attractives.

1.5.2 Les langages pour les interactions vocales

Étant donné le grand intérêt des applications à commande vocale, plusieurs langages intégrant les interactions vocales ont été développés. Nous présentons dans ce qui suit quelques uns.

1.5.2.1 VoiceXML

Le langage VoiceXML (Hocek and Cuddihy 2003; W3Ch 2010) (Voice eXtensible Markup Language ou langage de balisage extensible vocal) est l'une des applications qui relève du W3C. «Il trouve son origine en 1995 comme langage de création de dialogues fondé sur XML» (Hocek and Cuddihy 2003). Son objectif est de simplifier le processus de développement des applications de reconnaissance vocale et de rendre le contenu Internet et les informations accessibles via la voix et le téléphone.

Il permet d'utiliser la parole afin de réaliser des sélections ou d'envoyer des informations sans utiliser le clavier. VoiceXML contrôle les interactions entre une application et un utilisateur.

1.5.2.2 PLS (Pronunciation Lexicon Specification)

La recommandation du W3C concerne les domaines de la reconnaissance et de la synthèse vocale. Elle s'intéresse plus spécifiquement aux liens entre les mots, la façon de les écrire et celle de les prononcer.

Son objectif est « d'améliorer les performances des moteurs de reconnaissance et de synthèse vocale, en permettant à travers un langage standard de type XML, de spécifier un référentiel commun de prononciations. » (W3Ck 2010).

Le langage permet la prononciation (pour un mot ou une phrase) d'être spécifiée en utilisant un alphabet de prononciation standard ou si nécessaire en utilisant des alphabets spécifiques au fournisseur. Les prononciations sont regroupées dans un document PLS référencé à partir d'autres langages de balisage (W3Ck 2010) comme Speech Recognition Grammar Specification (SRGS) et the Speech Synthesis Markup Language (SSML).

1.5.3 Les langages multimodaux :

1.5.3.1 EMMA (Extended MultiModal Annotation markup language)

EMMA (Desmet, Balthazor et al. 2005; Johnston 2009; W3C 2010) est un langage générique de balise pour l'annotation de données générées par les médias. Il fait partie des normes de W3C pour les interactions multimodales.

Son objectif général est la représentation automatique des informations extraites des données entrées par l'utilisateur. Il représente les informations récupérées en entrée (les gestes, la parole...) sémantiquement pour les intégrer dans une application multimodale.

C'est un rassemblement d'informations multimédia, multimodales, multiplateformes.

1.5.3.2 X + V (XHTML plus Voice)

Le X+V est un langage développé par IBM et proposé au sein du W3C. Il est une combinaison de XHTML + voix (W3Cn 2010) destiné aux clients Web qui prennent en charge les interactions visuelle et vocale. Ce langage utilise une grammaire destinée à décrire les interactions vocales personne/machine.

1.5.4 Les langages pour les interactions gestuelles

1.5.4.1 InkML (Ink Markup Language Tags)

InkML (W3Cm 2010) (Ink Markup Language Tags) est un langage conçu par W3C pour permettre la présentation de l'écriture à main levée.

Il offre la possibilité d'échanger, entre différents périphériques et applications, les données entrées à l'aide d'un stylet électronique.

1.5.5 Conclusion

Nous pouvons dire que le W3C concentre ses efforts de standardisation dans le cadre des interactions multimodales, principalement sur le Web. Le W3C a engagé plusieurs groupes dans le but de développer plusieurs outils basés sur XML capables de gérer les différentes modalités existantes. Ce qui permet de faciliter la manipulation et le traitement des données reçues de différentes modalités.

De récentes recherches, qui décrivent les composantes technologiques issues des travaux de conception, ont permis la création de langages multimodaux basés sur XML. Cela permet d'interagir sur différentes modalités à partir d'une seule spécification (recommandations).

Le Tableau 1.2 résume les langages que nous avons cités.

Tableau 1.2 Récapitulatif des différents langages

Langages	Description
InkML	Représentation de l'écriture à main levée
EMMA	Langage de spécification multimodale (voix, touch, gestes...)
X+V	Langage de spécification multimodale (voix et visuel)
VoiceXML	Représente le dialogue et la reconnaissance
MPML	Supporte des fonctions de contrôle verbales

1.6 Application multimodales

1.6.1 Introduction

De nos jours, la technologie permet de produire des systèmes multimodaux distribués et entièrement contrôlés par les humains.

Ces systèmes sont munis d'interfaces multimodales permettant une interaction plus naturelle et plus efficace. L'utilisateur profite de ses modalités naturelles (ouïe, parole, geste, etc.) pour communiquer ou échanger des informations avec des applications.

Cette communication s'effectue à travers des applications distantes (ou application service Web) en utilisant différents outils tels que les appareils mobiles en général, les ordinateurs etc.

L'utilisation d'applications multimodales dans le service Web, avec l'intégration des modalités naturelles est une solution efficace aux personnes incapables d'utiliser un clavier, ou une souris, aux personnes ayant une déficience visuelle, aux utilisateurs « itinérants » équipés d'un téléphone sans fil/appareil mobile, aux personnes infirmes etc. Elle permet également d'offrir un meilleur confort d'utilisation pour les personnes ordinaires.

Dans ce qui suit, nous présentons des exemples d'applications multimodales utilisant les services Web et d'autres qui peuvent être implémentées dans le service Web.

1.6.2 Applications multimodales utilisant le service Web

L'intégration des applications multimodales dans les services Web est devenue de plus en plus fréquente grâce à l'emploi d'architectures distribuées sur le Web et l'utilisation de la communication naturelle. Ces dernières ont amélioré d'une façon remarquable l'accessibilité des personnes aux services.

1.6.2.1 Les événements catastrophiques

Dans (Caschera, D'Andrea et al. 2009), les auteurs nous ont présenté un système multimodal efficace en cas de catastrophe basé sur le service Web. Ils ont traité l'exemple de tremblement de terre. Les auteurs ont étalé trois scénarios pour expliquer le fonctionnement du système :

1.6.2.2 Scenario 1

Dans ce scénario, les auteurs expliquent comment l'utilisateur utilise son appareil mobile pour s'enregistrer à un réseau social « RISCOS » afin de réaliser l'évacuation et la sécurité en cas d'une catastrophe naturelle (tremblement de terre).

En premier lieu, l'utilisateur peut chercher le réseau social à partir de son téléphone à travers la voix « Réseau Social RISCOS ». Après l'affichage du résultat de la recherche sur l'écran de son téléphone, l'utilisateur défille les résultats avec son doigt, affiche et sélectionne le réseau désiré. Puis il peut dire / sélectionner « Enregistrer RISCOS » pour l'enregistrement.

En cas de tremblement de terre et pour savoir le lieu sécuritaire le plus proche de son emplacement, l'utilisateur n'a qu'à utiliser son téléphone et dire « RISCOS ». Dans ce cas, le téléphone se connecte automatiquement à ce réseau.

Quand la connexion s'établit, la carte du chemin menant à l'emplacement sécuritaire sera affichée.

Ensuite, l'utilisateur peut dire / choisir « agrandir » ou « minimiser » afin de bien afficher la carte.

1.6.2.3 Scenario 2

L'institut de sismologie a besoin de collecter des informations pour faire des statistiques, il envoie des questionnaires-textes aux téléphones des utilisateurs. Ces derniers répondent par la voix.

1.6.2.4 Scenario 3

Après le 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 places sécuritaires les plus proches. En se dirigeant vers ces places, les utilisateurs peuvent rencontrer des cas d'urgence ou entendre des voix dans des bâtiments effondrés. À ce moment, ils peuvent envoyer des demandes d'assistance au point de contrôle en attachant des photos ou des vidéos pour clarifier la situation.

La simplicité de l'utilisation en intégrant plusieurs modalités et l'accès instantané aux informations peut aider à sauver des vies. Ce système représente une façon très efficace de fournir des services d'urgence dans des situations critiques.

Mais dans ce cas d'applications multimodale, les auteurs se limitent aux utilisateurs qui sont inscrits dans le réseau social « RISCOS », mais dans le cas d'urgence on doit transmettre les informations nécessaires à tous le monde qui sont dans le lieu de catastrophe pour sauver le maximum de vies. Donc l'ajout d'un module pour détecter les gens qui ne sont pas inscrits à « RISCOS » et qui sont équipés des appareils mobiles est nécessaire.

1.6.2.5 Réservation de voitures de location

Les auteurs ont défini un système multimodal regroupant un appareil mobile et un ordinateur pour la location des voitures (Honkala and Pohja 2006). Le système consiste à gérer les réservations, les modifications et les annulations de location de voitures en utilisant les modalités : visuelle, vocale et textuelle.

Voici un exemple d'utilisation de ce système :

Au départ, l'utilisateur effectue une réservation de voiture en utilisant le site Web de la compagnie de location. Puis, il utilise un planificateur de trajet pour avoir le trajet; à partir des transports en commun jusqu'à l'emplacement du commerçant. Cette étape est effectuée en utilisant la modalité visuelle.

Par la suite, en se dirigeant vers la société de location, si l'utilisateur veut par exemple changer le type de voiture, il peut utiliser en entrée sa voix, combinée avec une commande visuelle, pour se connecter par téléphone au service de location.

Dans le cas où le lieu de livraison de la voiture est à changer aussi, l'utilisateur peut se connecter vocalement sur le planificateur de trajet et changer l'adresse de destination.

L'utilisateur peut également envoyer des messages texte entrés par la voix à ses amis, les informant qu'il a loué la voiture.

1.6.2.6 Accès aux services

L'accès aux informations de façon rapide et facile est devenu de plus en plus primordial. À titre d'exemple, dans un centre commercial (aéroport par exemple) la nécessité d'avoir de l'information sur les services offerts est élevée.

Dans (Robert, Khaled et al. 2005) les auteurs présentent les notions de découverte et de l'invocation des services. Ce concept est illustré en utilisant l'exemple suivant :

Un voyageur arrive à l'aéroport. Il reçoit une notification dans son appareil mobile l'informant des différents services existants à l'aéroport. Le voyageur peut accepter un service en disant «j'accepte ». En suite, il peut naviguer dans les différents services existants en utilisant les différentes modalités : la voix et le clavier.

Ce système est efficace car il peut être utilisé dans différents endroits comme les centres commerciaux, les musées etc., pour informer les visiteurs ou les clients des différents services fournis à cet endroit.

Dans ce qui suit, nous présentons des exemples de systèmes multimodaux qui peuvent être implémentés comme un service Web.

1.6.3 Applications multimodales qui peuvent être implémentée avec service Web

1.6.3.1 Système de conception de salle de bain

Dans (Pfleger 2004), l'auteur présente un système fréquemment utilisé dans la construction d'une maison et la conception de la salle de bain. La Figure 1.11 montre l'interaction entre l'utilisateur et l'ordinateur.



Figure 1.11 Démonstration du système.
Tirée de Pflieger (2004)

« L'interface multimodale de ce système inclut à l'entrée la parole et le stylet utilisés de manière intuitive et intégrée. Le feedback est généré par la voix, le graphique et les expressions faciales de la tête parlante » (Pflieger, 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 1.12).

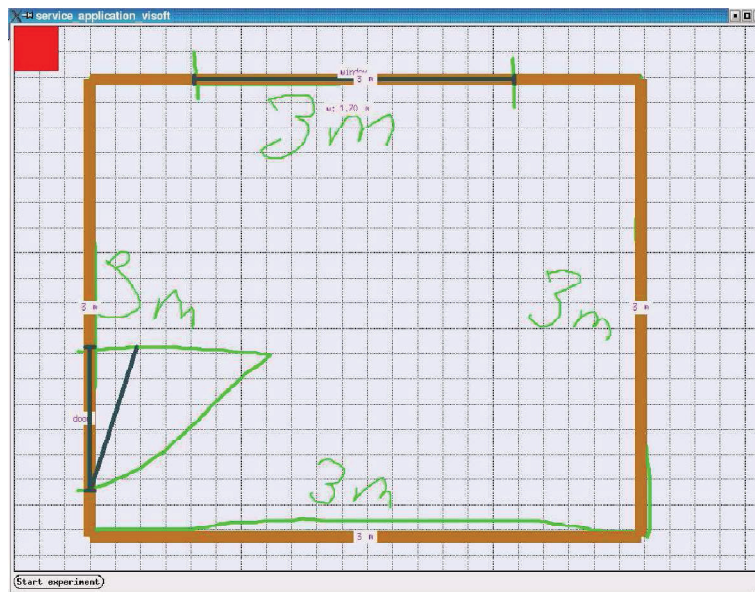


Figure 1.12 Exemple de capture d'écran de fonctionnement du système.
Tirée de Pflieger (2004)

1.6.3.2 Système multimodal utilise dans le domaine robotique

Dans (Giuliani and 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 1.13.

Le scénario que les auteurs ont présenté dans (Giuliani and Knoll 2008) est la construction et l'installation de panneaux de signalisation de chemin de fer. C'est un exemple où l'homme et le robot travaillent ensemble.



Figure 1.13 Exemple d'interaction multimodal entre un robot et un humain.

Tirée de Giuliani and Knoll (2008)

L'implémentation et le contrôle à distance d'un tel robot sera très bénéfique surtout dans des endroits où l'accès aux personnes représente un danger.

1.6.3.3 Enseignement assisté par ordinateurs

Dans (Wang, Zhang et al. 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 les aide à apprendre tout en jouant.

1.6.4 Conclusion

Pour conclure nous constatons que le nombre de modalités utilisées pour la plupart de ces exemples est limité à deux modalités.

L'étude des moteurs de fusions dans l'état de l'art montre clairement leurs limites pour l'intégration de plusieurs modalités dans un seul système. Tous les auteurs ont présenté une architecture dédiée juste à leurs applications.

L'utilisation de l'interaction multimodale pour la gestion des services et le partage d'informations est un point focal pour les situations d'urgence. En fait, l'utilisation de différentes modalités et/ou leurs combinaisons représente une manière efficace pour fournir des services dans des situations critiques.

L'intégration de plusieurs modalités dans un système est un moyen efficace pour simplifier l'utilisation des services aux utilisateurs.

CHAPITRE 2

FUSION MULTIMODALE DE L'INTERACTION : CONTRIBUTION

2.1 Architecture et approche

Dans cette partie, nous allons décrire l'approche proposée ainsi que les modules impliqués lors de la conception et l'implémentation de l'architecture de la fusion multimodale.

Notre architecture repose sur l'accès aux services Web en utilisant trois modules : « Parser », « Parameters Extractor » et « Fusion And Multimodality ». Ces modules communiquent ensemble pour échanger des informations. Ils sont chargés sur un ordinateur, un robot, ou tout appareil qui peut communiquer via Internet, réseau social, ... etc. L'architecture du système proposé est illustrée par la Figure 2.1.

Comme le montre la Figure 2.1, le système est composé des éléments suivants :

- *Parser* : ce module prend en entrée un fichier XML. Son rôle est d'extraire des informations du fichier XML qui sont importantes pour la fusion et indique en sortie la modalité convenable et ses paramètres associées.
- *Parameters Extractor* : la sortie du module « *Parser* » est l'entrée de ce module. Il extrait les paramètres de chaque modalité impliquée.
- *Fusion and Multimodality* : en se basant sur les paramètres de chaque modalité et en prenant en considération le temps (quand la modalité est survenue), ce module décide si la fusion est possible.

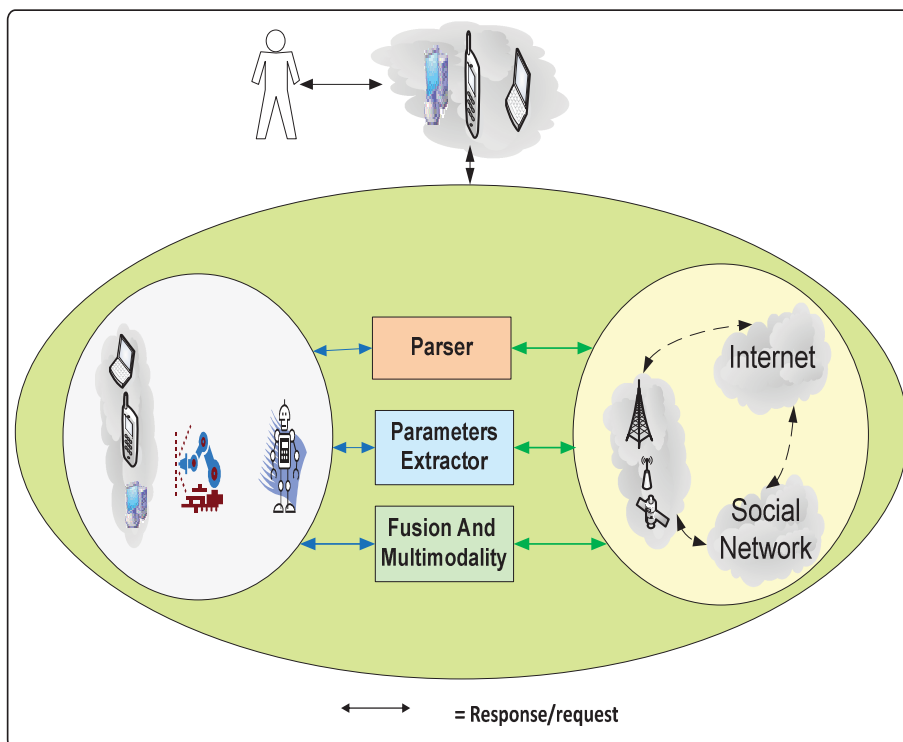


Figure 2.1 Architecture du système de fusion multimodale pour accéder aux services Web.

- *Internet/Réseau social* : c'est les réseaux par lequel l'utilisateur et la machine / ordinateur impliqué communiquent.
- *Appareil informatique / Robot / Téléphone* : c'est l'entité par laquelle l'utilisateur communique.

Considérons par exemple la commande « Remplacer ce fichier avec ce fichier » où l'utilisateur utilise la parole et un clic de souris pour indiquer « ce fichier » et un autre clic de souris pour indiquer l'autre fichier « ce fichier ». Dans ce cas, les modalités impliquées sont:

- La modalité d'entrée 1 : la voix et,
- La modalité d'entrée 2 : la souris.

Les processus impliqués dans la fusion de ces modalités sont les suivants (Figure 2.2) :

- *Reconnaissance (Recognition)* - Ce composant a pour rôle de convertir et d'identifier les activités des différentes modalités en des fichiers XML correspondants. Ce composant reçoit les informations concernant chaque modalité à partir des capteurs. Dans cet exemple, les capteurs sont un microphone et une souris.
- *Les modules « Parser », « Parameters Extractor » et « Multimodal and Fusion Module » ont les mêmes fonctionnalités que celles citées ci-dessus.*
- *Action* : il s'agit de la mesure correspondante à être entreprise après que la fusion ait été faite. La sortie peut être implémentée en utilisant les modalités de sortie de 1, 2, ..., n. Dans le même cas cité plus haut, la modalité de sortie est un simple écran. La confirmation de cette action peut être présentée également par un haut-parleur.
- *Contre-réaction (Feedback)* : en cas de conflit, l'utilisateur reçoit un feed-back du système pour corriger l'erreur. Par exemple, dans le cas cité, si « ce fichier » et « ce fichier » réfèrent à la même entité, l'utilisateur en est informé.

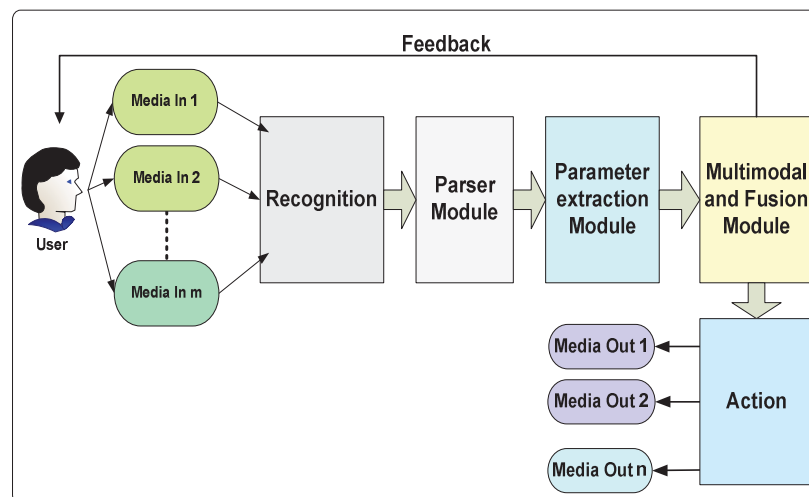


Figure 2.2 Framework de la fusion multimodale.

Tous ces modules peuvent être installés dans n'importe quel endroit du réseau. Les modules communiquent entre eux afin d'échanger des informations ou pour effectuer une tâche, par exemple donner des ordres à un robot pour exécuter une tâche quelconque.

2.2 Fusion multimodale

Ce module présente la partie cruciale de notre architecture. Son rôle est de combiner plusieurs modalités en une seule représentation sémantique qui donne un sens. Le type de la fusion utilisée c'est la fusion tardive (voir p. 18).

Ce module est constitué d'une base de données et d'un sous module fusion qui s'occupe de la fusion. La base de données contient toutes les informations concernant les modalités.

Supposons, par exemple, l'arrivée de la modalité A, avec ses paramètres (temps, ..., etc.) et une modalité B avec ses propres paramètres (temps, ..., etc.). L'agent de fusion produira alors une combinaison logique de A et B, donnant un résultat, que nous appelons C. La commande / événement C est ensuite envoyé à l'application ou à l'utilisateur pour la mise en œuvre.

En général, la règle de fusion est simple : si la modalité A (et ses paramètres) et la modalité B (et ses paramètres) sont présentes, alors C (et ses paramètres) est déduite (Figure 2.3).

La fusion multimodale peut être représentée par la relation $f: C = A + B$.

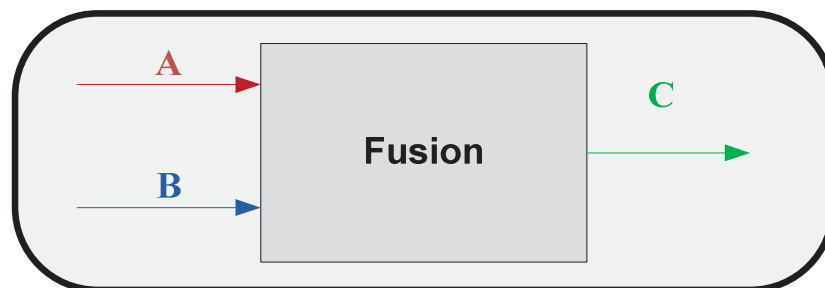


Figure 2.3 Fusion de deux modalités.

Les étapes du processus de fusion sont décrites par la Figure 2.4. Admettons que chaque fichier XML contient des données relatives aux modalités d'entrée, ainsi que leurs paramètres correspondants, les étapes prises en considération sont les suivantes:

- 1^{ère} étape : le système vérifie si le scénario est déjà dans la base de données ; c'est-à-dire si le scénario est déjà survenu auparavant.
Si oui, le système extrait le résultat (également enregistré dans la base de données) de la base de données. Ce résultat est envoyé vers les modalités de sorties. Aucun calcul de fusion n'est donc nécessaire. Sinon, l'étape 2 est exécutée.
- 2^{ème} étape : ceci indique que le scénario est nouveau pour le système. Le système vérifie si la sémantique des opérations des différentes modalités est équivalente.
Considérons par exemple un cas dans lequel l'utilisateur énonce la commande vocale: «*Ecrit 5* » et en utilisant le clavier, il tape également "5". Techniquement, les deux actions ont la même signification et sont donc sémantiquement équivalentes. Ce résultat signifie qu'il n'y a pas de fusion à faire. Les informations seront stockées dans la base de données de référence pour réutilisation si un tel événement se reproduit à l'avenir.
Le système exécute la requête. Aucun calcul de fusion n'est nécessaire. Ces commandes sont traitées comme unimodales.
- 3^{ème} étape : dans cette étape, le système teste si deux ou plusieurs commandes sont en conflit l'une avec l'autre. Par exemple, si l'utilisateur dit vocalement : "*Ecrit 5*" et en utilisant le stylet, par exemple, il écrit: "4". Dans ce cas, le système prend une décision de favoriser l'une de ses deux commandes ou d'envoyer un feedback à l'utilisateur.

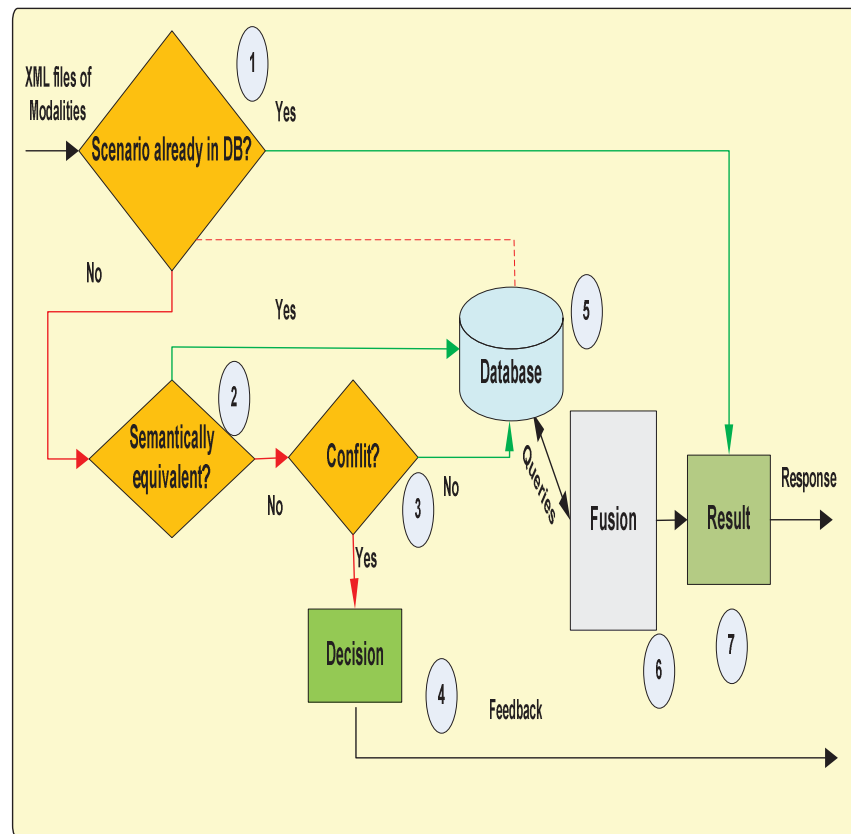


Figure 2.4 Le processus détaillé de la fusion.

- 4^{ème} étape : une commande est envoyée à l'utilisateur pour corriger un conflit détecté. Si l'utilisateur ne prend aucune mesure, le système continue de lui envoyer le feedback pour résoudre le problème. L'utilisateur doit décider laquelle des deux commandes (ou n commandes, en général) – l'une ou l'autre - doit être considérée par le système pour l'exécution.
- 5^{ème} étape : Si le scénario est complètement nouveau et il n'y a pas de données contradictoires dans la requête, toutes les données correspondantes seront enregistrées dans la base de données. Le processus de fusion peut alors démarrer.

- 6^{ème} étape : les requêtes sont envoyées à la base de données par l'agent de fusion pour extraire des informations concernant les modalités. La fusion est ensuite réalisée et le résultat est également stocké dans la base de données.
- 7^{ème} étape : le « résultat » désigne l'action désirée à être accomplie en utilisant les modalités impliquées.

Pour réaliser la fusion nous allons subdiviser le module en trois sous composantes comme le montre la Figure 2.5 :

- *Selector* : il interagit avec la base de données pour sélectionner les modalités désirés. Il extrait 1...n modalités en cas de besoin ;
- *Grammar* : vérifie les conditions grammaticales et toutes les permutations possibles entre les modalités concernées ;
- *Fusion* : c'est le module qui implémente la fusion.

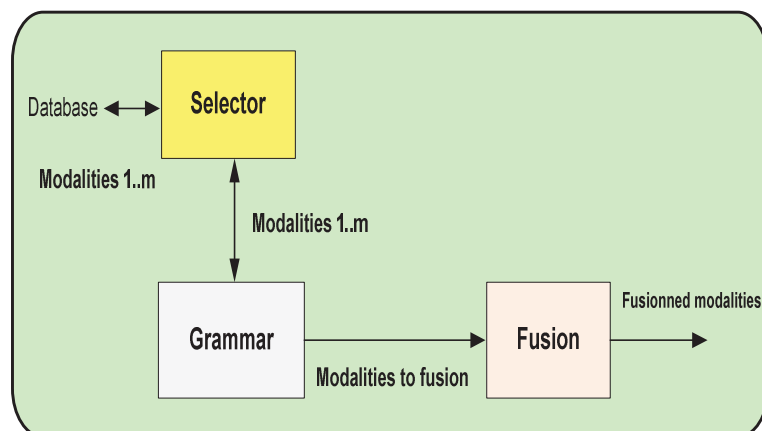


Figure 2.5 Agent de Fusion.

L'algorithme du processus de fusion est présenté dans la Figure 2.6. L'essence du temps est importante pour déterminer si les actions impliquant les modalités utilisées peuvent être fusionnées.

L'opération commence par l'accès à la base de données et la sélection des données associées à une modalité (étape 1). L'étape suivante consiste à vérifier les conditions grammaticales de la modalité associée (étape 2). Ici, l'analyse grammaticale pourrait déterminer si les mesures à prendre concernant la modalité pourraient être interprétées comme une action unimodale ou multimodale. Dans ce dernier cas, nous aurons besoin d'autres données complémentaires provenant d'autres modalités. Par exemple, la commande vocale « Mettez ça là-bas » implique que certaines données associées à « ça » et « là-bas » sont exigées.

Par contre, une commande vocale telle que « Dessine une boîte » ne nécessite pas des données complémentaires et peut donc être considérée comme une activité unimodale.

(Etape 3) Si l'action souhaitée nécessite des entrées multimodales, alors la fusion est réalisée. Mais si elle est unimodale, il n'y aura pas de fusion.

Dans les deux cas, les résultats sont stockés dans la base de données pour de futures références. Comme le montre le schéma, le processus de fusion est un processus continu, toujours en vigueur lorsque l'utilisateur est connecté au système.

Des erreurs pendant la vérification des conditions grammaticales peuvent également survenir. Par exemple, une commande vocale « pose là-bas » est un échec s'il n'y a pas une modalité complémentaire associée - telle que le toucher, le regard, le clic de souris, ..., etc. Si un tel cas se présente, le système recherche d'autres modalités qui relèvent du même intervalle de temps que celle précédemment analysée. Dans le cas où le système ne trouve aucune fusion possible, il rejette la commande ou envoie un feedback à l'utilisateur pour corriger l'erreur.

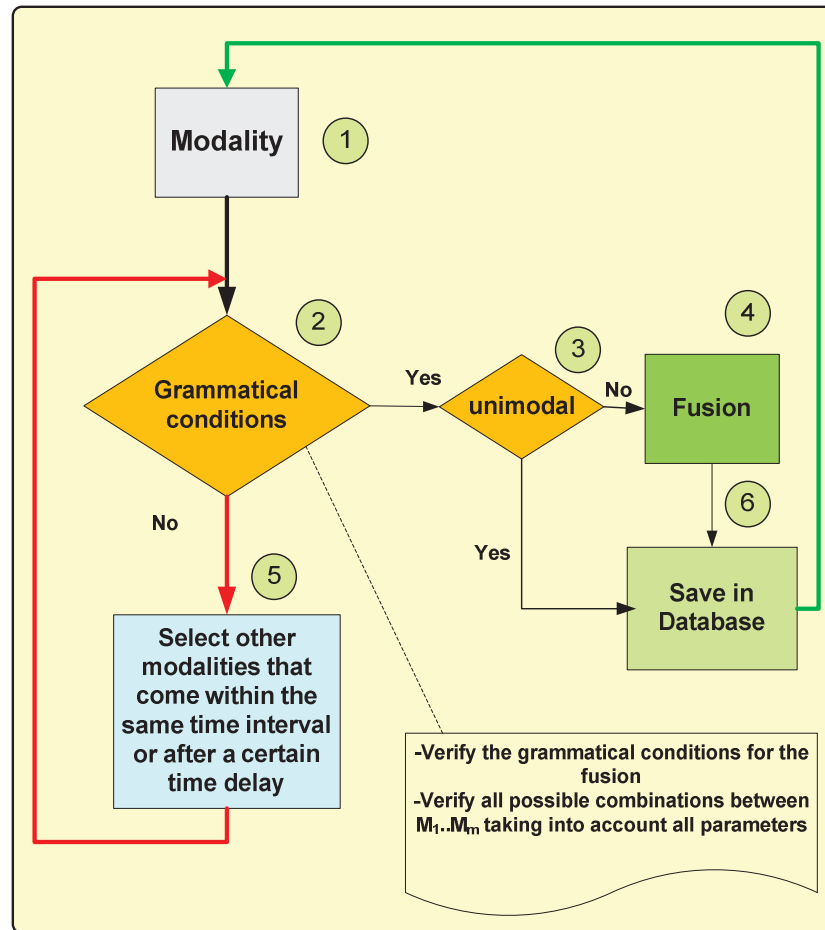


Figure 2.6 Algorithme de Fusion.

2.3 Élément temps dans le processus de fusion

Dans cette partie, nous allons illustrer différentes variations du cadre temporel qui nous permettent de décider si la fusion est possible. Pour tous les cas étudiés, nous utiliserons les notations suivantes :

- M_i = modalité i
- t_1M_i = temps d'arrivée de la modalité i
- t_2M_i = temps final de la modalité i
- $t_{\min}$ = le délai minimum accordé entre deux modalités pour les considérer comme multimodale

En référence avec la Figure 2.7, les cas suivants peuvent se présenter :

- Cas A : Modalité 1 commence en t_1M_1 et se termine en t_2M_1 . La Modalité 2 commence en t_1M_2 et se termine en t_2M_2 . Cependant, elles se produisent en des laps de temps différents et sont mutuellement exclusives. Comme elles sont indépendantes l'une de l'autre, chaque modalité est unimodale. Chacune est alors traitée séparément. Par conséquent, la fusion multimodale peut être représentée par les relations :
 $f_1: C_1 = M_1$ et $f_2: C_2 = M_2$.
- Cas B : Modalité 1 et Modalité 2 arrivent séparément mais le temps d'arrivée de ces deux modalités est inférieur à $t_{\min}$. Dans ce cas, un test est nécessaire pour déterminer si la fusion est possible. Si oui, la fonction résultante sera : $f: C = M_1 + M_2$
- Cas C : Les Modalités 1 et 2 arrivent dans le même laps de temps. Ici, la fusion est évidente et la fonction qui en résulte est $f: C = M_1 + M_2$.

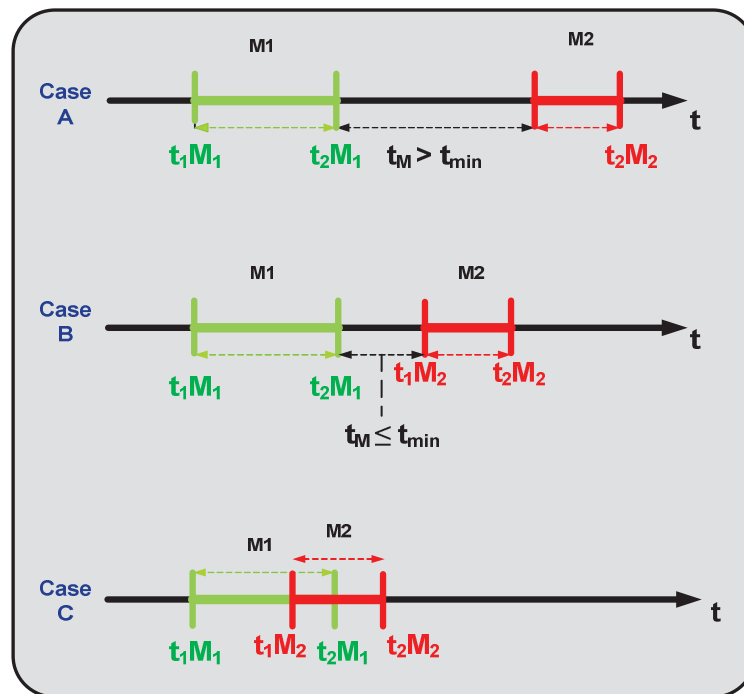


Figure 2.7 Le facteur temps impliqué dans deux modalités.

En général, le système peut faire la fusion en impliquant trois ou plusieurs modalités. Un exemple impliquant trois modalités est présenté à la Figure 2.8. Les événements suivants sont possibles :

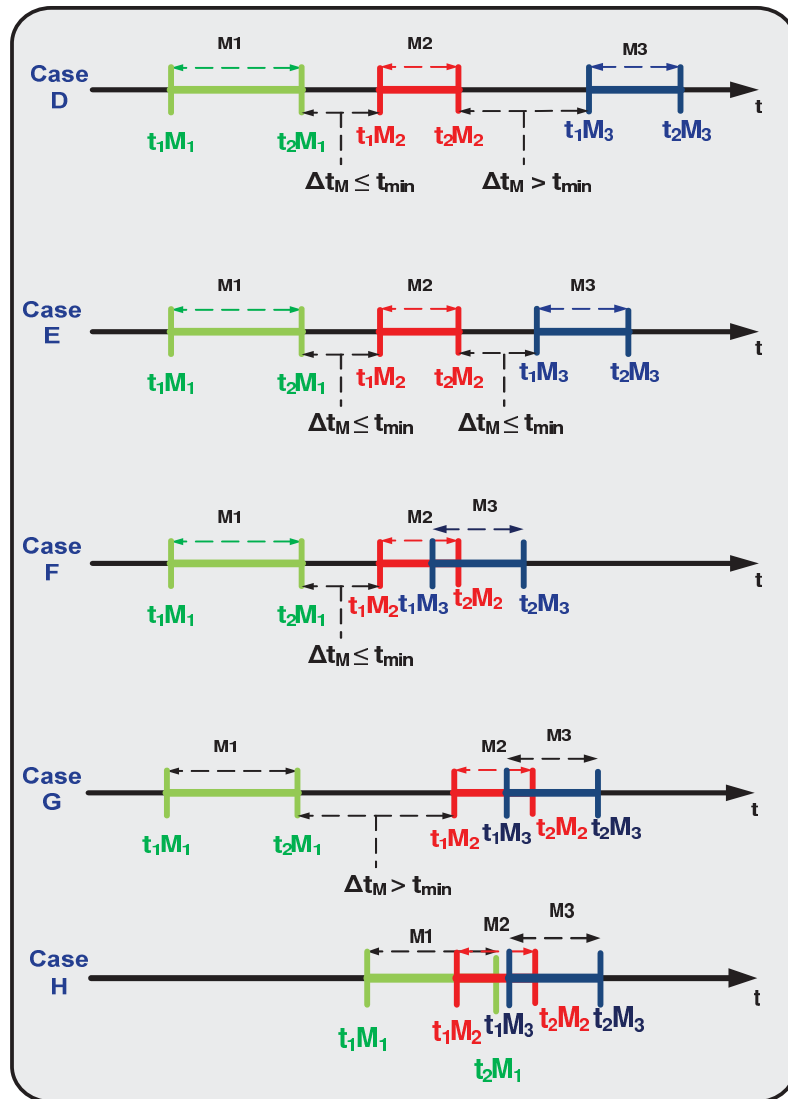


Figure 2.8 Le facteur temps impliqué dans trois modalités.

- Cas D: dans ce cas, les trois modalités se produisent à différents intervalles de temps et elles sont mutuellement exclusives. Comme elles sont indépendantes les unes des autres, chaque modalité est considérée unimodale et est traitée séparément. D'où :
 $f_1: C_1 = M_1$, $f_2: C_2 = M_2$ et $f_3: C_3 = M_3$.

- Cas E: les trois modalités surviennent séparément mais le temps de production de M_1 et M_2 est inférieur à $t_{\min}$. Un test est alors nécessaire pour déterminer si la fusion entre M_1 et M_2 est possible puis de vérifier si la fusion entre les trois modalités est possible. Dans l'affirmation, la fonction résultante serait :
f: $C = M_1 + M_2 + M_3$.
- Cas F : M_2 et M_3 surviennent dans le même intervalle de temps. M_1 survient avant les deux dernières. Le délai entre le temps de fin de la modalité 1 et le temps de début de la modalité 2 est inférieur à $t_{\min}$ ($t_1M_2 - t_2M_1 < t_{\min}$). Donc la fusion est évidente entre M_1 et M_2 et la fusion avec M_3 doit être vérifiée. La fonction résultante dans le cas affirmatif est :
f: $C = M_1 + M_2 + M_3$.
- Cas G : M_2 et M_3 surviennent dans le même intervalle de temps. M_1 survient avant les deux dernières. Le délai entre le temps de fin de la modalité 1 et le temps de début de la modalité 2 est supérieur à $t_{\min}$ ($t_1M_2 - t_2M_1 > t_{\min}$). La fusion est alors limitée à M_2 et M_3 . Et M_1 devrait être considérée comme une action unimodale. La fonction résultante est :
f: $C_1 = M_2 + M_3$, g: $C_2 = M_1$.
- Cas H: les trois modalités surviennent dans le même intervalle de temps, donnant une fonction : f: $C = M_1 + M_2 + M_3$.

2.4 Spécification et composants de fusion multimodale

Dans cette partie, nous présentons les différents composants impliqués dans le processus de fusion multimodale et nous donnons les spécifications de chaque composant à base de réseaux de Petri (RdP), cf. paragraphe 2.4.4. Nous avons utilisé l'outil de spécification formelle PIPE pour simuler les RdP afin d'élaborer les spécifications des composants du système. Nous avons également développé une application en Java pour valider l'approche proposée.

2.4.1 L'interface utilisateur

Comme pour tous les systèmes informatiques interactifs, notre système a une interface usager (Oviatt and Cohen 2000) avec laquelle l'utilisateur peut communiquer avec le système. Avec cette interface, l'utilisateur peut sélectionner les modalités qu'il souhaite utiliser selon ses besoins Figure 2.16 (B).

Pour faciliter l'utilisation, l'interface de la Figure 2.16 a été créée pour être la plus simple possible.

Le système reste en boucle jusqu'à ce qu'il détecte un événement impliquant une modalité (par exemple, est-ce qu'il y avait un clic de souris, une entrée vocale, ..., etc.). À ce moment, le système se connecte à la base de données pour vérifier si l'événement est valide. Un événement est invalide si, par exemple, l'utilisateur sélectionne deux événements en utilisant deux modalités en même temps alors que le système exige qu'un seul événement ne soit exécuté à la fois. Si l'événement impliquant la modalité est valide, un fichier XML est créé pour les modalités utilisées et leurs paramètres associés. Le fichier XML est transmis au module « Parser ». L'analyseur extrait ensuite les données à partir des fichiers XML et envoie le résultat obtenu au module de fusion multimodale. Le schéma de la Figure 2.9 illustre ce processus.

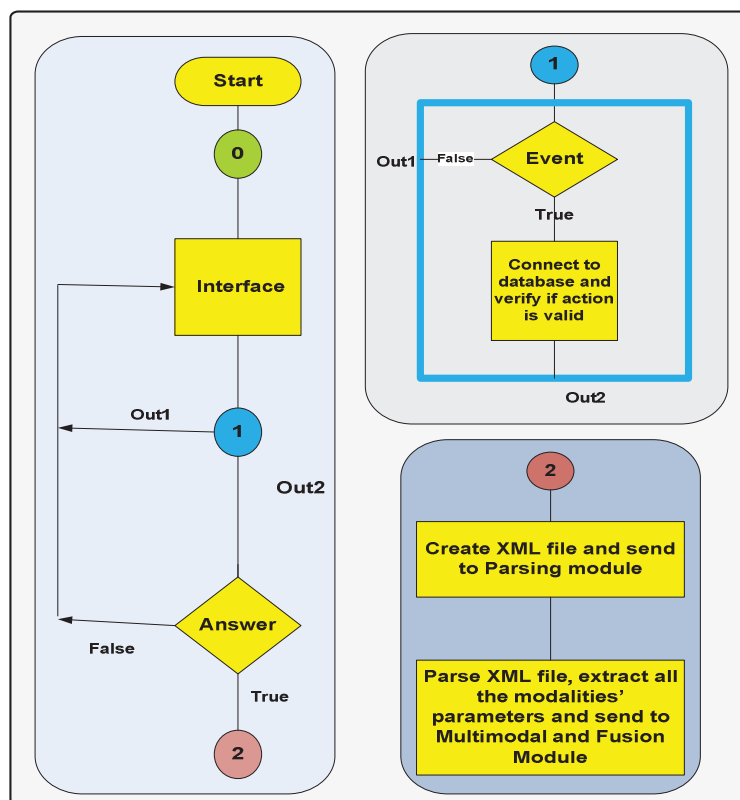


Figure 2.9 L'interface utilisateur du système.

2.4.2 L'analyseur

Ce module reçoit en entrée des fichiers XML contenant des données sur les modalités, généralement capturés au moyen de capteurs (par exemple, webcam, micro, écran tactile, ..., etc.). Ce module extrait de chaque fichier XML des données d'étiquette dont il a besoin pour la fusion. Ensuite, il crée un résultant XML contenant les modalités choisies et les paramètres correspondants de chacune d'elles. Un exemple détaillé est traité au paragraphe 2.4.4.

En conformité avec la norme du W3C sur les balises XML pour les applications multimodales, nous utilisons EMMA.

Par exemple, supposons qu'un module reçoit les fichiers XML suivant la notation EMMA (voir Figure 2.10). Les balises correspondantes EMMA et leurs significations sont les suivantes:

- **<emma:emma>** est le nœud racine, il contient la version EMMA et l'information concernant l'**espace de noms** (*namespace*) qui « désigne en informatique un lieu abstrait conçu pour accueillir des ensembles de termes appartenant à un même répertoire » (Wiki 2010) . Ce nœud est un conteneur pour un ou plusieurs des éléments suivants:
 - **<emma:interpretation>** est utilisé pour définir une interprétation des données d'entrée. Il contient un attribut spécifique;
 - **id** est une valeur unique pour chaque interprétation. Elle permet d'identifier l'interprétation dans le document EMMA.
 - **<emma:group>** est un conteneur général pour une ou plusieurs interprétations. Il peut être associé à des critères de groupement arbitraires. L'*emma:group* est un élément utilisé pour indiquer que les interprétations qu'il contient sont liées entre elles d'une certaine manière.
 - L'annotation *emma:model* peut faire référence à n'importe quel élément ou attribut dans le code ainsi que tous les éléments contenus dans EMMA (*emma:one-of*, *emma:group*, or *emma:sequence*).
 - L'annotation *emma:lang* est utilisée pour indiquer le langage humain utilisé pour annoter les données en entrée. Les valeurs de l'attribut d'*emma:lang* sont des indicatifs de langue. Par exemple, *emma: lang = "fr"* désigne le français alors que *emma :lang = "eng-US"* désigne l'anglais américain.
 - Le **score de confiance** dans EMMA est utilisé pour indiquer la qualité des données en entrée et c'est la valeur attribuée à *emma:confidence* dans l'espace de noms d'EMMA.
 - *emma:start* et *emma:end* sont des attributs indiquant les temps absolus du début et fin d'une donnée en entrée en millisecondes.

```

<emma:emma>
  <emma:group
    emma:medium="acoustic,tactile"
    emma:mode="voice,touch" emma:function="dialog">
      <emma:interpretation id="speech1"
        emma:confidence="0.9" emma:verbal="true"
        emma:start=" 2010-03-26T0:00:00.1 " emma:end=" 20010-03-26T0:00:03.1 "
        emma:medium="acoustic" emma:mode="voice"
        emma:confidence="0.8" emma:lang="en-US"
        emma:process="smm:type=asr&version=asr_eng2.4"
        emma:media-type="audio/amr; rate=8000">
        <emma:literal>Put this cube here
      </emma:literal>
    </emma:interpretation>
    <emma:interpretation id="touch1"
      emma:confidence="0.8"
      emma:medium="tactile" emma:mode="touch"
      emma:start=" 2010-03-26T0:00:01.6 "
      emma:end=" 2010-03-26T0:00:02.1 ">
      <x>0.253</x>
      <y>0.124</y>
    </emma:interpretation>
    <emma:interpretation id="touch2"
      emma:confidence="0.9"
      emma:medium="tactile" emma:mode="touch"
      emma:start=" 2010-03-26T0:00:03.5 "
      emma:end=" 2010-03-26T0:00:04.1 ">
      <x>3.768</x>
      <y>2.324</y>
    </emma:interpretation>
    <emma:group-info>temporal</emma:group-info>
  </emma:group>
</emma:emma>

```

**Figure 2.10 Modèle de fichier XML (EMMA)
contenant des modalités (parole et écran tactile).**

Le fichier XML combiné résultant de l'utilisation des modalités de la parole et de l'écran tactile, présentées dans la Figure 2.10, est semblable à celui illustré par la Figure 2.11 (à gauche). Ensuite, la fusion de ces deux modalités donne le résultat illustré par la Figure 2.11 (à droite). Le résultat de la fusion indique que le cube objet est déplacé au nouvel emplacement (a, b).

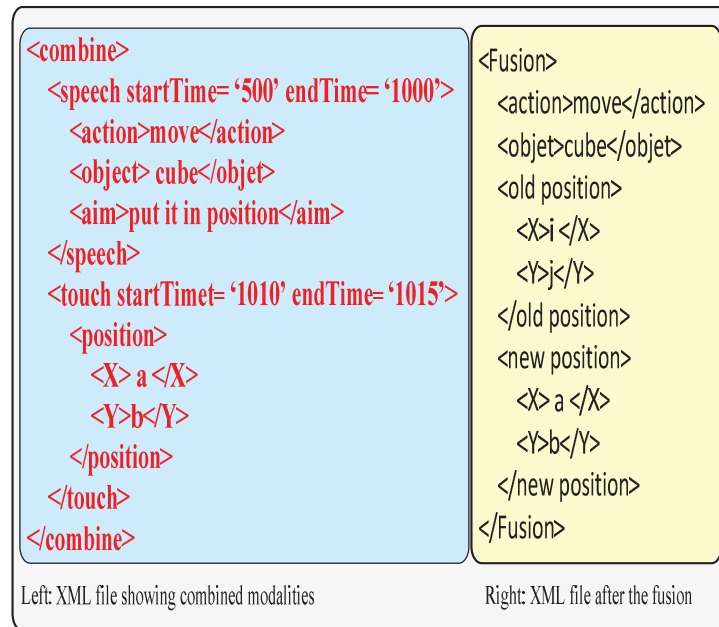


Figure 2.11 (gauche) le fichier XML combiné résultant et (droite) le fichier XML résultant après la fusion.

2.4.3 La base de données

La base de données stocke toutes les modalités définies par le système et leurs paramètres associés. En effet, Il existe de nombreuses bases de données disponibles, telles que PostgreSQL, MySQL, SQL Server, Oracle, Access, etc. Dans ce travail, la base de données utilisée est PostgreSQL (PostgreSQL 2010).

En utilisant le PostgreSQL, les paramètres, les valeurs et les entités de la base de données sont définies dynamiquement quand le module analyse le fichier XML.

Comme le montre la Figure 2.12, notre base de données se compose de sept tableaux.

Pour la suite, nous définissons les composantes de chaque tableau puis nous élaborons un exemple pour plus d'explication :

- **Modality** : ce tableau contient le nom des modalités, le temps du début d'une action impliquant la modalité et le moment où elle a fini;

- ***Modality_Added_Parameters*** : ce tableau contient tous les attributs de chaque modalité;
- ***Modality_Main_Parameters*** : ce tableau contient le nom de tous les paramètres et leurs valeurs respectives;
- ***Union_Modality_Main_Parameters*** : ce tableau fait le lien entre le nom de la modalité et ses paramètres;
- ***Fusion*** : ce tableau contient toutes les fusions qui ont été faite. Il nous permet de conserver les données fusionnées qui peuvent être utilisés plus tard comme référence;
- ***Fusion_Main_Parameters*** : ce tableau contient le nom des paramètres et leurs valeurs qui sont associées à la fusion multimodale;
- ***Union_Fusion_Main_Parameters*** : ce tableau sert de lien à la fusion multimodale qui vient d'être achevée, y compris ses paramètres correspondants.

Pour montrer quelles données sont stockées dans la base de données, citons un exemple en utilisant le fichier XML comme le montre la Figure 2.13.

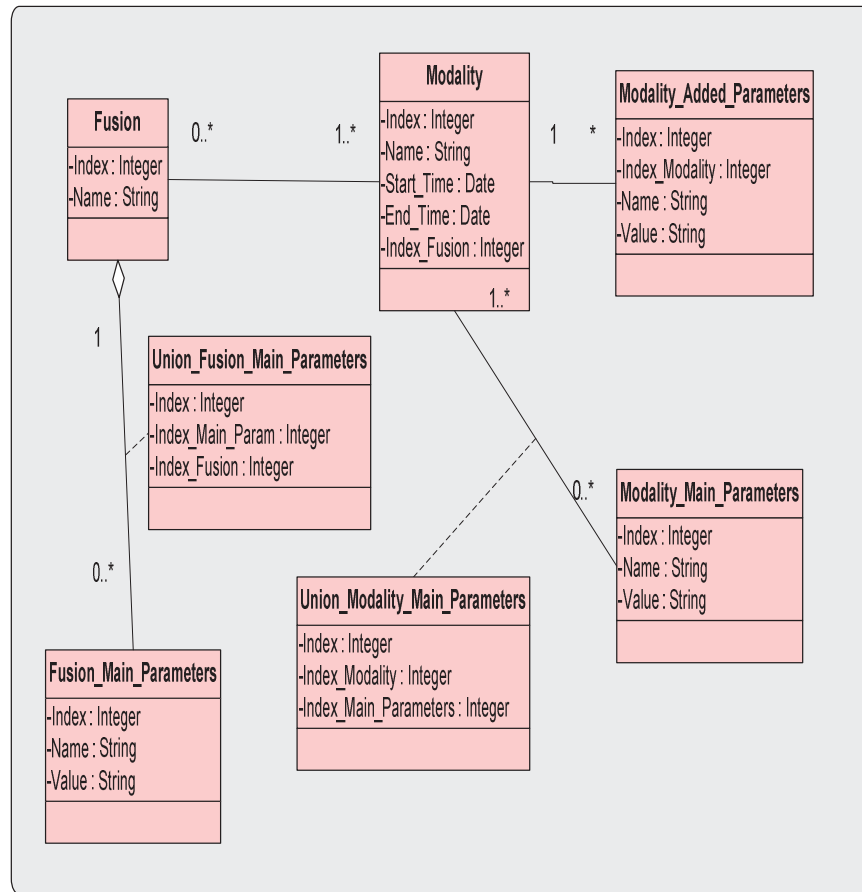


Figure 2.12 Diagramme de classe de la base de Données.

```

<combine>
  <speech startTime= '10:11:11' endTime = '10:12:13'>
    <action>move</action>
    <object>yellow box</object>
    <goal>put in POSITION</goal>
  </speech>
  <click startTime= '10:12:00' endTime= '10:13:01'>
    < positionX> a </ positionX>
    < positionY>b</ positionY>
  </click>
</combine>
  
```

Figure 2.13 Modèle d'un fichier XML.

Ce fichier XML contient deux modalités parole et click de souris. Avec la parole, l'utilisateur demande de déplacer la boîte jaune à une position qui a été définie par le click de souris.

Ainsi, étant donné le cas cité (Figure 2.13), le contenu du tableau de la Modalité est présenté au Tableau 2.1.

Tableau 2.1. Modèle d'un tableau de « MODALITY »

Index	Name	Start_Time	End_Time	Index_Fusion
1	Speech	10 :11 :11	10 :12 :13	1
2	Mouse click	10 :10 :11	10 :12 :13	1

Les composants de ce tableau et leurs significations sont :

- **Index** : c'est un « élément de redondance que l'on va spécifier pour permettre au Système de Gestion de Base de Données d'optimiser certaines requêtes » (Wiki 2010). Dans notre cas, il désigne l'index de chaque modalité. Cette colonne contient des nombres entiers qui s'incrémentent automatiquement lors de l'ajout de nouvelles modalités.
- **Name** : cette colonne contient le nom de toutes les modalités.
- **Start_Time** : indique le temps de début du déclenchement d'une modalité.
- **End_Time** : indique le temps de la fin de l'activité d'une modalité.
- **Index_Fusion** : indique l'index du tableau de fusion. Cette colonne est remplie avec les données nécessaires pendant le processus de fusion pour désigner les modalités qui ont été fusionnées.

Le contenu du tableau *Modality_Main_Parameters* (paramètres principaux de la modalité), pour le cas cité est indiqué au Tableau 2.2. Les significations de chaque élément de ce tableau sont les suivants:

- ***Index*** : présente l'index de chaque paramètre d'une modalité;
- ***Name*** : fournit le nom de chaque paramètre;
- ***Value*** : identifie la valeur de chaque paramètre.

Tableau 2.2. Modèle d'un tableau « *Modality_Main_Parameters* »

Index	Name	Value
1	Action	move
2	Object	Yellow
3	Aim	Put in position
4	Position X	a
5	Position Y	b

De même, le contenu du tableau « *Union_Modality_Main_Parameters* », pour le cas cité, est présenté au Tableau 2.3. Les éléments de ce tableau sont les suivantes :

- ***Index*** : même définition que la précédente;
- ***Index_Modality*** : c'est l'index du tableau « *Modality* »;
- ***Index_Main_Parameters*** : c'est l'index du tableau « *Index_Modality_Parameter* ».

Tableau 2.3. Modèle d'un tableau « Union_Modality_Main_Parameters »

Index	Index_Modality	Index_Main_Parameters
1	1	1
2	1	2
3	1	3
4	2	4
5	2	5

Les contenus du tableau Fusion, du tableau « *Union_Fusion_Main_Parameters* », et celui du « *Fusion_Main_Parameters* » sont tous indiqués dans les tableaux 2.4, 2.5, et 2.6, respectivement.

Les significations des colonnes du Tableau 2.4 sont les suivantes :

- **Index** : même définition ;
- **Name_Fusion** : identifie le nom de la fusion.

Tableau 2.4. Modèle d'un tableau de « fusion »

Index	Name_Fusion
1	fusion 1
2	fusion 1

Le contenu du Tableau 2.5 est comme suit :

- **Index** : même définition qu'auparavant ;

- Index_Param : désigne l'index des paramètres du tableau "Fusion_Main_Parameters" (paramètres principaux de fusion);
- Index_Fusion : indique le numéro d'index de la fusion associée à chaque paramètre.

Tableau 2.5. Modèle d'un tableau de « Union_Fusion_Main_Parameters »

Index	Index_Param	Index_Fusion
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1

Les significations des colonnes du Tableau 2.6 sont les suivantes :

- Index : même définition;
- Name : présente les noms de chaque paramètre impliqué dans la fusion;
- Value : présente les valeurs de chaque paramètre impliqué dans la fusion.

Tableau 2.6. Modèle d'un tableau de «Fusion_Main_Parameters »

Index	Name	Value
1	action	move
2	object	cube
3	old position X	i
4	old position Y	j
5	new position X	a
6	new position Y	b

La signification des colonnes du Tableau 2.7 est la suivante :

- Index : même définition;
- Index_Modality: sert comme index pour le tableau « Modality » ;
- Name : présente le nom des attributs de chaque modalité ;
- Value : contient la valeur des attributs de chaque modalité.

Le Tableau 2.7 contient les attributs de chaque modalité. Dans l'exemple cité, nous n'avons pas utilisé ces attributs. C'est la raison pour laquelle le tableau est vide.

Tableau 2.7. Paramètres ajoutés de la modalité

Index	Index_Modality	Name	Value
1			

2.4.4 Exemple et simulation à l'aide de réseaux de Petri

Afin de mettre en évidence les concepts présentés dans ce chapitre, nous avons développé une application : la location de voitures. Les caractéristiques de cette application sont décrites en utilisant les réseaux de Petri (RdP).

Nous examinons tout d'abord brièvement les concepts de base des RdP, afin de faciliter la compréhension des diagrammes présentés plus loin dans ce chapitre.

Un RdP (ISO/IEC-15909-2 2010; Petri-Nets-Steering-Committee 2010) est un graphe orienté. C'est une technique formelle, graphique « servant à représenter divers systèmes (informatiques, industriels,...) travaillant sur des variables discrètes » (Wiki 2010).

Les RdP sont utilisés pour des variantes déterministes et probabilistes. Ils sont un bon moyen pour modéliser des systèmes concurrents ou de collaboration.

Ils permettent aussi des analyses qualitative ou quantitative qui peuvent être utiles dans la validation du bon fonctionnement du modèle.

Les symboles et les notations utilisées dans un RdP sont présentés dans la Figure 2.14. En effet, les places (représentées par des cercles) sont des états dans un diagramme alors que les transitions (représentées par des rectangles) sont « des actions qui se déroule au sein du système et dont la réalisation dépend de l'état du système » (Wiki 2010).

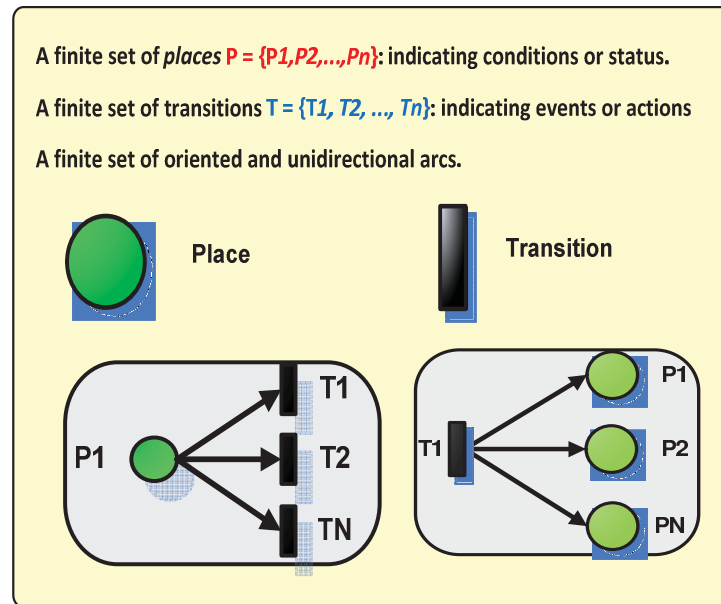


Figure 2.14 Symboles et notations des réseaux de Petri.

Généralement, une action commence dans un état initial (qui se manifeste dans les RdP par un jeton initial dans une place). Quand un jeton va de l'état « a » à l'état « b » à travers une transition, Il se manifeste dans le RdP par un mouvement de jeton de la place « a » vers la place « b » en passant par une transition « x » (Figure 2.15).

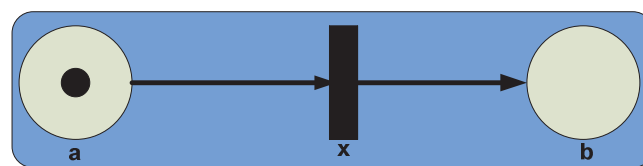


Figure 2.15 Exemple de réseau de Petri.

Le logiciel d'application PIPE2 est utilisé dans la simulation du RdP. PIPE2 (Bonet, Llado et al. 2010) est une source ouverte. C'est un outil indépendant de la plate-forme. Il est utilisé pour créer et analyser des RdP, y compris les RdP stochastiques généralisés.

Comme le montre la Figure 2.16, l'exemple de notre application est celui de la location de voitures. Le menu est composé de quatre sélections (Figure 2.16 (A)) - l'option de location, l'option d'inscription, l'option de gérer la réservation et l'option d'utilisation. Par souci de

simplicité de la discussion, l'option d'utilisation, Figure 2.16 (B), permet à l'utilisateur de sélectionner et d'identifier ses modalités privilégiées. Dans cet exemple, nous avons énuméré 4 modalités: (1) la voix, (2) l'écran tactile (3), le clavier et (4) le regard. Lorsque l'utilisateur s'inscrit pour une location de voiture, la période de location doit être spécifiée. Par conséquent, l'interface permet à l'utilisateur d'indiquer le mois, le jour, et le temps pour le début et la fin de la période de location (Figure 2.16 (C)). Enfin, dans la Figure 2.16 (D), nous présentons une interface qui permet à l'utilisateur de saisir ses coordonnées ainsi que les informations sur sa carte de crédit.

Pour utiliser l'application ci-dessus, citons un exemple. Voulant mieux connaître le pays, quatre étudiants ont décidé de faire un voyage pendant week-end. Ils ont décidé de louer une voiture pour voir certaines régions du pays. Une nuit, François a ouvert son ordinateur équipé d'un écran tactile et s'est connecté à un site de location de voiture. En utilisant la parole, l'écran tactile, le regard et le clavier, il a été capable de faire une réservation de voiture pour la semaine suivante.

Pendant le processus de réservation, des fichiers XML sont créés, avec les différentes modalités utilisées. Ces fichiers sont envoyés vers le serveur de l'entreprise de location de voiture au sein du réseau Internet en utilisant le protocole http. Ils doivent être envoyés d'abord au module "Parser" pour extraire toutes les modalités impliquées. Ensuite, ce module permettra de créer un autre fichier XML qui contient toutes les différentes modalités ainsi que leurs paramètres correspondants. En fin, ce fichier sera envoyé au module « Parameters extacter » qui extrait tous les paramètres des modalités concernées et les envoie au module « fusion ».

2.4.4.1 Scénario

François lance le logiciel de location de voiture ; l'interface initiale est affichée. En utilisant la voix, il choisit « Rent » ; la seconde interface est présentée. Au moyen de l'écran tactile, il choisit « Start Day » et Il tape « 25 » avec le clavier. Ensuite, en utilisant le regard, il

sélectionne « Start Month » et par la voix, il dit « January ». Puis, il sélectionne « Start Time » à l'aide du clavier, et il tape « 1:30 pm ». Ensuite, il choisit « End Day » par la voix et il dit « 27 ». Puis, il sélectionne « End Month » au moyen du clavier et il tape « 1 ». Enfin, il choisit « End Time » au regard et il dit « 6:00 pm ». À la fin du processus de réservation, il reçoit un message de confirmation à travers son ordinateur portable. Ce scénario est représenté dans le schéma de la Figure 2.18.

2.4.4.2 Grammaire

Le schéma de la Figure 2.19 montre la grammaire utilisée pour les interfaces A et B (voir Figure 2.16) de l'exemple de location de voiture. En effet, le choix, est défini comme une sélection de l'un des menus (location, inscrivez-vous, gérer la réservation et l'utilisation) dans l'interface utilisateur. Le choix du temps est au format de l'horloge américain (par exemple: 12 :30 pm), le choix du mois peut être numérique (par exemple : 1) ou alphabétique (par exemple : January) et l'interface C, entre autres, est définie comme un choix de soit une entrée du mois, jour ou l'heure, ou bien un retour / suivant / annuler la sélection.

2.4.4.3 Simulation 1

Le schéma de la Figure 2.20 montre les interactions impliquées dans l'interface A dans laquelle l'utilisateur devra choisir une option dans un menu de location de voiture. Le reste du diagramme montre l'ensemble des activités pendant et après le choix de l'option "Rent" via la parole. Comme il est déjà montré, un fichier XML est créé et envoyé par la suite au réseau. Le module « Parser » analyse les données XML et extrait les balises qui contiennent les informations de la modalité y compris ses paramètres associés. Le module d'extraction de paramètres extrait les paramètres nécessaires et les transmet ensuite au module « Fusion And Multimodality ». Comme il s'agit d'une action unique, dans cet exemple, aucune fusion n'est appliquée. Bien qu'il s'agisse d'une action unimodale, elle est tout de même enregistrée dans la base de données. L'interface B et tous les menus associés à l'option "Location" doivent être instanciés.

2.4.4.4 Simulation 2

Le schéma de la Figure 2.21 décrit les activités du système lorsque l'option « Rent » est sélectionnée (voir le jeton dans le lieu appelé "Rent Selected" au milieu du réseau). En même temps que cette option est sélectionnée, les trois modalités sont également sélectionnées (voir les jetons dans les modalités clavier, parole et écran tactile).

Le diagramme du RdP nous montre toutes les variations possibles qui pourraient survenir. Par exemple, lorsque le clavier est sélectionné (coin supérieur droit du réseau), la prochaine transition possible peut être l'une des suivantes: (1) Sélectionner le mois par le clavier, ou (2) Sélectionner le Jour par le clavier ou (3) Sélectionner l'heure par le clavier. La même remarque s'applique pour les deux autres modalités - parole et l'écran tactile. La sortie désirée est une entrée de données pour le mois, le jour ou l'heure. Elle doit être appliquée en utilisant une seule modalité par paramètre.

Dans le diagramme, nous donnons un exemple de l'un des nombreux résultats possibles. Ici, le mois est choisi par l'écran tactile, le jour est sélectionné en utilisant le clavier et le temps est choisi par la voix. Nous avons coloré les états pour faciliter la lecture : le jaune est associé avec la voix, le bleu pour la modalité clavier et le rose pour la modalité écran tactile, alors que le cercle rouge indique qu'il est supposé que l'option "Rent" est déjà sélectionnée.

2.4.4.5 Simulation 3

Le RdP de la Figure 2.22 montre les différents états et les transitions lorsque l'utilisateur utilise quatre modalités (par exemple le clavier, la parole, l'écran tactile et le regard) pour la sélection et la saisie des données, pour « Start Month », « Start Day » et « Start Time ». Ce diagramme montre une partie du RdP impliquée dans l'exécution des activités. Encore une fois, dans le but de simplicité, nous avons mis des couleurs sur les places du réseau: (1) le bleu - pour les opérations multimodales dans lesquelles l'écran tactile est utilisé pour

sélectionner « Start Day » et le clavier pour entrer « 25 », (2) le vert - pour montrer les places et les transitions en cause lorsque l'utilisateur utilise le regard pour sélectionner « Start Month » et la parole pour entrer « January », et (3) la couleur rose - lorsque « Start Time » est sélectionnée à l'aide du regard, « 1:30 » est prononcé vocalement, et « pm » est entrée par le clavier.

A la fin, l'utilisateur doit confirmer la réservation qu'il a effectuée. Le RdP Figure 2.23 montre la confirmation des transactions en utilisant des modalités différentes.

(A) Car rental menu

(B) Modalities selection available to the user

(C) Data entry for Start/End month, day and time

(D) Data entry for client's information

Figure 2.16 (A) Menu principal de la location de voiture (B) Modalités disponibles (C) Informations concernant date de locations (D) Informations concernant locateur.

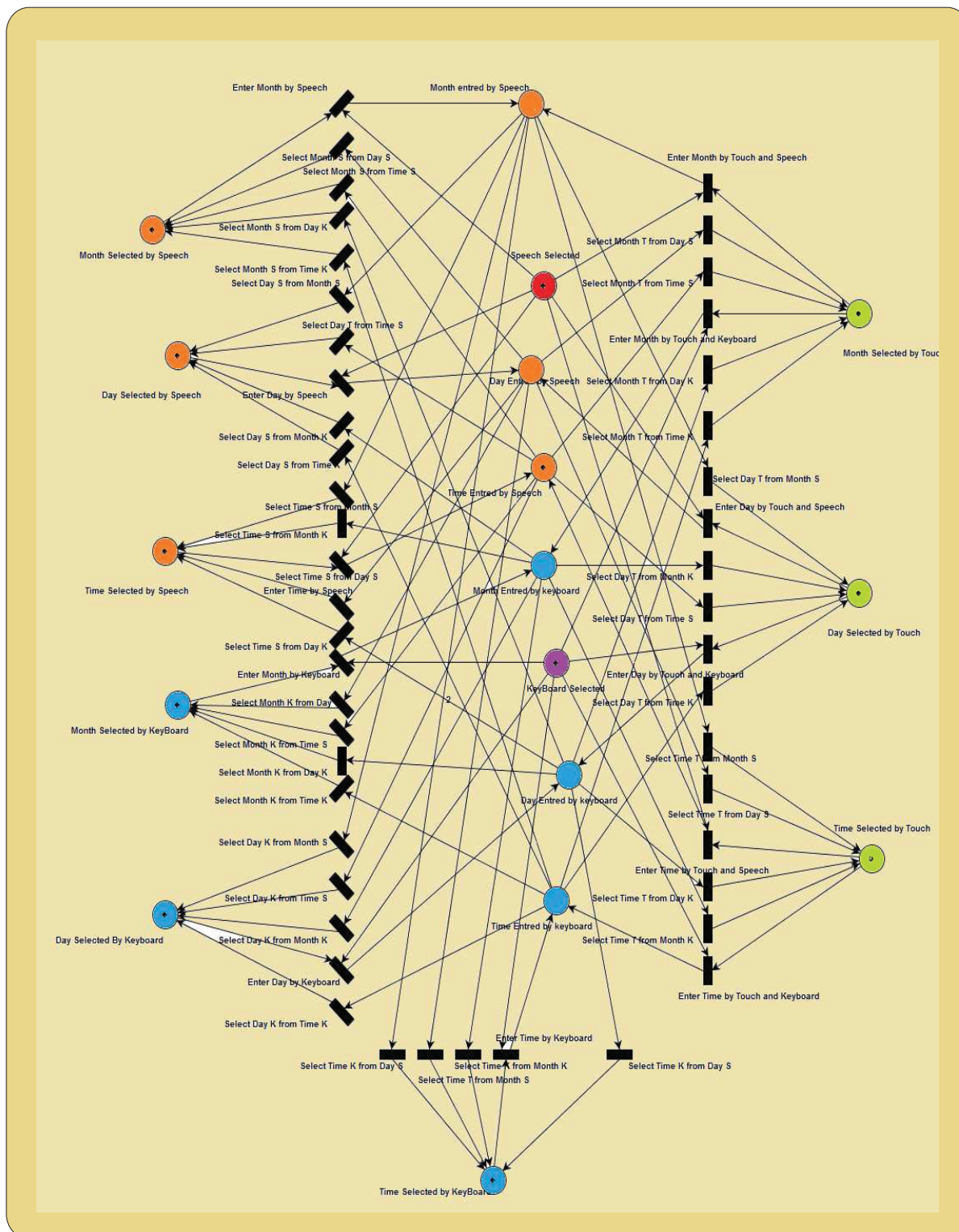


Figure 2.17 Réseau de Petri montre les entrées pour le mois , le jour et le temps en utilisant des modalités différentes.

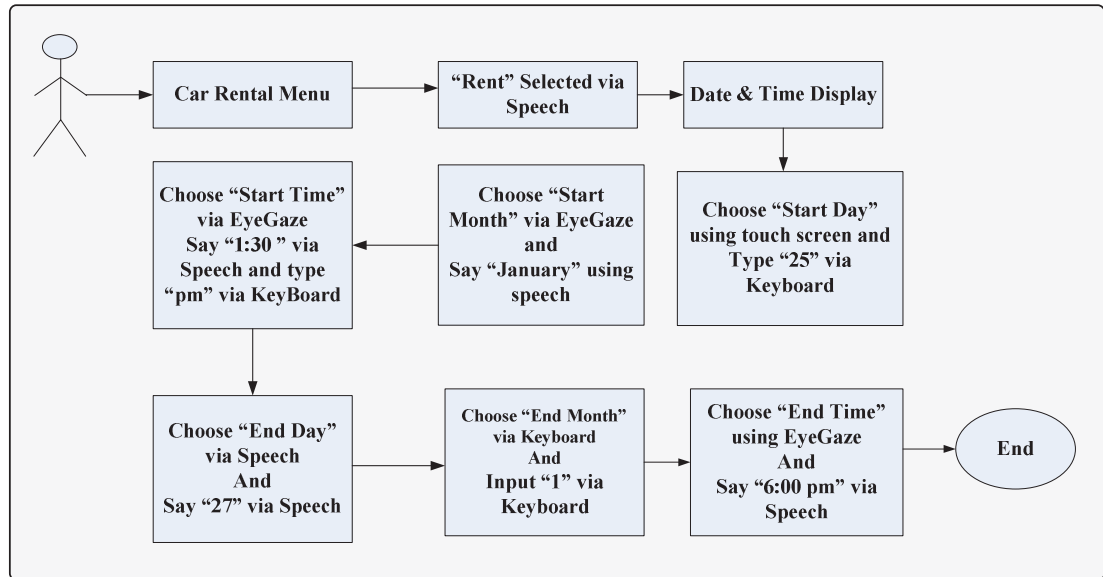


Figure 2.18 Un simple scenario montre les interactions multimodales entre l'utilisateur et la machine.

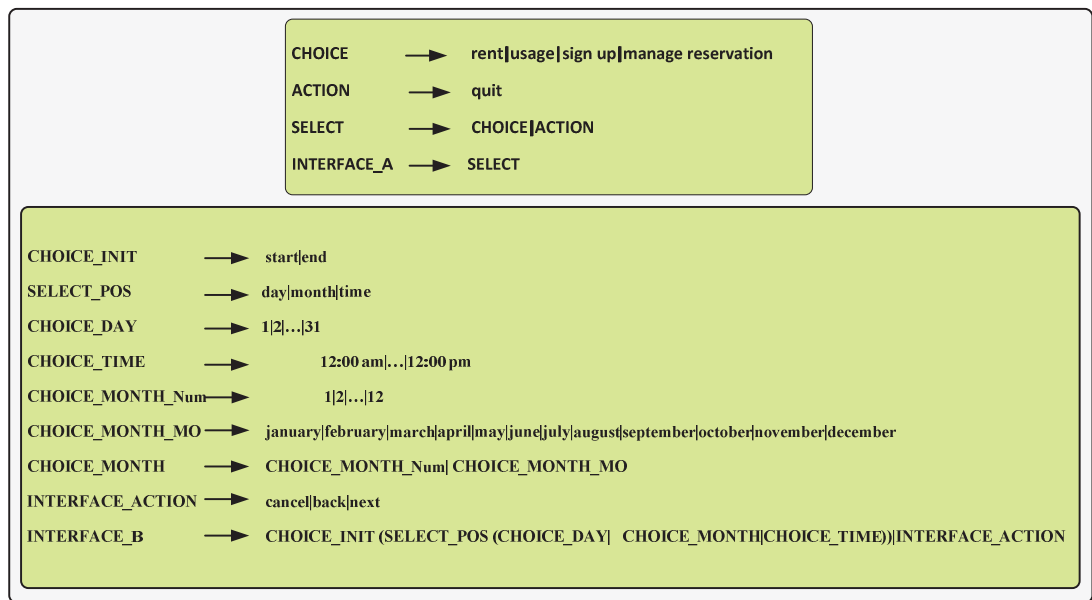


Figure 2.19 Grammaire pour la période de location de voiture.

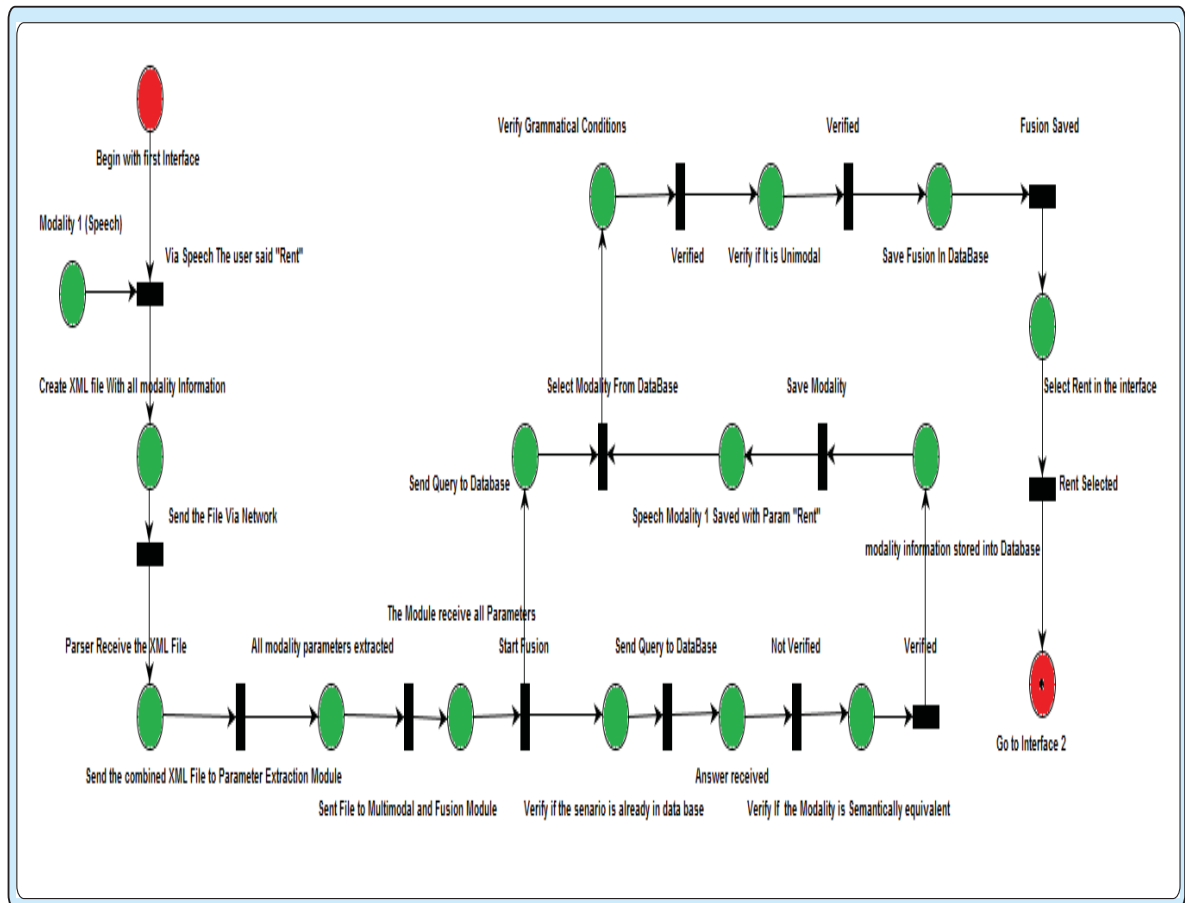


Figure 2.20 Les activités du système quand l'utilisateur choisi "Rent" en utilisant la voix.

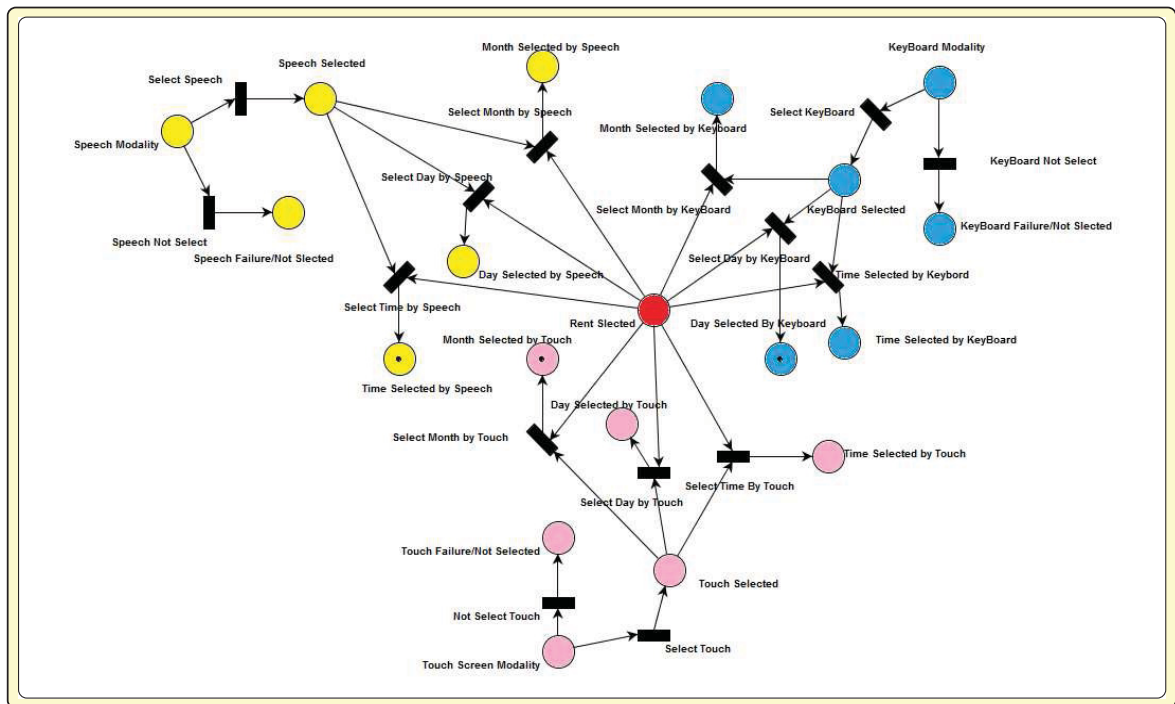


Figure 2.21 Réseaux de Petri montre toutes les variations possibles pour les données entrées pour mois, jour et temps en utilisant parole, clavier et écran tactile pour la sélection de "Rent".

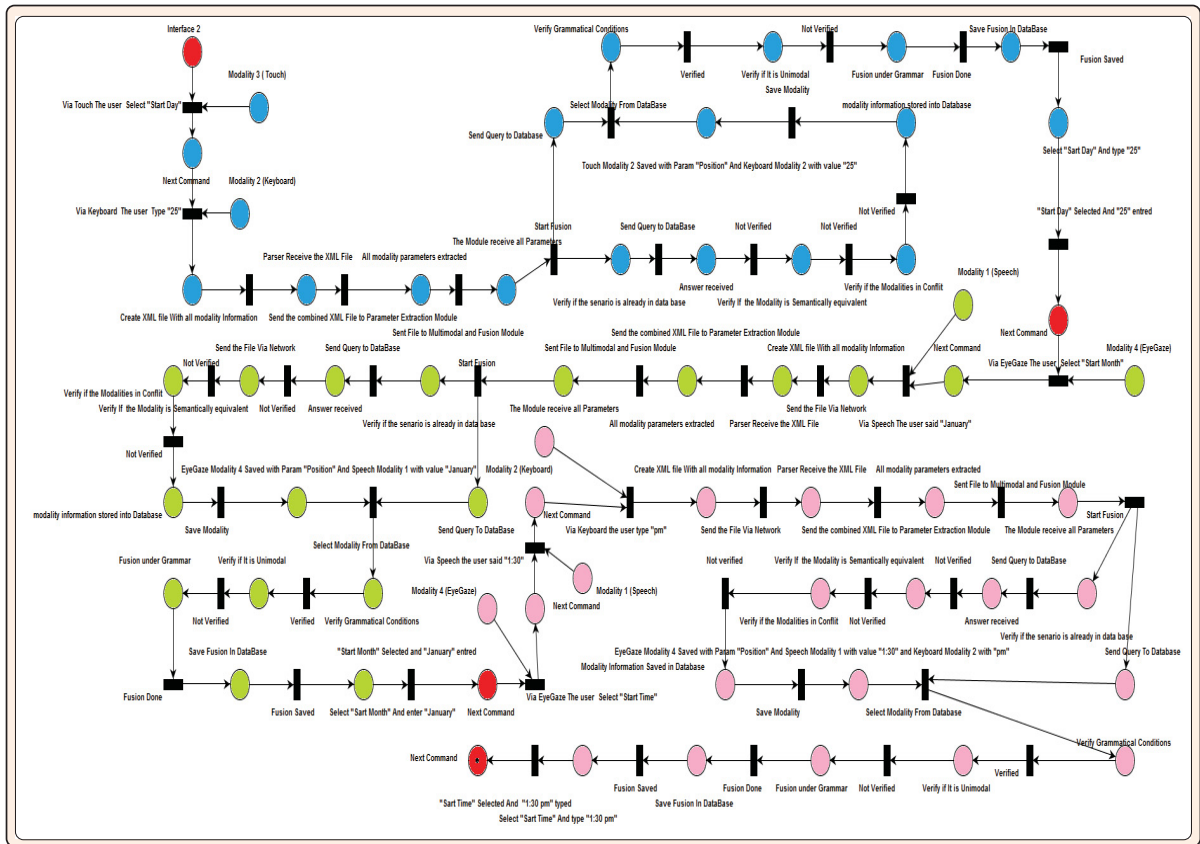


Figure 2.22 Réseau de Petri présente toutes les opérations impliquant différentes modalités que l'utilisateur choisit pour "Start Month ", " Start Day "et" Start Time ".

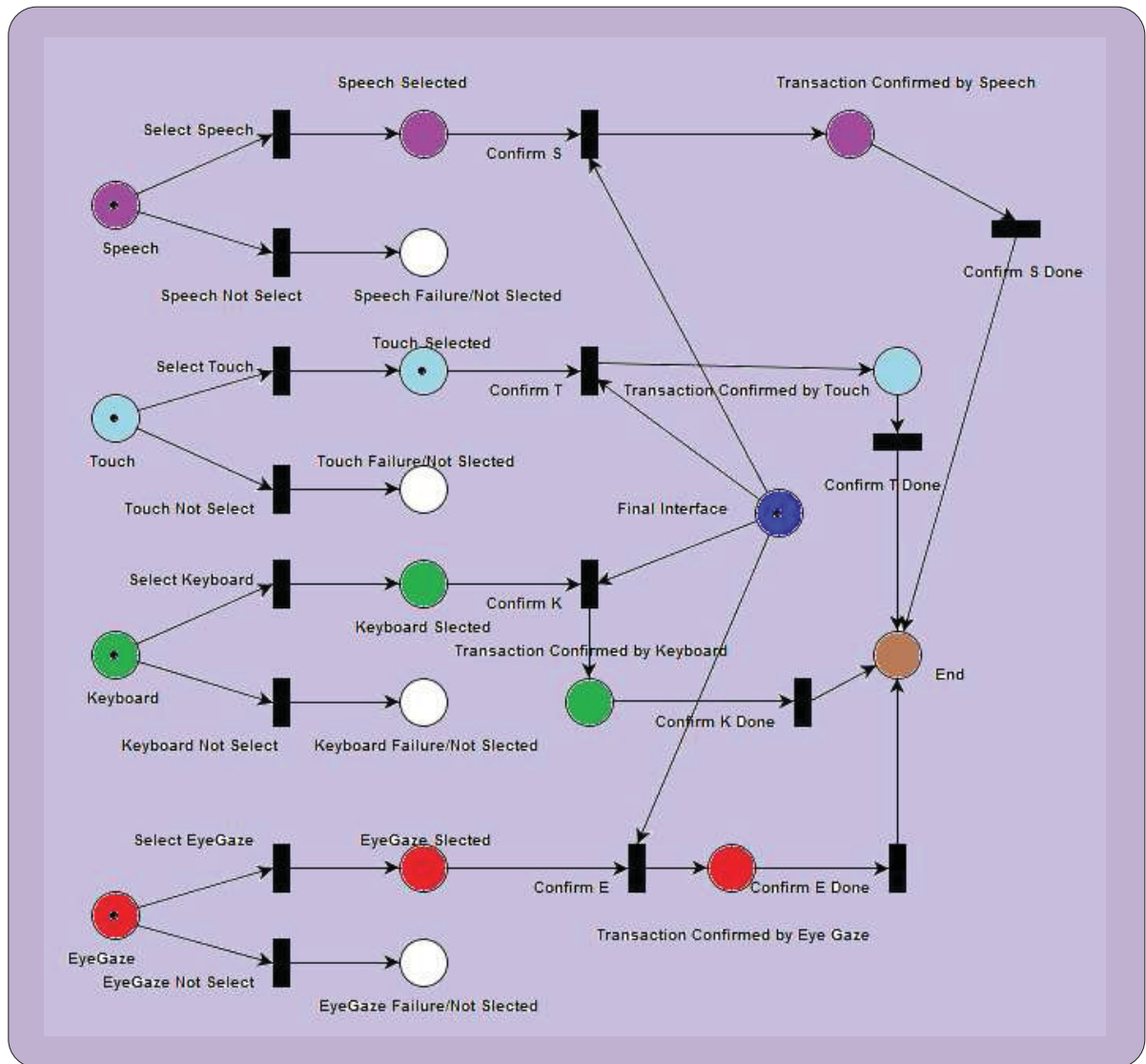


Figure 2.23 Réseau de Petri pour la confirmation des transactions.

2.5 Implémentation

En utilisant le langage de programmation JAVA nous avons implémentés les trois modules de notre système. Nous avons utilisé quatre ordinateurs connectés à internet pour réaliser notre implémentation. Figure 2.24.

Le Tableau 2.8 suivant décrit chaque ordinateur et le module qui le contient.

Tableau 2.8. Caractéristiques des ordinateurs utilisés

Ordinateur	Caractéristiques	Module
Ordinateur 1	Operating system : XP Connexion Internet : WIFI Adresse IP: 192.168.2.100 Processor: Xeon (TM) CPU 3.40 GHz	Utilisateur
Ordinateur 2	Operating system : Win7 Connexion Internet : file Adresse IP: 192.168.2.42 Processor: Intel®Core (TM) 2CPU 1.67 GHz	Parser
Ordinateur 3	Operating system : Vista Connexion Internet : file Adresse IP: 192.168.2.75 Processor: Pentium(4) 1.3 GHz	Parameter Extractor
Ordinateur 4	Operating system : XP Connexion Internet : file Addressee IP: 192.168.2.68 Processor: Intel®Core (TM) 2CPU 1.67 GHz	Database

L'étape (1) consiste à envoyer des fichiers XML a partir de l'ordinateur utilisateur (ordinateur 1) vers l'ordinateur qui contient le module « Parser » dans notre cas c'est l'ordinateur 2. Ce dernier reçoit les fichiers (étape 2), le module « Parser » extrait les modalités qu'on a besoin (étape 3) et envoi le fichier combiner (étape 4) vers l'ordinateur qui contient le module « Parameter Extractor ». Ce dernier reçoit le fichier combiné (étape 5) puis envoi des requêtes (étape 7) vers l'ordinateur qui contient la base de données (ordinateur 4) pour sauvegarder tous les paramètres (étape 9) de différentes modalités.

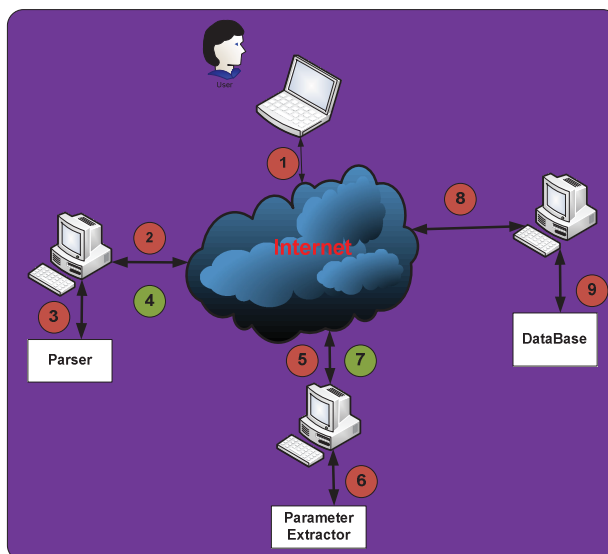


Figure 2.24 Implémentation du Système.

Le transfert des fichiers entre tous les modules est fait par l'utilisation de la classe Socket implémentée par JAVA. «Un socket représente un point terminal d'un canal de communication entre deux ou plusieurs processus ».

2.6 Conclusion

Au cours de ce chapitre, nous avons présenté une approche flexible comparée à celle citée dans la littérature. Cette approche permet à l'utilisateur de choisir les modalités appropriées à sa situation (plusieurs modalités). Le processus de fusion n'est plus fondé sur des modalités prédéfinies. À cet effet, nous avons implémenté un service Web qui permet la réalisation de cette fusion multimodale.

CONCLUSION

L'avancement technologique actuel a créé la nécessité de produire des machines de plus en plus performantes, facile à utiliser et permettant 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, des préférences et des intentions communicationnelles.

Au cours de notre étude, nous avons proposé une architecture pour les applications multimodales. Intégrée sous forme de service Web, cette architecture permet de réaliser la fusion multimodale. Dans notre travail, le processus de fusion n'est plus fondé sur des modalités qui sont prédéfinies dès le départ. De là, l'utilisateur choisit les modalités qu'il juge approprié pour sa situation.

De ces faits, nous avons créé un système qui utilise des modules distribués communiquant ensemble pour réaliser cette fusion. Des fichiers XML ont été utilisés pour capturer les différentes actions multimodales ainsi que leurs paramètres pour permettre cette communication. Un analyseur a été utilisé pour extraire les paramètres et les noms des modalités utilisés de ces fichiers XML. Ainsi, nous avons développé un algorithme permettant de réaliser la fusion de deux ou plusieurs modalités. Pour valider les différentes approches établies, nous avons simulé un exemple concret schématisant le fonctionnement de notre système. L'exemple d'application a été présenté en utilisant les réseaux de Petri pour un système de location de voitures.

Un développement possible sera l'extension de nos travaux au contexte d'interaction. Ce dernier, utilise des paramètres tels que le contexte de l'utilisateur, le contexte de l'environnement et le contexte du système dans le processus de fusion. Le contexte d'interaction permet la sélection de modalités appropriées selon le contexte actuel de l'utilisateur.

Un autre développement futur pour notre projet consiste à utiliser les réseaux de Petri colorés.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Ahuja, S. P. and R. Quintao (2000). Performance evaluation of Java RMI: a distributed object architecture for Internet based applications. Proceedings of MASCOTS 2000. Eighth International Symposium on Modeling, Analysis and Simulation and Telecommunication Systems. San Francisco, CA, USA: 565-569.
- Bates, B. and K. Sierra (2003). Head First Java: Your Brain on Java - A Learner's Guide, O'Reilly Media.
- Caschera, M. C., A. D'Andrea, et al. (2009). ME: Multimodal Environment Based on Web Services Architecture. OTM 2009 Workshops. Vilamoura, Portugal, Springer: 514-512.
- Daniel, J. (2003). Services Web, HUROPÉ, S.L.
- Daniel, J. (2003). Web : concepts, techniques et outils Vuibert informatique.
- Desmet, C., R. Balthazor, et al. (2005). "<emma>: re-forming composition with XML." Literary & Linguistic Computing **20**(1): 25-46.
- Djendi, H. (2007). Architecture et modèles dynamique dédiés aux applications multimodales. Électrique, École Technologie supérieur: 390.
- Englander, R. (2003). Java and Soap, O'Reilly.
- Fay-Wolfe, V., L. C. DiPippo, et al. "Real-time CORBA." IEEE Transactions on Parallel and Distributed Systems **11**(10): 1073-1089.
- Fierrez-Aguilar, J., J. Ortega-Garcia, et al. (2003). "A Comparative Evaluation of Fusion Strategies for Multimodal Biometric Verification." Audio- and Video-Based Biometric Person Authentication **2688**(10): 830-837.
- Giuliani, M. and A. Knoll (2008). MultiML: a general purpose representation language for multimodal human utterances. International Conference on Multimodal Interfaces Chania, Crete, Greece ACM: 165-172.
- Hocek, A. and D. Cuddihy (2003). Definitive VoiceXML Upper Saddle River.
- Honkala, M. and M. Pohja (2006). Multimodal Interaction with XForms. International Conference On Web Engineering. Palo Alto, California, USA, ACM: 201-208.

- Hoschka, P. (2002). "An introduction to the Synchronized Multimedia Integration Language." IEEE **5**(4): 84-89.
- ISO/IEC-15909-2. (2010). "Petri Nets." from <http://www.petrinets.info/>.
- Johnston, M. (2009). Building multimodal applications with EMMA. ICMI 05, International Conference on Multimodal Interfaces. Cambridge, MA, USA, ACM.
- Mohan, C. K., N. Dhananjaya, et al. (2008). Video shot segmentation using late fusion technique. 7th International Conference on Machine Learning and Applications. San Diego, CA, USA, IEEE: 267-270.
- Ngolo, M., L. B. Palma, et al. (2009). Architecture for remote laboratories based on REST web services. E-Learning in Industrial Electronics, 2009. ICELIE '09. 3rd IEEE International Conference on. Porto: 30 - 35
- Oviatt, S., P. Cohen, et al. (2000). "Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions." Human-Computer Interaction **15**(4): 263-322.
- Oviatt, S. L. and P. R. Cohen (2000). "Multimodal Interfaces that Process What Comes Naturally." Communications of the ACM **43**(3): 45 - 53.
- Pagni, M., J. Hau, et al. (2008). A multi-protocol bioinformatics Web service: use SOAP, take a REST or go with HTML. 2008 8th International Symposium on Cluster Computing and the Grid. Lyon, France, IEEE: 728-734.
- Paleari, M. and S. Antipolis (2006). Toward Multimodal Fusion of Affective Cues Proceedings of the 1st ACM international workshop on Human-centered multimedia. Santa Barbara, California, USA, ACM: 99 - 108
- Pérez, G., G. Amores, et al. (2005). Two strategies for multimodal fusion. ICMI'05 Workshop on Multimodal Interaction for the Visualisation and Exploration of Scientific Data. Trento, Italy, ACM.
- Petri-Nets-Steering-Committee. (2010). "Petri Nets World." from <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>.
- Pfleger, N. (2004). Context Based Multimodal Fusion. ICMI 04. Pennsylvania, USA, ACM: 265 - 272.
- Pitsikalis, V., A. Katsamanis, et al. (2009). "Adaptive Multimodal Fusion by Uncertainty Compensation." Audio, Speech, and Language Processing, IEEE Transactions on **17**(3).

- Portillo, P. M., G. P. García, et al. (2006). Multimodal fusion: a new hybrid strategy for dialogue systems. Proceedings of the 8th international conference on Multimodal interfaces Banff, Alberta, Canada 357 - 363.
- PostgreSQL. (2010). from <http://www.postgresql.org/>.
- Pous, M. and L. Ceccaroni (2010). Multimodal Interaction in Distributed and Ubiquitous Computing. Fifth International Conference on Internet and Web Applications and Services Barcelona, Spain: 457-462.
- Robert, P. (2010). le nouveau petit Robert de la langue française
- Robert, S., K. Khaled, et al. (2005). Mobile Web Services Discovery and Invocation Through Auto-Generation of Abstract Multimodal Interface. ITCC 2005 International conference on Information Technology. Las Vegas, NV, IEEE. **35-43**.
- Roy, M. and A. Ewald (1997). "Inside DCOM" DBMS **10**(4): 26-28, 30, 32,34.
- Snoek, C. G. M., M. Worring, et al. (2005). Early versus late fusion in semantic video analysis. 13th annual ACM international conference on Multimedia. Hilton, Singapore, ACM.
- Sun, Y., F. Chen, et al. (2006). A Novel Method for Multi-sensory Data Fusion in Multimodal Human Computer Interaction. Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments Sydney, Australia, ACM: 401 - 405.
- Sureswaran, R., H. Al Bazar, et al. (2009). Active e-mail system SMTP protocol monitoring algorithm. Proceedings of the 2009 2nd IEEE International Conference on Broadband Network & Multimedia Technology. Beijing, China, IEEE: 257-260.
- Tsutsui, T., S. Saeyor, et al. (2000). MPML: A multimodal presentation markup language with character agent control functions. Proceedings of WebNet 2000 World Conference on the WWW and Internet. San Antonio.
- Tzerefos, P., C. Smythe, et al. (1997). A comparative study of Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP) and X.400 Electronic Mail Protocols. Local Computer Networks, 1997. Proceedings., 22nd Annual Conference on Minneapolis, MN: 545 - 554.
- W3C. (2010). "EMMA." from <http://www.w3.org/TR/emma>.
- W3Cd. (2010). "FTP." from <http://www.w3.org/Protocols/rfc959/>.
- W3Ce. (2010). "HTTP." from <http://www.w3.org/Protocols/>.

- W3Cf. (2010). "SOAP." from <http://www.w3.org/TR/soap/>.
- W3Cg. (2010). "WSDL." from <http://www.w3.org/TR/wsdl>.
- W3Ch. (2010). "VOICEXML." from www.w3.org/TR/xhtml+xml+voice.
- W3Cj. (2010). from <http://www.w3.org/>.
- W3Ck. (2010). "PLS (Pronunciation Lexicon Specification)." from <http://www.w3.org/TR/2008/RFC-pronunciation-lexicon-20081014/#term-IPA>.
- W3Cm. (2010). "Ink Markup Language Tags." from www.w3.org/TR/InkML.
- W3Cn. (2010). "XHTML + Voice." from www.w3.org/TR/voice-tts-reqs.
- W3Cp. (2010). "XML." from www.w3.org/TR/XML.
- W3Cq. (2010). "Web Services." from <http://www.w3.org/2002/ws/>.
- Wang, D., J. Zhang, et al. (2006). A Multimodal Fusion Framework for Children's Storytelling Systems. LNCIS Berlin / Heidelberg, Springer-Verlag. **3942/2006**: 585-588.
- Wang, F., J. Li, et al. (2007). "A space efficient XML DOM parser." Data & Knowledge Engineering **60**(1): 185-207.
- Wang, F., J. Lia, et al. (2007). "A space efficient XML DOM parser." Data & Knowledge Engineering **60**(1): 185-207.
- Wiki. (2010). "namespace." from <http://fr.wikipedia.org/wiki>.
- Wöllmer, M., M. Al-Hames, et al. (2009). "A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams." Neurocomputing **73**(1-3): 366-380.
- Wyke, A., S. Rehman, et al. (2002). Manuel de Référence XML, Vivendi Universal Publishing Services.
- Xu, T., K. Yuan, et al. (2009). A real-time information hiding algorithm based on HTTP protocol. Conference on Network Infrastructure and Digital Content, IEEE IC-NIDC2009. Beijing, China, IEEE: 618-622.
- Zhang, R. and L. Guan (2009). Multimodal image retrieval via bayesian information fusion. Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on New York, NY 830-834.