



Design and Optimization of Distributed Information Systems of Services to Aid Urban Mobility Based on a Flexible Ontology in the Transport Domain

Sawsan Saad

► To cite this version:

Sawsan Saad. Design and Optimization of Distributed Information Systems of Services to Aid Urban Mobility Based on a Flexible Ontology in the Transport Domain. Other. Ecole Centrale de Lille, 2010. English. NNT : 2010ECLI0017 . tel-00586086

HAL Id: tel-00586086

<https://theses.hal.science/tel-00586086>

Submitted on 14 Apr 2011

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

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

N° d'ordre : 144

Ecole Centrale de Lille

THESE

Présentée en vue d'obtenir le grade de

DOCTEUR

Spécialité : Automatique et Informatique Industrielle

Par

Sawsan SAAD

Doctorat délivré par l'Ecole Centrale de Lille

**Conception et Optimisation Distribuée d'un Système d'Information des Services d'Aide à
la Mobilité Urbaine Basé sur une Ontologie Flexible dans le Domaine de Transport**

Soutenue publiquement le 10 décembre 2010 devant le jury :

Président :	Christophe DI POMPEO	Professeur, Université Lille 2
Rapporteurs :	Mariagrazia DOTOLI	Professeur, Poltecnico di Bari
	Mekki KSOURI	Professeur, , ENIT
Examineurs :	Farouk KAMOUN	Professeur, ENSI
	Moncef ZOUARI	Pdg, Sevenair
	Hayfa ZGAYA	Dr. MCU. Université Lille 2 (Co-encadreur)
Directeur :	Slim HAMMADI	Professeur, Ecole Centrale de Lille

Thèse préparée dans le laboratoire LAGIS FRE 3303 à l'Ecole Centrale de Lille.

Ecole Doctorale SPI 072

PRES Université Lille Nord-de-France

N° d'ordre : 144

Ecole Centrale de Lille

THESE

Présentée en vue d'obtenir le grade de

DOCTEUR

Spécialité : Automatique et Informatique Industrielle

Par

Sawsan SAAD

Doctorat délivré par l'Ecole Centrale de Lille

**Design and Optimization of Distributed Information Systems of Services to Aid Urban
Mobility Based on a Flexible Ontology in the Transport Domain**

Soutenue publiquement le 10 décembre 2010 devant le jury :

Président :	Christophe DI POMPEO	Professeur, Université Lille 2
Rapporteurs :	Mariagrazia DOTOLI	Professeur, Poltecnico di Bari
	Mekki KSOURI	Professeur, , ENIT
Examineurs :	Farouk KAMOUN	Professeur, ENSI
	Moncef ZOUARI	Pdg, Sevenair
	Hayfa ZGAYA	Dr. MCU. Université Lille 2 (Co-encadreur)
Directeur :	Slim HAMMADI	Professeur, Ecole Centrale de Lille

Thèse préparée dans le laboratoire LAGIS FRE 3303 à l'Ecole Centrale de Lille.

Ecole Doctorale SPI 072

PRES Université Lille Nord-de-France

Abstract

Nowadays, information related on displacement and mobility in a transport network represents certainly a significant potential. Indeed, we can imagine infinity of innovating services related to mobility, not only intended for general public, but also for companies, to a relevant control of the mobility for their displacement plans. The main goal of our work is to provide a system with a mobility assistance, which is related to the daily or occasional traffic motivations, tourism and culture occupations, etc. with a possibility to profit from relevant and exploitable heterogeneous information.

This work aims to modelling, to optimize and to implement an Information System of Services to Aid the Urban Mobility (ISSAUM). Indeed, our ISSAUM optimizes the management of the data flow of the users' requests, which can be simultaneous and numerous. That's why; the ISSAUM has firstly to decompose each set of simultaneous requests into a set of independent sub-requests called tasks. Each task corresponds to a service which can be proposed differently by several information providers, in competition, with different cost, response delay, different size and different format of the data. An information provider, which aims to propose some services through our ISSAUM, has firstly to register its information system and its ontology, by assuming the responsibility for the legal and qualitative aspects of the correspondent data. Thus, the ISSAUM is related to an Extended and distributed Transport Multimodal Network (ETMN) which contains several heterogeneous data sources including the different proposed services to transport users.

The dynamic and distributed aspects of the problem incite us to adopt a multi-agent approach to ensure a continual evolution and a pragmatic flexibility of the system. The proposed multi-agent system is based on meta-heuristic for the research and the composition of the services. So, we proposed to automate the modeling of services by using the ontology idea. The ontology solution aids the information retrieval between the different transports information providers (servers) in the EMTN. In fact, our ISSAUM aims to support the transport users in planning their travels. In this context, the user defines (the departure city, the arrival city, the date, etc...) of his travel. With this information, our system looks for all trips possibilities in the different servers by using his ontology. The services research is based on the Mobile Agent paradigm (MA) using a dynamic optimisation algorithm for the MA Workplans design.

Our ISSAUM takes into account possible disturbance through the EMTN (crash, bottlenecks, etc.) in order to satisfy user requests in all the cases. For that, we developed a negotiation protocol between optimiser agents and mobile agents of the system. The proposed ontology mapping negotiation model based on the knowledge

management system for supporting the semantic heterogeneity and it organized as follow: the first layer contains the *Negotiation Layer (NL)*. The second layer represents the *Semantic Layer (SEL)*, and the third layer is the *Knowledge Management Systems Layer (KMSEL)* which uses ontology in purpose of automatic classifying and using of the new ontologies and meta-ontologies. Our approach aims to make the agents able to understand each other when using these ontologies and by applying the mapping services to resolved the misunderstanding problem.

Finally, we detailed the reassignment process by using Dynamic Reassigned Tasks (DRT) algorithm supporting by ontology mapping approach. The DRT Mapping algorithm based on the current state of travelling mobile agents in their correspondent routes called Workplans. Our goal is to give users all needed information even if some information providers are no longer available. Thus we improve the Quality of Services (QoS) of the response time with the best cost.

Experimental results presented in this thesis, justify the usage of the ontology solution in our system and its role in the negotiation process. The different experimental scenarios show a pertinent management of any amount of Δ_ϵ – simultaneous requests. Indeed, a great number of user requests through a short period of time Δ_ϵ , does not affect the system functioning which decompose them by using its flexible ontology, identifies the required services and the possible information providers.

Key-words:

Urban mobility, multi-agent system, Multimodal transport, distributed optimization, Services Modelling, negotiation, mobile agent, ontology, mapping ontology.

Résumé

De nos jours, les informations liées au déplacement et à la mobilité dans un réseau de transport représentent sans aucun doute un potentiel important. En effet, on peut imaginer une infinité de services innovants liés à la mobilité, non seulement à destination du grand public, mais également à des entreprises, dans le conseil en mobilité pour leurs plans de déplacement. Le but de cette thèse est donc de fournir un système d'aide à la mobilité qui s'articule autour des motifs de déplacements quotidiens, occasionnels, de tourisme, de culture, etc. avec la possibilité de pouvoir bénéficier d'une information pertinente et exploitable.

Ces travaux de recherche visent à mettre en œuvre un Système d'Information de Service d'Aide à la Mobilité Urbaine (SISAMU) pour optimiser la gestion de flux des requêtes utilisateurs qui peuvent être nombreuses et simultanées. Dans ce cas, le SISAMU doit pouvoir procéder par des processus de décomposition des requêtes simultanées en un ensemble de tâches indépendantes. Chaque tâche correspond à un service qui peut être proposé par plusieurs fournisseurs d'information en concurrence, avec différents coûts, temps de réponse et formats. Un fournisseur d'information voulant proposer ses services via le SISAMU, doit d'abord y enregistrer son système d'information et son ontologie. Le SISAMU est donc lié à un Réseau informatique Etendu et distribué de Transport Multimodal (RETM) qui comporte plusieurs sources d'information hétérogènes des différents services proposés aux utilisateurs de transport.

L'aspect dynamique, distribué et ouvert du problème, nous a conduits à adopter une modélisation multi-agent pour assurer au système une évolution continue et une flexibilité pragmatique. Le système multi-agent proposé s'appuie sur les métaheuristiques pour la recherche et la composition des services. Pour ce faire, nous avons proposé d'automatiser la modélisation des services en utilisant la notion d'ontologie. En effet, l'utilisation de l'ontologie facilite la recherche d'information entre les différents fournisseurs de services de transport dans le RETM. Dans ce contexte, l'utilisateur définit (la ville de départ, la ville d'arrivée, la date, etc.) de son voyage. Avec ces informations, notre système cherche toutes les possibilités de voyages dans les serveurs différents en utilisant son ontologie. Le système proposé vise ainsi à soutenir les utilisateurs de transport pour la planification de leurs voyages. Par ailleurs, la recherche des services se base sur le paradigme Agent Mobile (AM) utilisant un algorithme d'optimisation dynamique de construction des Plans De Routes (PDR).

Notre SISAMU prend en considération les éventuelles perturbations sur le RETM (pannes, goulets d'étranglements, etc.) afin de satisfaire les requêtes utilisateurs dans

tous les cas de figure. A cet effet, nous avons créé un protocole de négociation entre les agents mobiles et les agents responsables des choix des fournisseurs d'information pour les services demandés, ces agents sont appelés agents Ordonnanceurs.

Le protocole proposé dépasse les limites d'une communication agent traditionnelle, ce qui nous a incités à associer au système une ontologie flexible qui permet d'automatiser les différents types d'échange entre les agents grâce à un vocabulaire approprié. Le protocole de négociation proposé qui utilise l'ontologie de la cartographie se base sur un système de gestion des connaissances pour soutenir l'hétérogénéité sémantique. Par la suite, l'architecture proposée est organisée sous forme de différentes couches : la Couche Négociation (CN), la Couche Sémantique (CS), et la Couche Systèmes de Gestion de Connaissances (CSGdC). La solution proposée vise à résoudre les problèmes d'incompréhension qui peuvent avoir lieu au cours des processus de négociation entre les agents communicants, et ce en utilisant le service d'ontologie de la cartographie.

Finalement, nous avons détaillé l'Algorithme de Reconstruction Dynamique des Chemins des Agents (ARDyCA) qui est basé sur l'approche de l'ontologie cartographique. L'ARDyCA se base sur la position courante des agents mobiles dans leurs itinéraires respectifs (appelés plans de travail). Notre objectif est de donner aux utilisateurs toutes les informations nécessaires, même si certains fournisseurs d'information ne sont plus disponibles. Ainsi, nous améliorons la qualité de services (QdS) en offrant un temps de réponse optimisé avec un moindre coût.

Les résultats expérimentaux présentés dans cette thèse justifient l'utilisation de l'ontologie dans notre système et son rôle dans le processus de négociation. Les simulations présentées montrent différents scénarios de gestion d'un nombre de requêtes Δ_ϵ – simultanées plus ou moins important. Quelque soit l'ordre d'importance du nombre de requêtes utilisateurs ; si celles-ci sont formulées pendant un court laps de temps Δ_ϵ ; le fonctionnement du système ne s'en trouve pas affecté. En effet, le système se charge de la décomposition des requêtes en utilisant l'ontologie flexible adoptée et l'identification des services demandés et des fournisseurs d'information susceptibles d'y répondre.

Mots-clés

Transport multimodal, optimisation distribuée, algorithmes évolutionnaires, systèmes multi-agents, négociation, agent mobile, ontologie, ontologie de la cartographie.

To the memory of my mother

To my dear father

To my dears brothers, sisters and their families

To my dear Mohannad

To my city Salamieh

To my country Syria

... with my love and gratitude

و اخيرا اليك و حذك أبي ... أهدي جهدي هذا و تعبني ... أحبك

Acknowledgements

First, I would like to thank my supervisor, Professor, Slim Hammadi, for his helpful supervision and continuous encouragement throughout, his guidance and stimulating discussions during the course of this work. He has always given me timely and constructive feedback for my research ideas, papers and thesis drafts. His suggestions have been helpful, and his comments and questions are very insightful. Without his consistent support, I would not have been able to complete this manuscript. I am lucky and happy to have been able to work with him during my PhD program.

I would also like to thank the Professor Christophe DI POMPEO. Professor at the Lille2 university for his accept to be the president of the jury that defences this thesis.

My sincerest thanks addressed to the Professor Mariagrazia DOTOLI, Professor Poltecnico di Bari in Italy, and to the Professor Mekki KSOURI. Professor ENIT in Tunisia. That have accepted to report this thesis. I thank them very much for the time spent for this purpose although all the responsibilities they have.

I thank the other members of the jury for accepting to be the examiners of this thesis. So, my thanks go to Professor Farouk KAMOUN, Professor ENSI in Tunisia. I would like also to thank the industrialist who accepted to discuss this work and participate in the jury M.Moncef ZOUARI, Pdg of Sevenair in Tunisia . My great thanks are addressed to my friend and my co-coordinator, Dr. Hafay Zgaya, doctor at the Lille2 University. She was and remains to be a truly inspiring example of a person who could balance extreme professionalism with care and humanity, a balance I will always aspire to.

I could not forget to thank the staff of LAGIS at th Ecole Centrale de Lille for their availability and helpfulness. My great thanks addressed to my team of research: Karama JERIBI, Manel SGHAIER, Ayda KADDOUCI and Amani DAKNOU, for their kindness, availability, their assistants and the time they devoted to me

My close friends "Heba, Maissa, Abir, Zeina ,William and Mohannad" deserve a huge 'thank you' for providing me with support, and for sharing with me the simple joys along the way.

Finally, I am deeply grateful to my family, who have always believed in me and provided love and support over all these years. I would never have achieved what I have without their encouragement. I cannot imagine completing my graduate study without their help. I am forever in their debt.

Sawsan SAAD

10 December 2010.

Lille , France

Table of Contents

Abstract	v
Résumé.....	vii
Acknowledgements	x
List of Figures	xvi
List of Tables.....	xix
List of abbreviations.....	xx
General Introduction.....	1
Chapter 1. Multi-Modal Intelligent Transport System and The Urban Mobility	5
1.1. Introduction.....	5
1.2. The Multi-Modal Transport Information Systems and The Urban Mobility	5
1.2.1. Urban Mobility.....	5
1.2.2. The Multi-Modal Transport Information Systems	6
1.3. Intelligent Transport System (ITS)	11
1.3.1. Definition of ITS	11
1.3.2. Main Components of ITS.....	12
1.3.3. ITS User Services	13
1.3.4. The Benefits of ITS in Urban Areas	16
1.3.5. The Existing Multi-Modal ITS	17
1.4. The Competitive Clusters in France.....	19
1.4.1. The Competitive Cluster Strategy	19
1.4.2. The “I-Trans” Project of The Competitive Cluster in the Nord Pas-de-Calais	20
1.4.3. The VIATIC.MOBILITY From I-Trans Project.....	22
1.5. Problem Statement	22
1.6. Summary	23
Chapter 2. Optimization and Ontology Negotiation Processes in Multi-Agent Systems	25
2.1. Introduction.....	25
2.2. Theoretical Foundations of Ontologies	25

2.2.1. Ontology Definitions from Philosophy to Artificial Intelligent.....	26
2.2.2. Ontology Components	29
2.2.3. Principal Design of Ontology	31
2.2.4. Main Languages of Ontology	31
2.2.5. Ontology Development Methodologies and Tools	42
2.3. Multi-Agent System (MAS) Overview	46
2.3.1. Agent Definitions and Architecture	46
2.3.2. Agent Communication Languages	48
2.3.3. Agent Knowledge Model	51
2.4. Using Optimization Solution for The Multi-Agent Systems in Transport Domain	52
2.4.1. Optimization Definitions.....	53
2.4.2. Optimization Methodologies.....	55
2.4.3. The Metaheuristics	57
2.4.4. The Evolutionary Algorithms in the Transport Domain	63
2.4.5. Multi-Agent System and Optimization	64
2.4.6. Towards a Distributed Architecture Based on Optimizing Agents in the Transport Domain.....	65
2.5. The Role of Ontologies in Multi-Agents Systems (MAS)	67
2.6. Semantic Heterogeneity and Interoperability in MAS	69
2.7. Ontologies and Combinations Problems	71
2.7.1. Terminology	71
2.7.2. Ontology Mismatching	72
2.7.3. Ontology Mapping, Merging and Alignment.....	75
2.8. Ontology Negotiation Protocol between the Agents Community	79
2.8.1. Negotiation Process Overview	79
2.8.2. Ontology Negotiation in Multi-Agent Systems	80
2.9. Summary	82
Chapter 3. Our Proposal: Modelling of Services to Aide Urban Mobility by Using Flexible Ontology in The Transport Domain.....	84
3.1. Introduction.....	84
3.2. Some Examples of Ontology Applications in Intelligent Transport Systems (ITS)	84
3.3. Modelling System Tasks by Using Travel Ontology for Planning Users Travels	85
3.3.1. Ontology Purpose.....	86
3.3.2. Conceptualization	87
3.3.3. Formalization	94

3.4. Services Modelling by Using Travel Ontology and The Dynamic Data Archiving	97
3.4.1. The Data Base Registration (DBR).....	98
3.4.2. The Data Base Archiving (DBA).....	98
3.4.3. Query Service.....	100
3.5. Summary	104
Chapter 4. Optimize Solution Based on Agent to Aid Urban Mobility by Using a Flexible Transport Ontology.....	106
4.1. Introduction.....	106
4.2. Multi-Agent System Architecture and The Optimize Solutions	107
4.2.1. Problem Statement	107
4.2.2. MAS Architecture	111
4.2.3. The Optimize Solutions by Scheduler Agents SAs.....	120
4.3. Ontology Mapping Negotiation Model.....	127
4.3.1. Perturbation Situation	127
4.3.2. A Negotiation Ontologies based on Knowledge Management Systems (NOKMS).....	127
4.4. Ontology Negotiation Protocols.....	133
4.4.1. Negotiation Initiators	133
4.4.2. Negotiation Participants.....	134
4.4.3. Ontology Negotiation Protocols.....	138
4.4.4. Agents Messages.....	143
4.5. Dynamic Reassigned Tasks (DRT) algorithm with Ontology Mapping Approach	146
4.5.1. Ontology Mapping Approach	146
4.5.2. The Dynamic Reassigned Tasks (DRT) algorithm	148
4.6. Summary	155
Chapter 5. Implementation and Results	156
5.1. Introduction.....	156
5.2. Implementation Programming Tools	156
5.2.1. The Multi-Agent Platform: Java Agent Development Framework(JADE)	156
5.2.2. Protégé and BeanGenerator	158
5.3. The Implementation of ISSAMU System.....	161
5.3.1. Ontology Validation And Generate Java Classes	161
5.3.2. The Implementation of The our Agents by JADE	163
5.4. The Results of Simulation.....	165
5.4.1. Services Modeling by The Flexible Ontology and Decomposition of Requests	165

5.4.2. The Optimize Solution by Scheduler Agent (SA) to Find the Work Plans.....	172
5.4.3. The Contribution of the Dynamic Data Archiving Model (DDAM)in ISSAUM	175
5.4.4. The Ontology Mapping Negotiation Process	178
5.5. Summary	191
Conclusion and Future Works.....	193
References.....	195
Appendix A	201

List of Figures

Figure 1.1. Example Of Multi-Modale Transport Network with Multi-Operator.....	8
Figure 1.2. User Service Logical Flows for Managing Traffic.....	15
Figure 1.3. The Four Keys Components of The Canada ITS.....	19
Figure 1.4. I-Trains Services.....	21
Figure 2.1. Three Interpretation Levels of an Ontology.	27
Figure 2.2. Example of Parent Ontology.....	30
Figure 2.3. Ontology Markup Languages.....	32
Figure 2.4. XML Code Example.	33
Figure 2.5. RDF Graph.....	34
Figure 2.6. RDF Code Examlpe.	34
Figure 2.7. OIL Reserch Fileds.....	36
Figure 2.8. OIL Code Example.	37
Figure 2.9. DAML+OILCode Example.	37
Figure 2.10. OWL Code Example.....	40
Figure 2.11 Berners-Lee’s Architecture of Semantic Web on XML.....	41
Figure 2.12. Agent Classification.	47
Figure 2.13. KQML Message Example	47
Figure 2.14. FIPA-ACL Message Example	50
Figure 2.15. The Conversion Performed by the JADE Support for Content Languages and Ontologies	51
Figure 2.16. Content Reference Model.....	52
Figure 2.17. The Different Research Spaces.....	54
Figure 2.18. The Different Minimum	54
Figure 2.19. The Pseudo-Code of Roulette Wheel Selection.	59
Figure 2.20. Example of Coding of the Scheduling Landings	63
Figure 2.20. Example of Coding of the Scheduling Landings	63
Figure 2.21. Single Ontology Approach.....	70
Figure 2.22. Multiple Ontologies Approach.....	70
Figure 2.23. Hybrid Ontologies Approach.	71
Figure 2.24. Ontology Mismatching.	72
Figure 2.25. Ontology Mapping & Alignment.....	76

Figure 2.26. Ontology Merging	77
Figure 3.1. General View of System Using the Ontology to Support Traveller's Planning.....	87
Figure 3.2. Agent Architecture	90
Figure 3.3. Domain Ontology	90
Figure 3.4. Action Model.....	91
Figure 3.5. Travel & Geographic Ontology	92
Figure 3.6. Transport Ontology	93
Figure 3.7. Travel Ontology	93
Figure 3.8. The Dynamic Data Archiving.....	99
Figure 3.9. RDQL Query Example.....	100
Figure 3.10. OWL-QL Query Example.....	101
Figure 3.11. SPRQL Query Example	102
Figure 3.12. Query Service and the Dynamic Data Archiving Model.....	104
Figure 4.1. Requests Decomposition	108
Figure 4.2. Nodes Identification.....	109
Figure 4.3. Nodes Assignment	109
Figure 4.4. System Architecture	112
Figure 4.5. Interface Agent Behavior.....	113
Figure 4.6. Identifier Agent Behavior.....	114
Figure 4.7. Scheduler Agent Behavior.....	115
Figure 4.8. Intelligent Collector Agent Behavior	116
Figure 4.9. Translation Agent Behavior	117
Figure 4.10. Fusion Agent Behavior	118
Figure 4.11. The Sequence Diagram of Agent Society	119
Figure 4.12. Multi-Agents Structure.....	128
Figure 4.13. Negotiation Layer (NL)	129
Figure 4.14. Semantic Layer (SEL)	130
Figure 4.15. Knowledge Management System Layer (KMSL)	131
Figure 4.16. Example of The Agent Priority	134
Figure 4.17. Example of The Agent Preference.....	135
Figure 4.18. Example of The Agent Constraint.....	135
Figure 4.19. Exam Partial View of the Transport Ontologies of both the SA and ICA Agents.....	137

Figure 4.20. Ontology Negotiation Protocol (ONP)	140
Figure 4.21. Ontology Mapping Protocol (OMP)	142
Figure 4.22. Mapping Process	147
Figure 4.23. DRTOnto Process	151
Figure 5.1. Snapshot of the RMA GUI	158
Figure 5.2. Protégé Interface	160
Figure 5.3. Bean Generator Interface	161
Figure 5.4. Ontology Hierarchy in Protégé	162
Figure 5.5. Example of BeanGerator Calss	163
Figure 5.6. The agents Communication in ISSAUM	164
Figure 5.7. The Results Interface	165
Figure 5.8. The User's Interface	166
Figure 5.9. The Result of Services Decomposition	171
Figure 5.10. The FWps of the ICA agents with the Assigned Tasks	173
Figure 5.11. The Simulation Result of Our Example	175
Figure 5.12. Example of the improvement of ISSAUM performance by using ADDM	176
Figure 5.13. The Perturbation Case	178
Figure 5.14. The Modification of the FWps after the Perturbation Case.....	179
Figure 5.15. The form of NOT_UNDERSTOOD Message	181
Figure 5.16. The Mapping Process in Our Example	183
Figure 5.17. Mapping Results	185
Figure 5.18.Negotiation Tours According to OMP	189
Figure 5.19.Negotiation Tours According to NOKMS	191

List of Tables

Table 2.1. RDF Triple Example.....	34
Table 2.2. Comparison between OWL Lite, OWL DL and OWL full.	40
Table 2.3. Optimisation Problem Classifications	55
Table 2.4. Summary of Ontology Mapping Systems	79
Table 3.1. Description Glossary.....	89
Table 3.2. OWL Constructor	94
Table 3.3. OWL Axioms	94
Table 3.4. Time and Updating Indicators for Information Classification	99
Table 4.1. Table of Tasks	108
Table 4.2. Table of Associated Concepts	108
Table 4.3. Example of Available Services(Processing Time, Cost, and Data Size) ..	111
Table 4.4. Notations	121
Table 4.5. Example of a FeTAR Instance.....	126
Table 4.6. FIPA Performative of Communicative Acts.....	139
Table 4.7. QOM: Features and Measures for Similarity	148
Table 5.1. The Table of Associated Concepts of Our Example	169
Table 5.2. Decomposition of Requests by IdA Agents	171
Table 5.3. The IWps of ICAs	172
Table 5.4. Chromosome CH (generated FeTAR instance)	174
Table 5.5. The FWps of ICAs	172
Table 5.6. Time and Updating Indicators	177
Table 5.7. The SA agent Result For Mapping Service	182
Table 5.8. The ICA agent Result For Mapping Service	183
Table 5.9. The Result of Similarity Measures	184
Table 5.10. The Concepts Memorized During a Set of Negotiations Tours	190

List of abbreviations

Abbreviation	Description
AI	Artificial Intelligence
ACL	Agent Communication Language
CIDAT	Inter-ministerial Regional Planning and Development Committee
CNP	Contract Net Protocol
CRM	Content Reference Model
Cv	Communication Vocabularies
DAI	Distributed Artificial Intelligent
DAML	Darpa Agent Markup Language
DBR	Data Base Registration
DBA	Data Base Archiving
DL	Description Logic
DOnto	Domain Ontology
DQL	DAML Query Language
DRT	Dynamic Reassigned Tasks
ETMN	Extended and distributed Transport Multimodal Network
FA	Fusion Agent
FeTAR	Flexible Tasks Assignment Representation
FIPA	Foundation for Intelligent Physical Agents
FWp	Final Workplan
GAs	Genetic Algorithms
ICA	Intelligent Collector agent
IdA	Identifier agent
IKB	Intelligent Knowledge Base
IndT	Time indicator
IndU	Updating indicator
ISSAUM	Information System of Services to Aid the Urban Mobility
ITS	Intelligent Transport System
IWp	Initial Workplan
JADE	Java Agent DEvelopment Framework
KB	Knowledge Base
KC	Knowledge Creation
KIF	Knowledge Interchange Format
KMSL	Knowledge Management Systems Layer
KQML	Knowledge Query and Manipulating Language
KR	Knowledge Retrieval
KT	Knowledge Translation
LOS	logistics optimization style
MA	Mobile Agent
MAS	Multi-Agent System
MN	Multimodal Network
MTIS	Multimodal Transport Information System
NL	Negotiation Layer
NOKMS	Negotiation Ontologies based on Knowledge Management Systems
OIL	Ontology Inference Layer
OKBC	Open Knowledge Based Connectivity

OMP	Ontology Mapping Protocol
ONP	Ontology Negotiation Protocol
QoS	Quality of Services
OWL	Web Ontology Language
OWL-QL	OWL Query Language
QOM	Quick Ontology Mapping
RDF	Resource Description Framework
RDFS	RDF Schema
RDQL	RDF Data Query Language
SA	Scheduler Agent
SA	Simulated Annealing
SD	Steepest Descent
SEL	Semantic Layer
SPARQL	Simple Protocol And RDF Query Language
SQL	Structured Query Language
TS	Tabu Search
TSP	Travelling salesman problem
TOVE	Toronto Virtual Enterprise
W3C	World Wide Web Consortium
WTO	World Tourism Organization
XML	eXtensible Markup Language

General Introduction

Transport users require relevant, interactive and instantaneous information during their travels. An Information System of Services to Aid the Urban Mobility (ISSAUM) can offer a support tool to response and help network customers to make good decisions when they are travelling by providing them all needed information in any existent and chosen format (text, multimedia...). In addition, through there are different handheld wireless devices such as PDAs, laptops, cell phones, etc.

The objective, which we aim to realize, represents a challenge compared with the continuing growth of distributed information in a large scale networks. In fact, the user's requests correspond to a set of services. Each service can be proposed differently by several information providers, in competition, with different cost, response delay, different size and different format of the data. This set of services can be demanded simultaneous by a great numbers of users, with different design measures and by taking into account the optimization methods to support the risk of crash of the information systems.

Therefore, this work aims to model, optimize and implement an Information System of Services to Aid the Urban Mobility (ISSAUM). This system is dynamic and capable to manage distributed and heterogeneous data flow. That data can be using different formats to represent the services. Moreover, information providers may be use knowledge representations (ontologies) that differ significantly either syntactically or semantically. This heterogeneity in the representation of the transport services is a critical impediment to an efficient information exchange.

The dynamic and distributed aspects of the problem incite us to adopt a multi-agent approach to ensure a continuous evolution and a pragmatic flexibility of the system. The proposed Multi-Agent System (MAS) is based on meta-heuristic for the research and the composition of the services. So, we proposed to automate the modeling of services by using the ontology idea. The ontology solution aids the information retrieval between the different transports information providers (servers) in the Extended and distributed Transport Multimodal Network (ETMN). The services research is based on the Mobile Agent paradigm (MA) using a dynamic optimisation algorithm for the MA workplans design.

Our ISSAUM takes into account the possible disturbance through the EMTN (crash, bottlenecks, etc.) in order to satisfy user requests in all the cases. For that, we developed a negotiation protocol between optimiser agents and mobile agents of the system. The proposed ontology mapping negotiation model based on the knowledge management system for supporting the semantic heterogeneity is organized as

follow: the first layer contains the *Negotiation Layer (NL)*. The second layer represents the *Semantic Layer (SEL)*, and the third layer is the *Knowledge Management Systems Layer (KMSL)* which uses ontology for the purpose of automatic classifying and using of the news ontologies and meta-ontologies. Our approach aims to make the agents able to understand each other when using these ontologies and by applying the mapping services to resolve the misunderstanding problem.

Finally, we detailed the reassignment process by using a Dynamic Reassigned Tasks (DRT) algorithm supported by ontology mapping approach. The DRT Mapping algorithm based on the current state of travelling mobile agents in their correspond routes called Workplans. Our goal is to give users all needed information even if some information providers are no longer available. Thus we improve the Quality of Services (QoS) of the response time with the best cost.

The rest of this thesis is organized in 5 chapters as follow:

1. First of all, we give an introduction about the people movements and urban mobility. We explain the different problems which are related to the multi-modal transport networks. We define also the Intelligent Transport System (ITS) with its main components, its user services, the benefit of ITS for urban areas and we present the existing ITS.
2. Chapter (2) presents an overview of the theoretical foundations of ontologies. We review the different meanings that the term ontology takes in Artificial Intelligence (AI), as well as the importance of developing ontologies. In addition to the ontology research, we present an over view of the MAS with its communication languages and its knowledge model. We explain also the optimization methods proposed for MAS and their relation with transport systems. Finally, we identify the main interoperability and heterogeneity problems found in open MAS and explain the importance of using ontologies.
3. In chapter (3), we illustrate our modeling of services approach by using flexible ontology applied in the domain of transport. In fact, we address the design and the development phases of our travel ontology (purpose, conceptualization, formalization, and validation.).
4. Chapter (4) illustrate the architecture of our system and describe the role of each agent and their interaction. We discuss also the architecture proposed to facilitate the negotiation process. We explain the process of mapping multiple ontologies by using both, the Ontology Negotiation Protocol (ONP) and Ontology Mapping Protocol (OMP). Finally, we explain the reassignment tasks process by using Dynamic Reassigned Tasks (DRT) algorithm supported by ontology mapping approach.

5. In Chapter 5, we explain the implementation and present some experiments with different scenarios in the transport domain.

Publications List:

Some of the results presented in this thesis have already been published in various articles and reports, which are listed below:

- **S.Saad, H.Zgaya, S.Hammadi:** *"The Flexible Negotiation Ontology-based Knowledge Management System: The Transport Ontology Case Study"*, In proceedings of the IEEE, the International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA'08). April 7 - 11, 2008 Damas, Syrie.
- **S.Saad, H.Zgaya, S.Hammadi.** *"Une Ontologie de négociation flexible basée sur un système de gestion de connaissances : Cas d'étude d'Une Ontologie dans le domaine de Transport"*. Le congré de la Recherche Opérationnelle et de l'Aide à la Décision sont éligibles pour la conférence (ROADEF'08). Februry, 25-27,2008 , à Clermont-Ferrand, France
- **S.Saad, H.Zgaya, S.Hammadi.** *"Using Ontology to Solve the Negotiation Problems in Mobile Agent Information Systems"*, In proceedings of the IEEE, the International Conference on Systems, Man, and Cybernetics (SMC'08). October 12-15, 2008, Singapore.
- **S.Saad, H.Zgaya, H.Hammadi:** *"Novel Ontology Model for Communicating Heterogeneous Negotiation Mobile-Agent in a Transport Environment"*, Studies in Informatics and Control Journal (SIC), National Institute for R&D in Informatics ICI Bucharest, Volume 17 - Num 4, page (333-352) , December 2008.
- **S.Saad, H.Zgaya, H.Hammadi.** *"Dynamic Reassigned Tasks during the Negotiation Process by Ontology Approach between Mobile Agents"*, In proceedings of the IEEE, International Conference on Intelligent Agent Technology (IAT-08). Decembre 9-12 Sydney,2008, Australia.

- **S.Saad, H.Zgaya, H.Hammadi.**“*Knowledge Management - integral part of information and communications technology*”. Chapter:” Towards the Optimization of Client Transport Services: Negotiating by Ontology Mapping Approach between Mobile Agents”.*page (195-220)*. April 2010; isbn: 978-953-7619-94-7.

Chapter 1. Multi-Modal Intelligent Transport System and The Urban Mobility

1.1. Introduction

Our society becomes more and more mobile. Every day, we spend time in transportation. Our displacements are different according to our activities (work, hobbies...). Based on these displacements; there are many modes of transport available, such as cars, trains, ships, airplanes, subways and buses.

However, most of transport services are independently operated and managed. The multi-modal transport systems, such as an integrated network between buses and trains or ships and trains, are helping countries respond to this growing demand of multi-modal transport services. The multi-modal transport systems also provide the opportunity to manage the transport chain through the integration of all modes of transport under a single transport document. Aided by the revolution in information technology, multi-modal transport operators in Canada, America, Europe and a few other countries have applied the variety of transport's useful information systems and services. These systems are becoming available daily to assist travellers, and can provide an important contribution in an intelligent mobility which meets future requirements, and afford new economic opportunities to the transport technology sector. This is the goal of this thesis to aid the travellers in their urban mobility by using an Intelligent Transport System (ITS).

This chapter is organized as follows. In section 1.2, we address the multi-modal transport information systems and the urban mobility where we present their definitions and the problems related to multi-modal transport networks. In section 1.3 we define the Intelligent Transport System ITS with its main components, its user services, the benefit of ITS for urban areas and we present the existing ITS. In section 1.4, we explain the competitive clusters in France which is the research project for our work group. Finally, the conclusion of the first chapter will be in section 1.5.

1.2. The Multi-Modal Transport Information Systems and The Urban Mobility

1.2.1. Urban Mobility

Mobility is a concept which is studied in many research areas such as astronomy, sociology, physics, and so on. Its meaning is different according to studied systems. For example, computer scientists focus on mobile computing or mobile computation whereas geographers are interested in population movements. In addition to that, there are several methods to describe mobility in a same research area. A lot of works divide mobility into four main parts, [Marilleau, 05]: daily mobilities, travels, residential mobilities and migrations.

Mobility study which is interest of us is the urban mobility. Urban movement study aims to looking and designing human displacements in an urban environment by different transport modes and for different reasons, from a departure to a destination. The concept of daily urban mobility goes beyond the notion of travel with its causes and consequences. Daily mobilities include displacements which are done during our current life. Every day, we move in order to achieve some tasks. For example, every day, we go to work, to supermarket..., so, we move because of our goals.

In fact, daily urban mobility based on the location in transport networks, the quality of the transport services, the number of automobiles, the number of holders of driving licenses for the household and the possible use of a vehicle for each person. An approximation of the level of sociability may be assessed by the rate of people moving, the number of trips per person per day and the number of activities performed outside the home. The number of movement per travel gives an indication of the complexity of programs which needs to adapt to the situations of difficult mobility.

1.2.2. The Multi-Modal Transport Information Systems

We presented in previous section the definition of the urban mobility and now we explain the definition of multi-modal transport information systems with their problems in the urban area.

1.2.2.1. *Definition*

A mode of transport may be provided by several operators. So, a given route can be inter- or multi-operator, and therefore used in different ways by one or more operators. This mode arises the definition of “*Multi-Modality*” which is use several transport modes (ex. bus, metro, tram, trains, TGV, plain, etc.) to achieve the travellers needs. This explains why there is a set of transport modes offering connections between a set of origins and destinations. One of these connections is called “*Inter-modal*” which is the movements of passengers or freight from one

mode of transport to another in one travel. Although inter-modal transportation is possible, it does not necessarily occur. The “*Multi-Modal Information*” can be an effective mean for the orientation of the traveller and the development of collective transports. This information produces the best role between the modes of transport. In this way, the multi-modal information is apparently one among many means to achieve this ideal goal. Finally, we have to define the “*Multi-Modal Information Services*” which are distributed through various channels and supports to assist travelers in their urban daily movements (box office, web, phone, PDA, etc.). These services may correspond to transport services (ex. routing, travel times, information on disruptions and delays, etc) as well as the services related to transport (lodgment reservations, cultural events, weather, tourist information, etc), [Zgaya ,07], [Zidi, 06].

1.2.2.2. *The Problem of Multi-Modal Transport Networks*

A multi-modal transportation network extended over a large geographical area which can to be a combination of several subnets. The public transport network in France consists of a national network station with several regional transport networks linked by exchange cluster of the SNCF¹, representing the common stations. A regional network may also consist of several departmental networks and a department network, in turn, may also join several networks. At the level of departments, the network connections are through a combination of buses and regional TER. In all cases, the operator (private or public) has the information of the multi-modal transportation network that it operates, and the generation of multi-modal information on a wide geographic area involves the different operators of transport networks who cover at least part of the area.

For example, the figure.1.1 illustrates the public multi-modal transport network in the *Nord-Pas de Calais* region in France. The agglomeration *Lille* and *Valenciennes* towns that each has own network of multi-modal public transport (bus, metro, tram). These two agglomerations are linked to the regional network TER of SNCF via exchange clusters (SNCF stations), which are also stops of bus, tram or metro for two local networks. Each operator has the information about the multi-modal transportation system which it operates. Thus, to produce a multi-modal information system on a wide geographical area, it is necessary to involve different operators that the transport networks cover, at least in part, the area concerned.

¹ [http:// www.sncf.com/](http://www.sncf.com/)

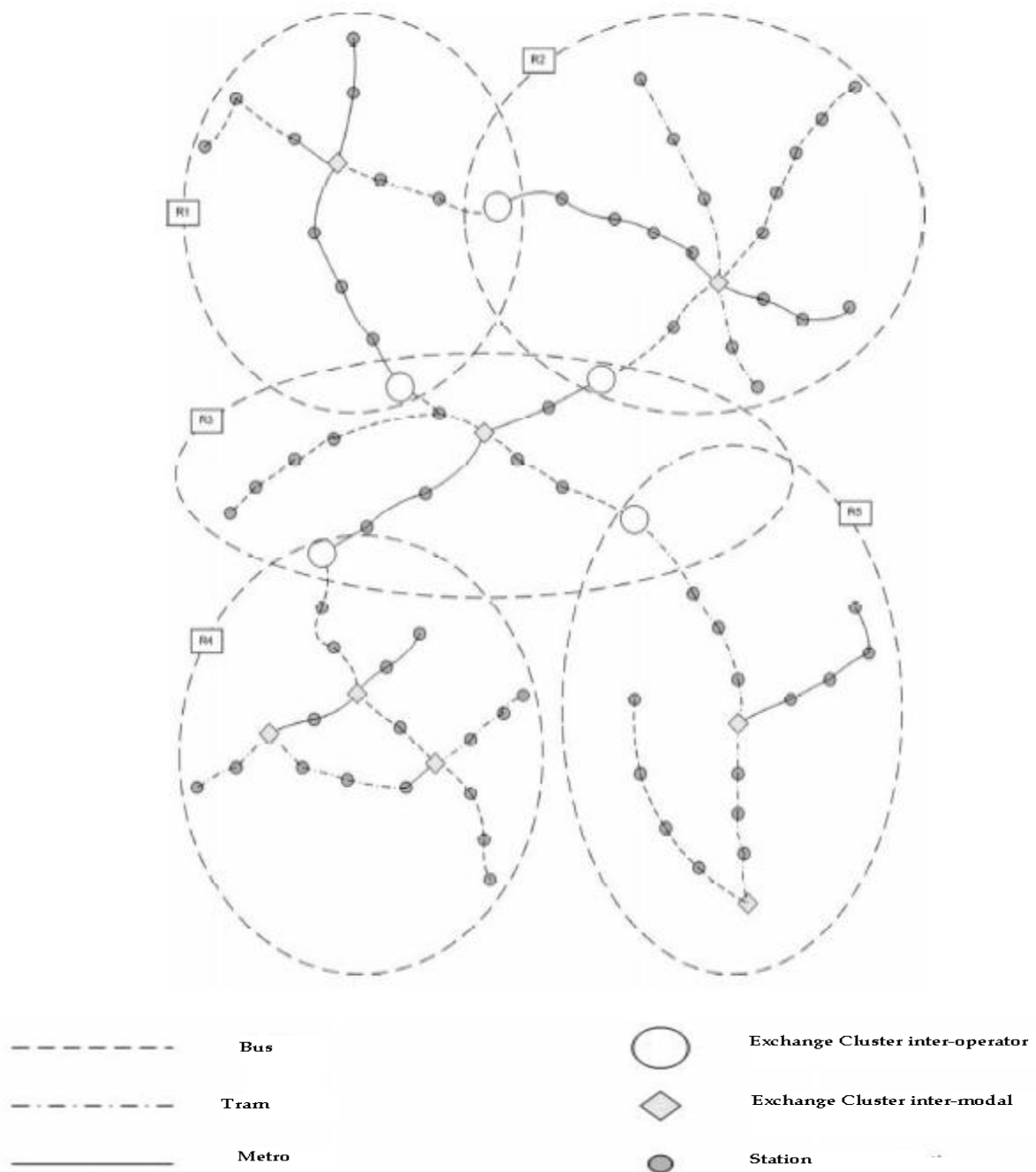


Figure 1.1 Example of Multi-Modal Transport Network with Multi-Operator

Indeed, in our team of logistics optimization style (LOS) at LAGIS laboratory, the studies related to multi-modal transport networks can be classified into two categories *Regulator-Side Transport* and *Client-Side Transport*. The former represents the axe of traffic control system research which support by the decision making systems. The second one represents the traveller's information systems by modelling, implementation and optimization of multi-modal transport information system to best serve the transport customers before and during their travel. We illustrate the two sides in the following:

Regulator-Side Transport:

The domain of transport is very large to cover all different research axes. In fact, the transport companies try to find a match between the offers and the requests by respecting the Quality of Services (QoS). In this sense, an important aspect is to control this request and put it in the transport circuit. Now, this step is not happening without problems related by the complexity of managing a fleet of vehicles, with different modes and on all of the transportation networks systems. Thus, this is the role of the regulator which based on to solve the real-time problems, to propose the strategies when there are the networks perturbations by applying the suitable modification.

In all the urban transport networks, the use of a line consists of two distinct phases. The former is the design and the development phase of the production program. The second is the adaptation phase of the production program where in this phase applied the real exploitation conditions.

Fayeche [Fayeche et al., 02] presented a decision making system for regulating transport system which aim to solve the problem of offers modification in a real exploitation conditions. Thus, they found a solution to help the regulators in various assigned tasks to them. The proposed system performs the traffic surveillance and the accidents detection. This system is based on the agent idea for controlling and regulating (in the familiar situations) and evolutionary approach for temporal regulating or spatial-temporal (in the not familiar situations).

In another work, [Ould Sidi,...] developed a decision support module for the generation and evaluation of regulation strategies. Their approach is based on the evolutionary algorithms. Those algorithms help them in find the close solutions to the optimal one for the NP problems. Their system has been integrated in the regulators workstations where it is easy to use. The impact of this system on travelers is to minimize their waiting time and to ensure the continuity of their movements in the multimodal transportation system. All that, for improving the quality of services provided to passengers.

Finally,[Melki, 08] has provided a tool of decision making for multi-modal transport regulators ,which helps in the choice of operating strategy, to adopt in the presence of simultaneous perturbations in the network.

Client-Side Transport:

The research in this side is to provide the transport clients all the necessary information related to his travel. The goal is to serve the traveller when he looks for the information's which has become harder to find and to optimize them. All that, because of, the growth in the volume of data information networks which becomes more widespread.

Recently, the research, in this domain, interests in providing the traveller by all the necessary information from his point of departure to his destination. And it must offer to him the varied services, which affect not only transport services but also the services related transport. For this side, a comparison between centralized management and distributed management is realized by Zargayouna [Zargayouna, 03] which was focused on traveller information systems.

A work has been done for optimization of routes by [Kamoun, 07] [Zidi, 06], without covering the design and optimization of an information system assistance of urban mobility in every sense of the term. This means that the system aids the traveller with generic optimized information. In this sense too, [Zgaya, 07] presented, optimized and implemented a Multimodal Transport Information System (MTIS) to optimize the management of the data flow of the users' requests, which can be simultaneous and numerous. The MTIS is related to an Extended and distributed Transport Multimodal Network (ETMN) which contains several heterogeneous data sources including the different proposed services to transport users. In the same context, this thesis, which is the continuous of the Zgaya's thesis, tries to provide the optimized solutions for heterogeneous data when there are perturbations in the transport networks to facilitate the urban mobility by using intelligent methods.

Finally, after having seen the multi-modal transport information systems and the urban mobility where we have presented their definitions and the problems related to multi-modal transport networks. We saw that the big cities suffer from traffic congestion worldwide and all consequences that come with it. There is no certain solution for this problem, but several improvements have been suggested in the field of urban and traffic management, provided an information system which can provide information to both the traffic experts and the user of the system. Such an information system has to incorporate features of an Intelligent Transport System (ITS) as we will present in next section. This thesis discusses the model and

implementation of such an information system which is based on a model of simulation and implemented by using agent technologies.

1.3. Intelligent Transport System (ITS)

Transportation System is a complex, large scale, and integrated open system. It is complex because it involves multi stake-holders and comprises different infrastructure facilities as well as multi-modes operating in different spectrum of regulating environment. It is an integrated open system because it allows the addition of sub-systems into its operating space. All these sub systems are inter-related and loosely 'integrated', not in a positive sense but, in a situation that they are mutually affecting one another. It is no longer acceptable for a transportation agency to develop a system without worrying about interfaces with other functions (police, public transport operators, traffic signal systems operators i.e., local authorities).

In this section we define the Intelligent Transport System as well as we explain its components, its user services, its benefits in urban area and we present some example about existing ITSs.

1.3.1. Definition of ITS

An Intelligent Transport System (ITS) contains the application of technology such as communication systems, computers, electronics, and information technology to improve the efficiency and safety of the transportation network. ITS is utilizing cooperation technologies and system engineering concepts to develop and improve transportation systems of all kinds.

A number of definitions of ITS has been advanced. For example, according to Transport Canada [Canada, 03]:

"ITS, refers to the integrated application of information processing, communications, and sensor technologies, to transportation infrastructure and operations. These systems bring together users, vehicles and infrastructure into a dynamic relationship of information exchange, resulting in better management strategies and more efficient use of available resources".

Another definition, as defined by the United States Department of Transportation, [United States, 03]:

"Intelligent Transportation Systems (ITS) apply well-established technologies in communications, control, electronics and computer (hardware and software) to improve surface transportation system performance."

The most ITS activities are devised in four categories of technologies:

- **Sensing:** The ability to note the position and speed of vehicles using the infrastructure (e.g. rail lines, roadways, bridges, tunnels).
- **Communicating:** The ability to send and receive information, between vehicles, between vehicles and infrastructure, and between infrastructure and centralized transportation operations and management centers.
- **Computing:** The ability to process large amounts of data collected and communicated so that conclusions can be drawn and assessments made.
- **Algorithms:** Computer programs which process information gathered by ITS and develop operating strategies for transportation facilities.

As we mentioned previously that we try to provide the optimized solutions where there are the perturbations in the transport networks to facilitate the urban mobility. This axe belong to the algorithms category in the ITS activities which try to find the optimized solution by processing the traveller information.

1.3.2. Main Components of ITS

The ITS have six main components, as follow, [Smadi and Miner, 05]:

- **Safety:** ITS can help to reduce injuries and save lives, time and money by making transport safer. It helps the driver by providing him all the necessary information (ex. about the state of the road, the traffic and the weather). ITS can detect the crashes that do occur, and help emergency management services provide assistance.
- **Security:** In security aspects, ITS can help to prepare for prevent and respond to disaster situations. ITS can also help to keep watch over transport facilities and can help to provide personal security for people using the public transport system. Moreover, ITS can help identify the best routes for evacuating people at risk and directing emergency services to incidents and disaster sites.
- **Efficiency / Economy:** The ITS can save time and money for travelers and the freight industry, because the ITS can deliver fast, accurate and complete travel information to help travelers decide whether to make a trip, when to start, and what travel modes to use. The ITS can also provide information both prior to a trip and as the trip proceeds.

- ***Mobility and Access:*** ITS provides travel opportunities and traditional travel choices for more people in more ways, whether they live, work and place regardless of age or disability. Travellers can plan and take trips that use the best and more convenient combination of travel modes such as private car, public transport, etc. ITS can also help to make it easier to pay for transport services. ITS can convey the needs and interest of transport system customers to the people who manage the system customers, helping to ensure a transport system that is responsive to those needs and interests. Management of transport system for safer services and simultaneously available for motorists, cyclists, pedestrians, and users of public transport can be gained with ITS.
- ***The Environment:*** ITS can help make the transport faster and smoother, eliminate unnecessary travel, and reduce time caught in traffic congestions. This is because, the ITS can help keep traffic flowing in urban freeways, on toll roads at commercial vehicle checkpoints and elsewhere. Reducing delays due to congestions and incidents and the pollution. In this context, ITS can help vehicles operate more efficiently. ITS can provide location specific information about the weather and road conditions.
- ***A Transport System for All:*** ITS affects the way everyone where he lives, works and plays and its benefits will increase in the future and will help make transport services available and affordable for everyone, getting people and goods to their destinations safely and efficiently. Customer satisfaction can be improved with transport and it helps to make it more environmentally friendly and more secure. People can manage and operate the transport system to provide better, safer and more responsive service to all its users while helping to safeguard the environment. ITS has been demonstrating its value for over 10years and is now beginning to mature and meet its promise to make real difference to society as a whole.

1.3.3. ITS User Services

ITS user services are defined, not along lines of common technologies, but rather by how they meet the safety, mobility, comfort and other needs of transportation users and providers. These services are discussed of the major ITS user service and they cover. We summariz these services as follow, [Smadi and Miner, 05]:

- **Arterial Management Systems** manage traffic along arterial roadways, employing traffic detectors, traffic signals, and various means of communicating information to travellers

- *Transit Management Systems*, transit ITS services include surveillance and communications which enable transit agencies to improve the operational efficiency, safety, and security of the nation's public transportation systems.
- *Emergency Management Systems*, ITS applications in emergency management include hazardous materials management, the deployment of emergency medical services, and (large-/small-) scale emergency response.
- *Freeway Management Systems* employ traffic detectors, surveillance cameras, and other means of monitoring traffic flow on freeways to support the implementation of traffic management strategies.
- *Incident Management Systems* can reduce the effects of incident-related congestion by decreasing the time to detect incidents, the time for responding vehicles to arrive, and the time required for traffic to return to normal conditions. Incident management systems make use of a variety of surveillance technologies, often shared with freeway and arterial management systems to facilitate coordinated response to incidents.
- *Electronic Payment Systems* employ various communication and electronic technologies to facilitate commerce between travelers and transportation agencies.
- *Traveller Information* use a variety of technologies, including Internet websites, telephone hotlines, as well as television and radio, to allow users to make more informed decisions regarding trip departures, routes, and mode of travel.
- *Information Management* supports the archiving and retrieval of data generated by other ITS applications. Decision support systems, predictive information, and performance monitoring are some ITS applications enabled by ITS information management. In addition, ITS information management systems can assist in transportation planning, research, and safety management.
- *Road Weather Management*, ITS applications assist with the monitoring and forecasting of roadway and atmospheric conditions, dissemination of weather-related information to travellers, weather-related traffic control measures such as variable speed limits, and both fixed and mobile winter maintenance activities.
- *Roadway Operations & Maintenance*, ITS applications monitor, analyze, and disseminate roadway and infrastructure data for operational, maintenance, and managerial uses. ITS can help secure the safety of workers and travelers

in a work zone while facilitating traffic flow through and around the construction area.

- **Commercial Vehicle Operations**, ITS applications are designed to enhance communication between motor carriers and regulatory agencies.
- **Driver Assistance Systems**, Numerous intelligent vehicle technologies exist to assist the driver in operating the vehicle safely. Systems are available to aid with navigation, while others are intended to facilitate safe driving during adverse conditions. Other systems assist with difficult driving tasks such as transit and commercial vehicle docking.
- **Collision Notification Systems**: In an effort to improve response times and save lives, collision notification systems have been designed to detect and report the location and severity of incidents to agencies and services responsible for coordinating appropriate emergency response actions.

User services represent what the system will do from the perspective of the user, who may be (a motorist, a transit rider, a system operator, etc). For that, a number of functions are required to accomplish each user service. To reflect this, each of the user services was broken down into successively more detailed functional statements called user service requirements. The logical architecture defines a set of functions and information flows that respond to the selected user service requirements. For example, figure 1.2 below describes the user service logical flows for the function of managing traffic.

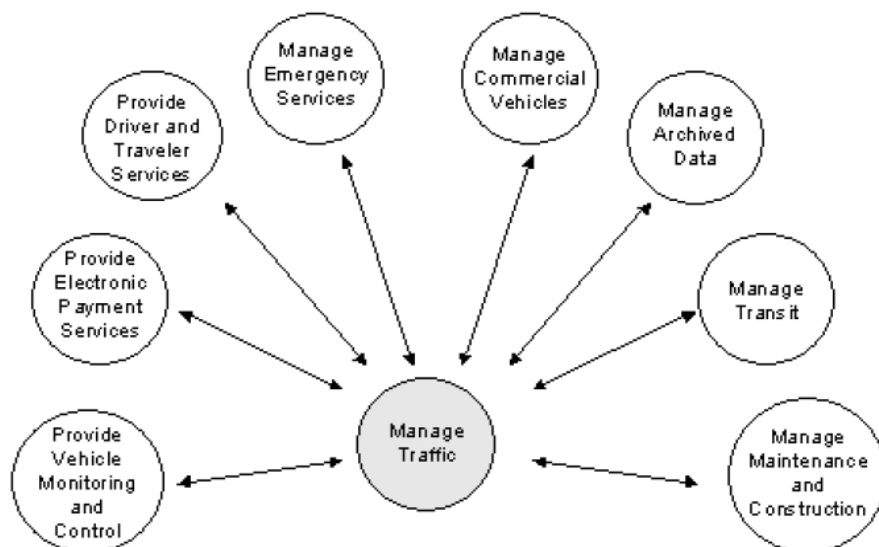


Figure 1.2 User Service Logical Flows for Managing Traffic

1.3.4. The Benefits of ITS in Urban Areas

ITSs provide a new set of tools for achieving urban local transport policies. These systems provide services using modern computing and communications technologies. The systems collect information about the current state of the transport network, process that information, and either directly manage the network, or allow people to decide how best to use the network. ITS can deliver noticeable economic benefits through reduced journey times and increased journey time reliability, as well as improvements in safety and reductions in pollution. The benefits of using ITS include:

- Making travel more efficient (safer, less polluting, cheaper, better informed travel);
- Helping to achieve 'Best Value' within network management as a result of greater information gathering and improved decision making;
- Simplifying public transport use by providing accurate real time information about services;
- Reducing the effects of pollution from vehicles by better traffic management;
- Reducing the number of accidents by providing drivers with more information about conditions on the roads they are using;
- Helping drivers find the best route to their destination, and changing that route if major incidents occur on it;
- Improving the security of public transport passengers and staff by providing extra communications;
- Helping to monitor and evaluate network performance by automatically collecting and analyzing data;
- Protecting vulnerable road users such as children and the disabled;
- Improving planning decisions by making more historic information available;
- Integrating different systems by providing a mechanism for sharing information between systems and co-coordinating strategy across different organizations;

An important benefit of ITS that it can be developed/modified relatively easily.

1.3.5. The Existing Multi-Modal ITS

The majority of the implemented transport's information systems are mono-modal, dealing with only one means of transport. Other information systems are multi-modal, concerning several means of transport, but they always concern one transport operator. Since each one belongs to a single transport operator, these last mentioned information systems still geographically limited.

In this part, we present some examples of ITS in France, Europe and in the international level to identify the characteristics of these systems and their applications. We therefore examined the various reports provided by departments and the different web sites bearing the label ITS.

ITS France²

The ITS France refers to the ITS that is the applications of the information technologies, the transport communications and the transport modes. The researchers define the applications that they are mainly related to the land transport interfaces with maritime and air transport. Therefore, they confirmed that challenge in improving ITS is concerned by the management of multi-modal transport networks as well as they relate to different aspects : security, traveler information and the detection of perturbation.

The multi-nationality of ITS involves the need for harmonization in the various countries for the mobility of people and goods. For that in 1994, the ITS France has organized the first international congress held in Paris between the French Ministry of transport and ERTICO (European Road Telematic Implementation Coordination Organization).

In France there are two intelligent transport systems which are ITS France and I-Trans. Here, we have summarized the first system, but for the second one, which our research group participate in the part of it, we will illustrate it in the section (2.4).

ITS Europe (ERTICO)³

ERTICO is the network of Intelligent Transport Systems and Services stakeholders in Europe. It consists of (the public authorities, industry players, infrastructure operators, users, national ITS associations and other organizations). Its objective is to facilitate the safe and comfortable mobility of people and goods in Europe through the widespread deployment of ITS.

² <http://www.atec-itsfrance.net/>

³ <http://www.ertico.com/>

ERTICO tries to make the European network more safety, secure, reliable and comfortable; this for the traveler but for the shipments is automatically directed to available parking spaces in urban areas. In brief, ERTICO tries to make a world in which people and goods are connected by the necessary information

The vision of ERTICO is to bring intelligence into mobility, working together in public/private partnership towards **(zero accidents, zero delays, reduced impact on the environment, fully informed people)** where services are affordable and seamless, privacy is respected and security is ensured. To achieve this vision, all modes of transport should cooperate to achieve an optimal and sustainable use of all transport modes.

International ITSs

We present the Canada ITS⁴ as an example of the international ITSs.

Over the last twenty years, the minister of the transport has outlined a framework to bring Canada's transportation system into the 21st Century that encourages the best use of all modes of transportation and builds on partnerships with all jurisdictions and with all players in the transportation sector. This framework aims to make Canada's transportation safe, efficient, affordable, integrated and environmentally friendly.

In order to better understand how ITS is changing the transportation system, and how technologies can be used to accomplish this, the researchers decided to use the potential functions of ITS within each of the four key components of the system: the vehicle, the user, the infrastructure and the communication system.

The Vehicle: the system allows the vehicle to be located, identified, assessed and controlled. The objective is to successful fleet management and to providing in-vehicle navigation and routing advice. As well as improve the safety and efficiency of the transportation system.

The User: can use the different services of the ITS where it offers navigation, provision of traveler information and monitoring capability to system users. The ITS also monitors the driver performance and conditions. In order to detect fatigue, inattention, or other circumstances that might otherwise result in an accident, could help to provide a safer and more comfortable environment.

The Infrastructure: ITS provides monitoring, detection, response, control and administration functions to the Infrastructure.

⁴ <http://www.itscanada.ca/>

The Communications System: provides the ability to exchange information between the above three functions in the system to allow for the gathering of data that can be processed into intelligence, and can then be used to determine and activate appropriate command and control actions.

The figure1.3 explains the relationship between the four key components of the Canada ITS.

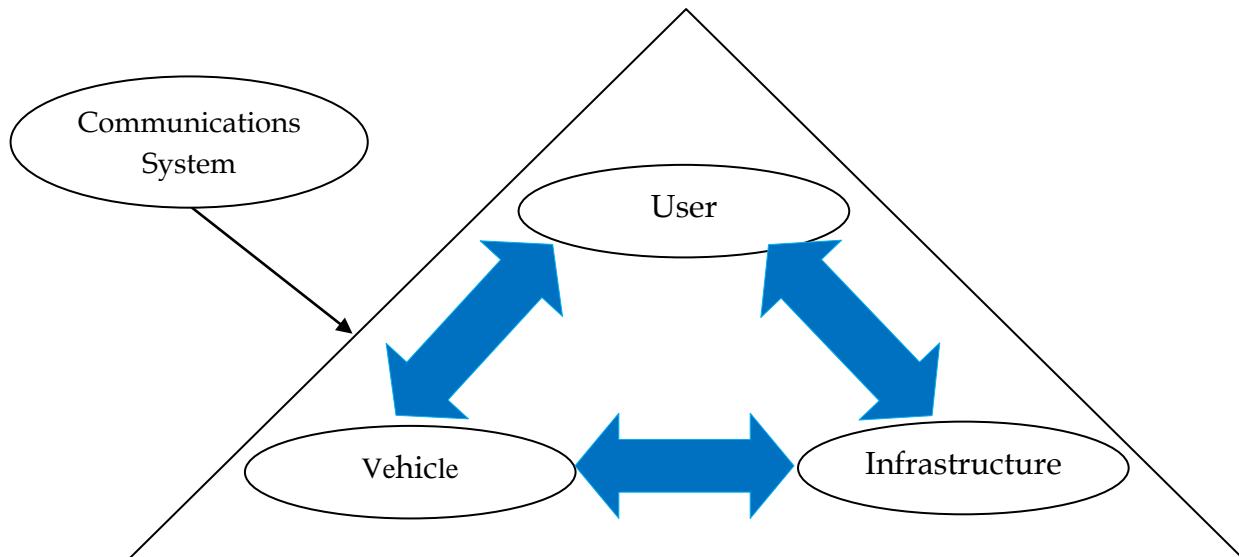


Figure1.3. The Four Key Components of the Canada ITS.

1.4. The Competitive Clusters in France

Changes in the international economic environment have prompted France to launch a large-scale industrial policy to promote and develop key elements of its industrial competitiveness. The Interministerial Regional Planning and Development Committee (CIDAT) meeting on 13 December 2002 decided on **a new industrial strategy based on the development of competitive clusters** to increase French industrial potential and create the conditions to foster the emergence of new activities with a high international profile. This strategy is based on active partnership between manufacturers, research centers and training bodies.

“A competitive cluster (or pôles de compétitivité) is defined as a geographical concentration of businesses, training centres and (public, private) research units working in partnership on innovative projects.”

1.4.1. The Competitive Cluster Strategy

Competitive clusters hence use a partnership approach and define joint, innovative projects. This **strategy** was developed by the CIADT meeting of 14 September 2004. It meets three main aims:

- To increase the economy's growth potential;
- To more effectively combat deindustrialization and relocations;
- To maintain technological expertise at the highest level.

The development strategy for competitive clusters also takes place in the European level in keeping with the goal set at the Lisbon Summit in 2000 to make Europe the most competitive economy in the world.

Following a call for projects launched on 2 December 2004, the government published **a list of the economic development projects** granted the "competitive cluster" label by the CIDAT meeting of 12 July 2005. The selected projects cover both emerging technological fields such as nanotechnologies, BI technologies⁵ and microelectronics and existing areas such as motor vehicles and aeronautics.

Each approved cluster will be individually monitored by a co-ordination committee set up by the region's prefect. This committee will be responsible for drawing up a framework contract to define the cluster's operations, strategy, research priorities and performance evaluation criteria.

There are two main types of clusters:

- Very high-tech clusters with an international profile
- Clusters built on a "classic" industrial basis in French areas of specialization or development.

1.4.2. The "I-Trans" Project of The Competitive Cluster in the Nord Pas-de-Calais

In section (2.3.5), we have summarized the ITS in France and we said that they are two ITS which are ITS-France and I-Trains. Here we will illustrate the I-Trans project.

The **I-Trans**⁶ competitiveness cluster, operating under the care of the **Transports Terrestres Promotion Northern France** association, brings together industry leaders and key players in research and education relating to rail, automobiles, logistics and intelligent transport systems in the **Nord-Pas de Calais and Picardy** regions of Northern France (figure 1.4). Together, they aim to build international recognition for Northern France as a unique focus of excellence and innovation in terrestrial transport (Railways in the heart of Innovative Transport Systems). I-Trans is **one of the 16 competitiveness clusters in France** that have been officially recognized for

⁵ <http://www.i-trans.org/>

⁶ <http://www.bitechnologies.com/>

their potential as centres of scientific and technological know-how with worldwide reach. The strategy of I-Trains contains 4 “I”:

- Intermodality
- Interoperability
- Intelligent
- Innovation



Figure 1.4. I-Trains Services

Y. Ravalard is the Project Manager of 7 Programs committees:

- CP 1: Formation and Expertise
- CP 2: Basic Research
- CP 3: Railways Interoperability
- CP 4: SAEE (Safety and Acoustics of Onboard Equipments)
- CP 5: Freight Intermodality
- CP 6: Passengers Intermodality
- CP 7: Urban Guided Transports

1.4.3. The VIATIC.MOBILITY⁷ From I-Trans Project

Many information systems dealing with mobility coexist on Lille metropolitan area, and on a larger scale in Nord-Pas de Calais Region. They propose many kind of information, such as:

- Information on public transport supply in all major towns, intercity train and bus transports, ...
- Information on traffic conditions on Lille major urban road and motorway networks,
- Dynamic information on public car parks capacity.

Information sources related to those various kinds of information are provided by numerous operators and located in numerous places: public transport station (bus, metro, train...), variable message signs on motorway and urban network, radio, internet, etc.... Therefore global information collect is difficult for users, especially for occasional travellers.

INRETS, as a member of the I-TRANS competitiveness cluster, is supporting the concept of VIATIC. The VIATIC project (I-Trans CP6) is accompanying *intelligent mobility*. The aim is to design the future services for mobility by developing the technical elements for experimentation in a TER (regional express transport) carriage, and making travel information available to the traveller during his journey (multi-modal information) and information about amenities during his trip (news, culture, entertainment, tourism, games, ...) This information is accessible near transport systems and via on-board systems.

1.5. Problem Statement

The research tasks presented in this thesis continue the research of Zgaya's thesis, [Zgaya, 07], which has belonged to the French national project VIATIC.MOBILITE. Indeed, we can imagine infinity of innovating services related to mobility, not only intended for general public, but also for companies, to a relevant control of the mobility for their displacement plans. The main goal of our work is to provide a system with a mobility assistance, which is related to the daily or occasional traffic motivations, tourism and culture occupations, etc. with a possibility to profit from relevant and exploitable information.

⁷ <http://viatic.inrets.fr/>

In the Zgaya's thesis, she has proposed a work which design, optimize and implement a Multimodal Transport Information System (MTIS) to optimize the management of the data flow of the users' requests, which can be simultaneous and numerous. The MTIS firstly decompose each set of simultaneous requests into a set of independent sub-requests called tasks. Each task corresponds to a service which can be proposed, differently, by several information providers. In her research, she has used a manual method to decompose the requests. The MTIS is related to an Extended and distributed Transport Multimodal Network (ETMN) which contains several heterogeneous data sources including the different proposed services to transport users. In case of disturbance through the EMTN (crash, bottlenecks, etc.). In order to satisfy user requests in all the cases, she has designed a negotiation protocol without using ontology between mobile agents. Her protocol suffers from many of problems in the implementation phase.

In this thesis, we automate the services modelling by using the ontology idea. Because this solution aids the information retrieval between the different transports information providers (servers) in our EMTN. Indeed, our system aims to support the transport users in planning their travels. In this context, the user defines (the departure city, the arrival city, the date, etc...) of his travel. With this information, our system looks for all trips possibilities in the different servers by using his ontology.

In addition, we propose an approach that will improve the negotiation process in the mobile agent systems by using different ontologies. We use the negotiation protocol which Zgaya's proposed. According to this protocol, we present an ontology solution based on the knowledge management system for semantic heterogeneity. The proposed solution facilitates, automates the communications and prevents the misunderstanding during the negotiation through the agents' communications. Our approach aims to make the agents able to understand each other when using these ontologies by applying the ontology mapping services.

1.6. Summary

Urban mobility of people displacements in everyday life can be perfectly framed by multi-modal transport information. Therefore, the transport information systems which support the urban mobility are very important, if they offer all the useful transport information.

However, the most existing transport information systems are developed by using the algorithms to calculate the route for proposing the best transport offers depending on the user's needs. This is already an important progress but the urban

mobility of people beyond the simple idea of moving. We propose therefore in this thesis a multi-modal information system which aide the urban mobility. This system provides the transport services (ex. routing, travel times, information on disruptions and delays, etc) in addition any services related transport (cultural events, weather, tourist information, etc). The proposed system is based on a multi-agent system to optimize the management of the data flow of the users' requests, which can be simultaneous and numerous. The system is related to an Extended and distributed Transport Multimodal Network (ETMN) which contains several heterogeneous data sources including the different proposed services to transport users. In case of disturbance through the EMTN (crash, bottlenecks, etc.). In order to satisfy user requests in all the cases and to achieve efficient interoperability between information systems. The ontologies solution plays an important role in resolving semantic heterogeneity and applying the negotiation process between system providers. In the next chapters, we present the state of the art about the theoretical overview of the ontologies domain and negotiation ontology between the agents.

Chapter 2. Optimization and Ontology Negotiation Processes in Multi-Agent Systems

2.1. Introduction

In a Multi-Agent System (MAS) environment, when the heterogeneous agents want to engage in communication. They may have the problem of not understanding each other, unless they share some content language ontology. This chapter is focused on the roles of ontologies and the optimization algorithms in open MAS. Indeed, when agents have private ontologies, it is necessary to provide some bidirectional translation between them to optimize the communication time and to solve the heterogeneity and interoperability problems, which block the communication between agents in open MAS. Ontology negotiation also offers an integrated approach that enables agents to gradually build towards a semantically integrated system by sharing parts of their ontologies.

In this chapter, firstly we present the theoretical foundations about the features of ontologies in (section 2.2). Then, an overview on the definition of multi-agent system and the communication languages which used between the agents to understand each other will be explained in (section 2.3). The use of optimized solution in the MAS and the illustration of different optimized algorithms used in transport domain explain in (section 2.4). The role of ontology in MAS will be illustrated in (section 2.5). The semantic heterogeneity and interoperability problems which effected the communication in open MAS are explained and the operations involving ontologies are presented (section 2.6). Those communication problems should be solved, rather than avoided. For that, ontologies and combinations problems will be discussed in (section 2.7). Ontology negotiation and different approaches which presented to enable a good conversation between the agents will be presented and analyzed in (section 2.8). Finally, the conclusion of the second chapter is in (section 2.9).

2.2. Theoretical Foundations of Ontologies

Research on ontology has evolved in the last decades in the computer science community. While the term of ontology has been rather confined to the philosophical sphere in the past, it is now gaining a specific role in Artificial Intelligence (AI), Computational Linguistics, and Database Theory. In particular, its importance is being recognized in research fields as diverse as knowledge engineering, knowledge representation, qualitative modelling, database design, and

information (modelling, integration, retrieval and extraction). Nowadays, ontology is used for multi-agent systems, natural language translation, medicine, electronic commerce and the newest areas of interest, in the Semantic Web domain.

In this section we present a theoretical overview of the ontologies domain; we start by some definitions of ontologies from philosophy to artificial intelligent in the section (2.2.1). Then, we present the components of existing ontology (section 2.2.2). We address the principal design phases of ontology in section 2.2.3. We refer to the main ontology languages in section 2.2.4. Finally, the principal ontology issue about ontology development methodologies and tools will explain in section 2.2.5.

2.2.1. Ontology Definitions from Philosophy to Artificial Intelligent

The meaning of the term ontology has different connotations in philosophy and in computer science. So, an important distinction that should be drawn is between the notions of *Ontology* and *ontology* [Guarino, 98]. The difference is subtle but important. The former, written with a **capitalised ‘O’**, is an uncountable noun with no plural. It refers to the philosophical discipline that studies the nature of being. It is an old discipline introduced by Aristotle, who attempts to address questions such as: *‘What is being?’*

When written with a **lowercase ‘o’**, this form it is a countable noun for which a plural form, *ontologies*, exists. And still considered in a philosophical sense, in this context, *Guarino* referred to it as:

“Ontology as a particular system of categories accounting for a certain vision of the world”
[Guarino, 98].

In this perspective, ontology is independent from the language used to describe it. On the other hand, in its most prevalent use in AI, the word *ontology* takes a different meaning in AI; ontology refers to an *engineering artifact*, constituted by a specific *vocabulary* used to describe a certain reality, plus a set of explicit assumptions regarding the *intended meaning* of the vocabulary words.

The two readings of “ontology” described above are indeed related to each other, but in the AI notion of *ontology* is language dependent as opposed to the philosophical one. Although the AI community has agreed on the use and on the meaning of the term “ontology”, there is no a formal definition that is fully accepted and agreed upon by the community. A commonly agreed definition of ontology has given by Gruber:

“Ontology is an explicit specification of a conceptualisation” [Gruber, 93a].

Gruber's definition builds on the idea that a conceptualisation is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualisation, explicitly or implicitly. Figure 2.1 shows a distinction between different levels of interpretation of an ontology as an UML class diagram.

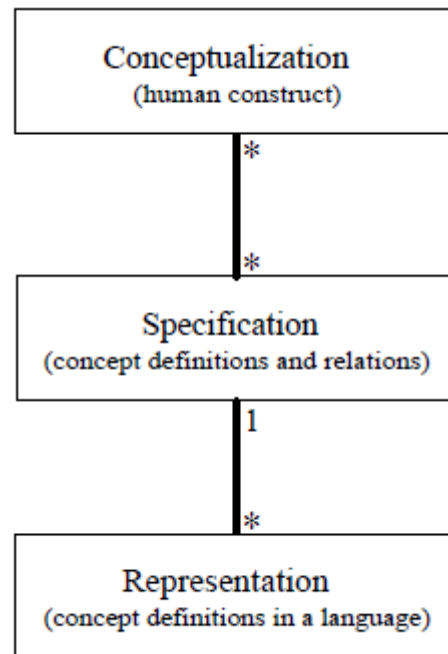


Figure 2.1. Three Interpretation Levels of an Ontology.

At the *conceptualisation level* is the actual interpretation of a specific domain by a (number of) human(s), which basically is an opinion about the important concepts and their relations. The *specification level* contains precise definitions of the concepts and the relations between them. Finally, the *representation level* is the actual formalism in which the specification is expressed. Simply said, different representations coincide with different ontology languages.

Gruber's and Guarino are not the only definitions of ontologies presented in the literature, although they are the most used. In fact, each research group working in the ontological field has tried to clarify their view on ontologies and have thus ended up developing their own definition of ontology. Indeed, these definitions depend on the purposes for which they have been developed. But all the definitions refer to the ontology as a common understanding of a domain, and represent it as repository of vocabulary for the knowledge of a domain. The vocabulary contains both formal and informal definitions. Some of these ontology's definitions are:

Borst has extended Gruber's definition [Borst, 97]:

"Ontology is a formal specification of a shared conceptualisation."

Studer and colleagues have merged Gruber's and Borst's definition, and have provided an explanation for the terms used [Studer et al.98]:

"Ontology is an explicit and formal specification of a shared conceptualisation of a domain of interest"

In this definition, a *conceptualisation* refers to an abstract model of some phenomenon in the world which identifies the concepts that are relevant to the phenomenon; *explicit* means that the type of concepts used, and that the constraints on their use are explicitly defined; *formal* refers to the fact that an ontology should be machine-readable and finally *shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

Ontologies also are playing an increasingly important role in knowledge-based systems. The relationship between ontologies and knowledge bases has been included in the definition by [Gruber, 93]:

"Ontology is a set of definitions of the content-specific knowledge representation primitives: classes, relations, functions, and object constants"

In this definition, there are: concepts (also known as classes), relations (properties), functions, instances, and axioms. This set of objects and the relationships among them are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. The main motivation of the ontologies is that they allow sharing and reuse of the formally represented knowledge bodies in computational form.

In another sense, Bernaras and colleagues have defined what a knowledge base provides while designing ontology:

"Ontology provides the means for describing explicitly the conceptualisation behind the knowledge represented in a knowledge base." [Bernaras et al. 96]

[Noy and McGuinness, 01], define it as:

"Ontology is a formal explicit description of concepts in a domain of discourse"

Here, ontology together with a set of individual instances of classes constitutes a knowledge base. In reality, there is a fine line where the ontology ends and the knowledge base begins.

Finally, the ontology community distinguishes between ontologies that are mainly taxonomy from ontologies that model a domain, providing constraints about the semantics of the domain:

- *Lightweight ontology* includes concepts, taxonomy of concepts, relations between concepts and properties that describe the concepts.
- *Heavyweight ontology* adds to the previous definition axioms and constraints.

Lightweight and heavyweight ontologies can be modelled with different modelling techniques and can be implemented in different languages and tools [Uschold, 98]. Ontologies can be classified in different categories depending on how they are expressed. For the purposes of this thesis, ontology is a formal explicit description of concepts in a discourse domain which is transport domain in our case.

2.2.2. Ontology Components

Before illustrating the different types of ontologies we introduce in what follows the ontologies components. There are different techniques that can be used to model and represent ontologies such as frames, first-order logic [Gruber, 93], description logics [Baader et al., 03], software engineering techniques [Cranefield and Purvis, 99] or database technologies [Thalheim, 00].

For all techniques, knowledge about a domain in ontologies is formalized using five kinds of components: concepts, relations, functions, axioms and instances [Gruber, 93]. An ontology structure O is defined as:

$$O = \{C, I, R, F, A\}$$

Where:

- **C:** Concepts are used in a broad sense, a concept can be anything about which something is said and could also be the description of a task, function, action, strategy, reasoning process, etc.
- **I:** Instances are used to represent elements that are the actual objects of the world.
- **R:** Relations represent a type of interaction between concepts of the domain. They are formally defined as any subset of a product of n sets, that is:

$$R: C_1 \times C_2 \times \dots \times C_n$$

Examples of binary relations include: *subconcept-of* is the pair (C_p, C_c) , where C_p is the parent concept and C_c is the child concept.

- **F:** Functions are a special case of relations where the set of functions defined on the set of concepts and that return a concept. Formally, functions are defined as:

$$F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$$

Example of function is *price-of-a-used-car* that calculates the price of car depending on the car-model, manufacturing date and number of kilometres.

- **A:** Axioms is first order logic predicates that constrain the meaning of concepts, relationships and functions.

Concepts, relations and instances are used to model lightweight ontologies. Heavyweight ontologies include also axioms and functions. The term *concept* and *class* are used as synonyms. Concepts are organized into is-a hierarchy which permits *inheritance* to be exploited in the structure. Example, if A is an ancestor of B (denoted by $A \rightarrow B$) and $B \rightarrow C$ then, $A \rightarrow C$.

Concepts can be defined in terms of characteristic features describing them, which are called *attributes*. Attributes are shared by concepts either in their original form or modified in order to give the inheriting class, known also as *subclass*, a more restrictive definition than the one provided by the parent concept. If the concepts are organized in a is-a hierarchy, then the inheritance is extended also to attributes.

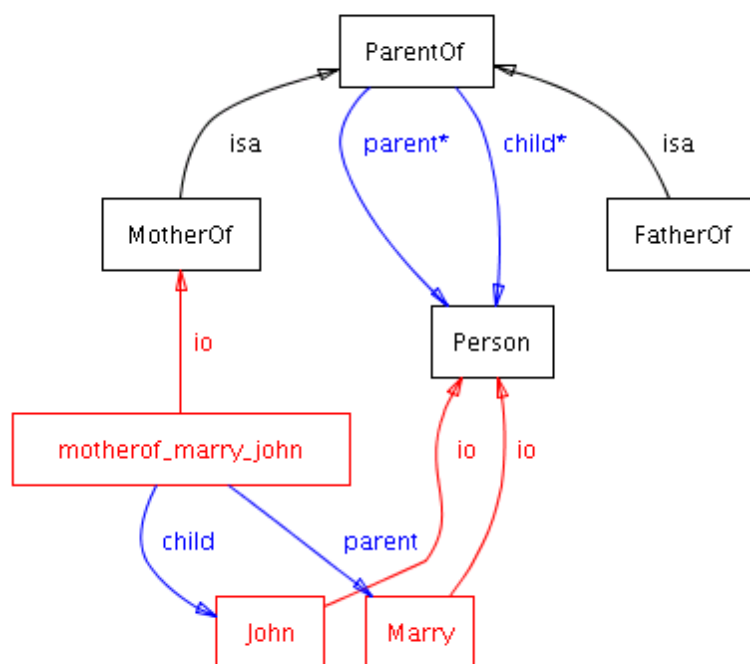


Figure 2.2. Example of Parent Ontology.

Figure 2.2 is shown a well known example of an ontology use. Ontology defines parent-child relations. In this ontology (ParentOf, Person, FatherOf, MotherOf) are the concepts. The relations between the concepts represent in (is-a, parent, child). In this example (John and Marry) are instances of the concept Person. Based on parent ontology model, the axiom information “Mary is mother of John” is stored. If one

will ask the system who is ParentOf John then the system can answer Mary, even if such information can be only inferred and it is not directly stored in the system.

2.2.3. Principal Design of Ontology

[Gruber, 93] propose formal ontologies design. He identifies five principles criteria to guide and evaluate ontology designs:

1. **Clarity:** The ontology should effectively communicate the intended meaning of defined terms. Definitions should be objective. While the motivation for defining a concept might arise from social situations or computational requirements, the definition should be independent of social or computational context. Formalism is a means to this end. All definitions should be documented with natural language.
2. **Coherence:** The ontology should be coherent, i.e., it should sanction inferences that are consistent with the definitions. At the least, the defining axioms should be logically consistent. Coherence should also apply to the concepts that are defined informally, such as those described in natural language documentation and examples.
3. **Extendibility:** The ontology should be designed to anticipate the uses of the shared vocabulary. It should be able to define new terms for special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions.
4. **Minimal encoding bias:** The conceptualisation should be specified at the knowledge level without depending on a particular symbol-level encoding. The encoding bias should be minimized, because knowledge-sharing agents may be implemented in different representation systems and styles of representation.
5. **Minimal ontological commitment:** The ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities.

2.2.4. Main Languages of Ontology

Languages for representing data and knowledge are an important element of the ontology. There are different computer languages that play an important role in the ontology (XML Schemas, DTDs, CycL (Cyc Language), Ontolingua, XOL, SHOE, etc...). But, in this section, we explain the languages which are web-standard

languages (such as: XML and RDF) and web-based language (such as: OIL, DAML and OWL) which are the most important languages for our work, (Figure 2.3).

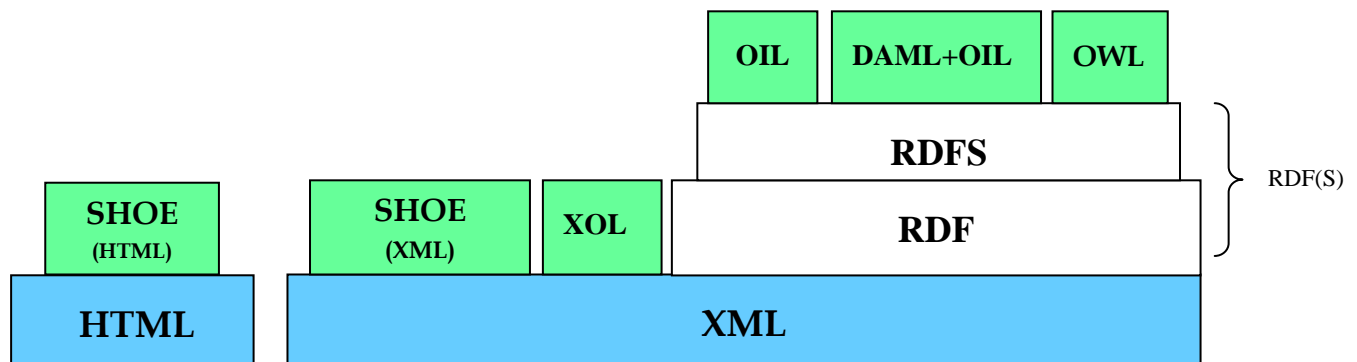


Figure 2.3. Ontology Markup Languages.

2.2.4.1. XML

XML (eXtensible Markup Language) describes a class of data objects called XML documents and partially describes the behaviour of computer programs which process them. XML is a subset of Standard Generalized Markup Language (SGML) which its goal is to enable generic SGML to be served, received and processed on the Web. XML has been designed for ease of implementation and for interoperability with both SGML and HyperText Markup Language (HTML) [Bray et al., 04].

For more details about XML definition, the *eXtensible* indicates an important difference and a main characteristic of XML. Indeed, XML only provides a data format for structured documents, without specifying an actual vocabulary. This makes XML universally applicable. Besides many proprietary languages a number of standard languages are defined in XML (called XML applications). For example, XHTML is a redefinition of HTML 4.0 in XML.

The *Markup* means that certain sequences of characters in the document contain information indicating the role of the document's content. It takes the form of words between angle brackets, called tags—for example, <name> or <h1>. In this aspect, XML looks very much like the well-known language HTML. Figure 2.4 presents a piece of XML which consists of several components that describe the mark-up of the different parts of a document. As follow:

```
<?xml version="1.0"?>
  <Employees> List of persons in university:
    <person >
```

```

    <Name>John Smith</Name>
    <Position>Professor </Position>
  </person>
</employees>

```

Figure 2.4. XML Code Example

In this example, we see that XML document includes special commands passing along the program that processes or views the XML document. In the XML example above, `<?xml version="1.0"?>` is a processing instrument. Here, this piece of XML consist normally of an opening tag and a closing tag, for example, `<person>` and `</person>`. Elements might contain other elements or text. If an element has no content, it can be abbreviated as `<person/>`. Elements should be properly nested: a child element's opening and closing tags must be within its parent's opening and closing tags. Every XML document must have exactly one root element. Elements can carry attributes with values, encoded as additional "word = value" pairs inside an element tag.

2.2.4.2. RDF and RDF Schema

[Lassila and Swick, 99] was developed by the World Wide Web Consortium (W3C) as a language for processing metadata. Resource Description Framework (RDF) provides the interoperability for applications to exchange machine-understandable information on the web. RDF includes three basic object types:

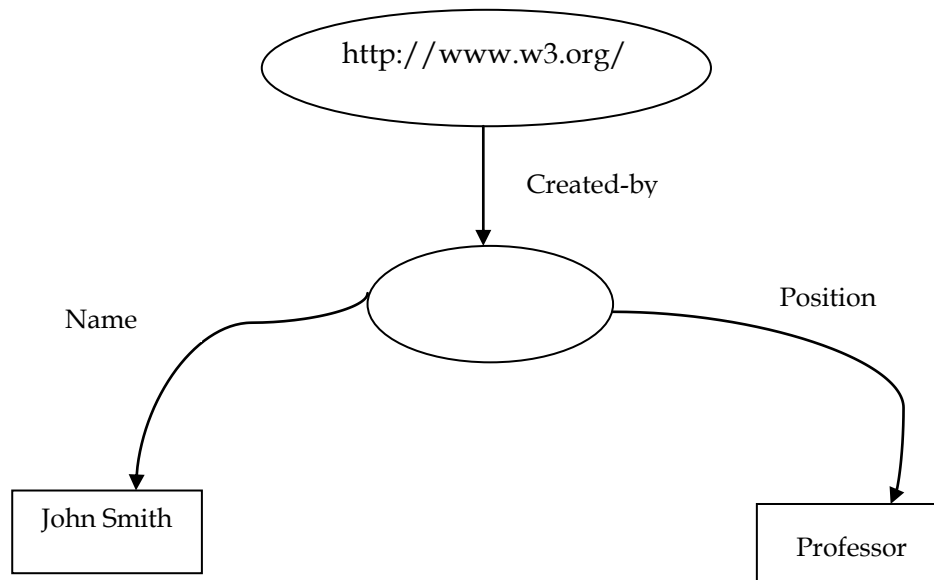
- **Subject** is the resource (URI or a blank node) from which the arc leaves,
- **Predicate** is the property that labels the arc,
- **Object** is the resource or literal pointed to by the arc.

For example, to state that a specific web page was created by a person in the university whose his name is "John Smith" and his position is a "Professor". You can easily depict an RDF model as a directed labelled graph (Figure 2.5). Each arc in an RDF Model is a *statement*. Each statement asserts a fact about a resource. A statement is sometimes called a *triple*, because of its three parts. The following three triples are required for our example (Table 2.1).

Subject	Predicate	Object
http://www.w3.org/	Created-by	anon 1
anon 1	Name	"John Smith"
anon 1	Position	"Professor"

Table2.1. RDF Triple Example

Figure (2.5) shows such a graph for the triples listed above.

**Figure2.5. RDF Graph**

The specification of the data model includes such an XML-based encoding for RDF. In this syntax, the triples above could be expressed as follows (Figure 2.6):

```
<?xml version="1.0"?>
  <rdf:RDF>
    <rdf:Description rdf:about="http://www.w3.org/">
      <created_by>
        <rdf:Description>
          <Name>John Smith</Name>
          <Position>Professor </Position>
        </rdf:Description>
      </created_by>
    </rdf:Description>
  </rdf:RDF>
```

Figure2.6. RDF Code Example

As with XML, an RDF model does not define the semantics of any application domain or make assumptions about a particular application domain. It just provides a domain-neutral mechanism to describe metadata. Defining domain-specific properties and their semantics require additional facilities.

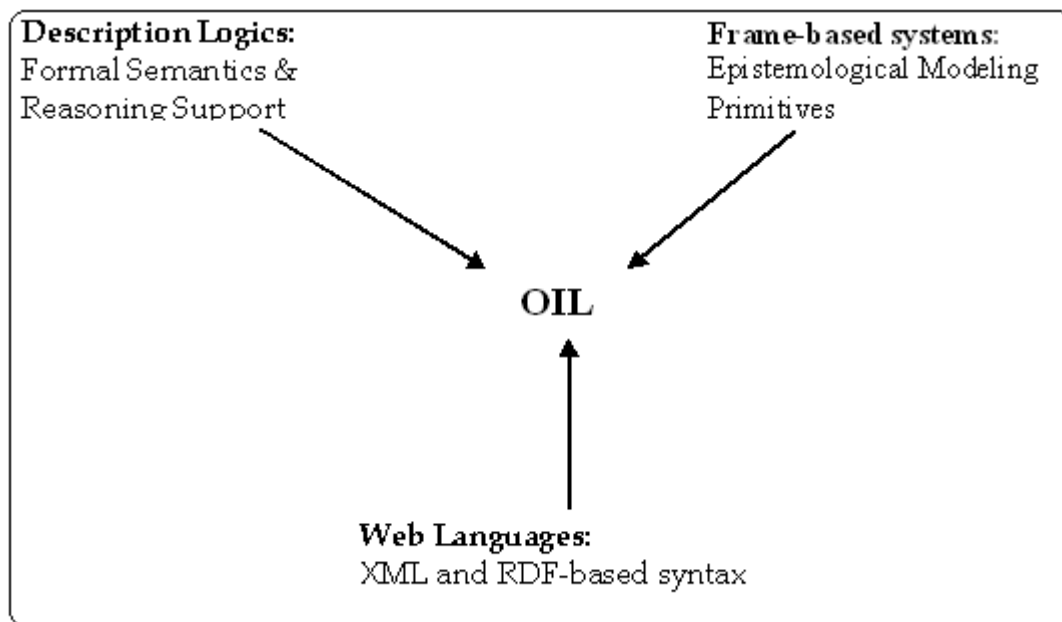
RDF Schema

To allow for the creation of controlled, sharable, and extensible vocabularies, the RDF working group has developed the RDF Schema Specification [Brickley and Guha 99]. This specification defines a number of properties that have specific semantics. The property `rdf:type` is used to express that a resource is a member of a given class, while the property `rdfs:subClassOf` essentially states that one class is a subset of another. With the `rdfs:subClassOf` property, schema designers can build taxonomies of classes for organizing their resources. RDF Schema also provides properties for describing properties: the property `rdfs:subPropertyOf` allows properties to be specialized in a way similar to classes, while the properties `rdfs:domain` and `rdfs:range` allow constraints to be placed on the domain and range of a property.

One problem with RDF and RDFS is their lack of a clear distinction between the object and meta-levels. In RDF and RDFS, properties are the central modelling primitive. This model is very hard for the knowledge modellers to understand and use. The second problem is that the semantics of their primitives are loosely defined. There is no inference model that precisely fits the semantics of the RDF modelling primitives. The third problem is that RDF and RDF Schema allow only simple semantics to be associated with identifiers. Other RDF-based languages, including OIL, DAML and OWL, try to solve these three problems by introducing description logic into the web. As we will in the next sections.

2.2.4.3. OIL

The Ontology Inference Layer (OIL) [Fensel, 00], is a project which proposed by a group of European researchers with the support of the European Commission, suggests a web-based representation and inference layer for ontologies. Compatible with RDFS, and OIL includes a precise semantic for describing term meanings and implied information



Figuer2.7. OIL Research Fields

OIL unifies three important aspects provided by three different research fields (Figure 2.7):

- **Description Logic (DL):** DLs describe knowledge in terms of concepts and role restrictions that are used to automatically derive classification taxonomies. OIL inherits from DL its *formal semantics* and the *efficient reasoning support* developed for these languages.
- **Frame-based Systems:** The Frame-based System in OIL is classes (i.e., frames) with certain properties called attributes. So, OIL is based on the notion of a concept and the definition of its superclasses and attributes. Relations can also be defined not as attributes of a class, but as independent entities having a certain domain and range. Actually, concepts are defined as subclasses of role restriction.
- **Web Standards:** The syntax of OIL is based on RDF and RDFS, which provide two important contributions: a standardized syntax for writing ontologies and a standard set of modelling primitives including subclass, domain and range relationships.

An example of OIL ontology consists of:

```

class-def Product
  slot-def Price
    domain Product
  
```

```

class-def DellNotebook
      subclass-of Product
      slot-constraint Price
      has-value "$779"

```

Figure2.8. OIL Code Example

Figure 2.8 defines the property “Price” for class “Product”. The instance “DellNotebook” has super class “Product”. As a result, “DellNotebook” inherits the property “Price”, which has value “\$779”.

2.2.4.4. DAML+OIL

The Darpa Agent Markup Language (DAML) project is the joint effort from United States DAML group and Europe Semantic Web Technologies. The aim of this project is to achieve semantic interoperability between web pages, databases and programs. Based on RDF/RDFS and benefited from OIL, the DAML language provides a set of tools for programmers to incorporate broad concepts into their web pages.

An ontology language, DAML+OIL is designed to describe the *structure* of a domain. DAML+OIL takes an object oriented approach, with the structure of the domain being described in terms of *classes* and *properties*. Ontology consists of a set of *axioms* that assert characteristics of these classes and properties. Asserting that resources are instances of DAML+OIL classes or those resources are related by properties is left to RDF, a task for which it is well suited [Horrocks et al., 02].

Figure 2.9 summarises the constructors in DAML+OIL. The standard DL syntax is used in this example for compactness as the RDF syntax. In the RDF syntax, for example, Human[] Male can be written as:

```

<daml:Class>
  <daml:intersectionOf
    rdf:parseType="daml:collection">
      <daml:Class rdf:about="#Human"/>
      <daml:Class rdf:about="#Male"/>
    </daml:intersectionOf>
  </daml:Class>

```

Figure2.9. DAML+OIL Code Exemple

There are a few differences between OIL and DAML+OIL, which are described at the DAML+OIL⁸. The most important differences are:

- The RDF syntax is different. The RDF encoding of DAML+OIL has lost some features of the OIL RDF encoding. Indeed, RDF syntax, most notably in the use of lists in DAML+OIL, [Bechhofer et al., 01a].
- OIL has better “backwards compatibility” with RDFS. In case of defined (non-primitive) concepts, half of the two way implication is still accessible to RDFS agents, because of the use of `rdfs:subClassOf`.
- DAML+OIL have an explicit `samePropertyAs` property. In OIL this should be expressed using mutual `rdfs:subPropertyOf` statements.
- DAML+OIL have two mechanisms to state disjointness. DAML+OIL provides both a “`disjointWith`” that assert two classes are disjoint and a “`Disjoint`” class that can be used to assert pair wise disjointness amongst all the classes in a list. OIL simply uses *disjoint* to assert disjointness amongst two or more classes.
- Different property characteristics are supported: DAML+OIL do not support `SymmetricProperty`, whereas OIL does not support `UnambiguousProperty`. However, logically both notions can be expressed via the combination of other characteristics.

2.2.4.5. OWL

Web Ontology Language (OWL)⁹ is the result of a standardization process of the DAML+OIL language by the World Wide Web Consortium (W3C)¹⁰. OWL has become a W3C Recommendation in [February, 04], [McGuinness and van Harmelen, 04]. OWL is designed for the applications to process the content of information instead of just presenting information to humans. By providing additional vocabularies along with formal semantics, OWL facilitates greater machine interpretability of web content than that supported by XML, RDF or RDFS.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

- ***OWL Lite*** supports those users who primarily needing a classification hierarchy and simple constraints. For example, it is envisaged that OWL-Lite

⁸ <http://www.daml.org/2000/12/differences-oil.html>

⁹ <http://www.w3.org/TR/owl-ref/>

¹⁰ <http://www.w3.org>

will provide a quick migration path for existing thesauri and other taxonomies.

- **OWL DL** supports those users who want the maximum expressiveness while retaining computational completeness and decidability. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions. In Fact, OWL DL is so named due to its correspondence with *description logics*, a field of research that has studied the logics that form the formal foundation of OWL.
- **OWL Full** is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is therefore not possible to perform automated reasoning on OWL-Full ontologies.

Ontology developers adopting OWL should consider which sublanguage best suits their needs. The choice between OWL Lite and OWL DL depends on the extent to which users require the more-expressive constructs provided by OWL DL. The choice between OWL DL and OWL Full mainly depends on the extent to which users require the meta-modeling facilities of RDF Schema. When using OWL Full as compared to OWL DL, reasoning support is less predictable since complete OWL Full implementations do not currently exist [Peter, 03]. Table 2.2 provides a comparison between these three sublanguages¹¹.

	OWL Lite	OWL DL	OWL Full
Compatibility with RDF	Theoretically, no rdf document can be assumed to be compatible with OWL Lite	Theoretically, no rdf document can be assumed to be compatible with OWL DL	All valid rdf documents are OWL full
Restrictions on class definition	Requires separation of classes, instances, properties, and data values	Requires separation of classes, instances, properties, and data values	Classes can be instances or properties at the same time
Cardinality Constraints	Cardinality: 0/1 MinCardinality: 0/1 MaxCardinality: 0/1	Cardinality >= 0 MaxCardinality >= 0 MinCardinality >= 0	Cardinality >= 0 MaxCardinality >= 0 MinCardinality >= 0
Meta-modeling	Does not allow meta-modelling	Does not allow meta-modelling	Allows meta-modelling.

¹¹ <http://ragrawal.wordpress.com/2007/02/20/difference-between-owl-lite-dl-and-full/>

Class	OWL: class is subclass of RDFS: class	OWL: <i>class is subclass of RDFS: class.</i>	RDFS: class and OWL: class are equivalent
--------------	---------------------------------------	---	---

Table 2.2 Comparison between OWL Lite, OWL DL and OWL full

An example where OWL is a vocabulary extension of RDF, Figure 2.10, as follow:

```
<owl:Class rdf:ID="Opera">
  <rdfs:subClassOf rdf:resource = "#MusicalWork" />
</owl:Class>
```

Figure2.10. OWL Code Example

This class axiom (Figure2.9) declares a subclass relation between two OWL classes that are described through their names (Opera and MusicalWork). Subclass relations provide necessary conditions for belonging to a class. In this case, to be an opera the individual also needs to be a musical work.

2.2.4.6. Comparison

In this section we explain some comparison between the ontology languages.

XML vs. RDF: RDF is an application of XML to represent meta-data where it provides a standard way to meta-data in XML. RDFS provides a fixed set of modelling primitives for defining an ontology (classes, resources, properties, is-a, element-of relationship, etc), and a standard way on how to encode them in XML.

RDF(S) vs. OIL: RDF can be used as a representation format for OIL. To ensure maximal compatibility with existing RDF/RDFs applications and vocabularies:

- The abstract OIL class *OntologyExpression* is a subclass of *rdfs:Resource*. The abstract OIL class *OntologyConstraint* is a subclass of *rdfs:ConstraintResource*.
- OIL slots are realized as instances of *rdf:Property* or as subproperties of *rdf:Property*. The subslot relationship can be expressed via *rdfs:subPropertyOf*. *rdf:Property* is enriched in OIL with a number of properties that specify inverse and transitive roles and cardinality constraints, what is not possible in RDF/RDFS.

OIL uses the existing primitives of RDFS as much as possible to retain an unambiguous mapping between the original OIL specification and its RDFS serialization.

DAML+OIL vs. OWL: The difference between DAML+OIL and OWL is not a great deal. But OWL abstract syntax has reverted to grouping axioms into frame like structure, which makes frame-based tools such as Protégé or DL based ones like OilEd easy to use (see section 2.6.2). In this sense, OWL is closer to OIL due to its frame-based feature while DAML+OIL is more like DL.

Finally, in 2000, Tim Berners-Lee,[Berners-Lee,02], proposed the information architecture of Semantic Web (as shown in Figure 2.11) It indicates that ontology plays a role as the core of semantic information and the foundation of enabling reasoning services. In this vision, RDF and OWL based ontology languages are in the core of the architecture.

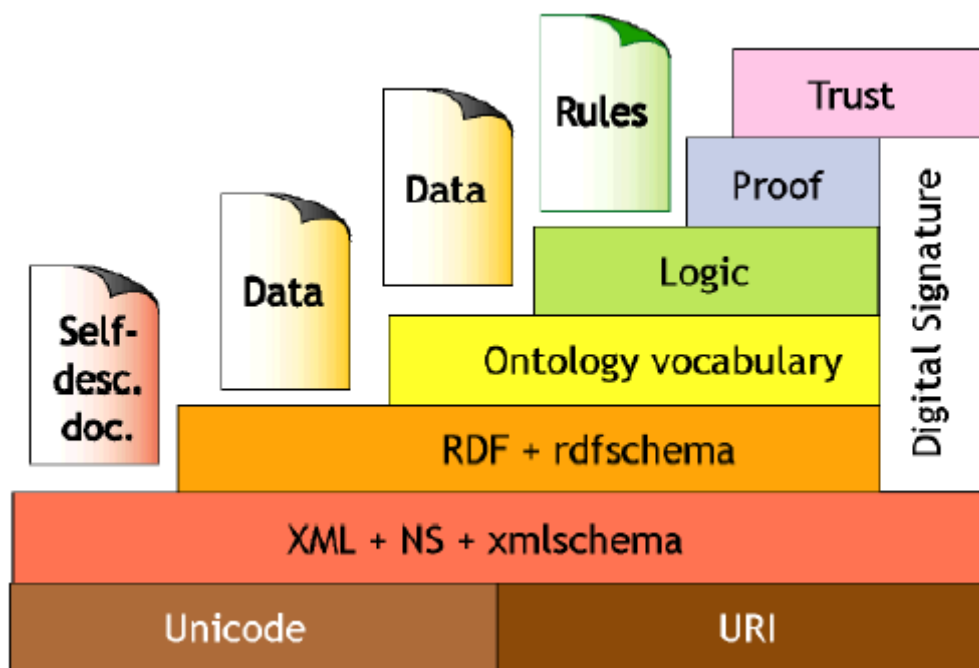


Figure 2.11 Berners-Lee's Architecture of Semantic Web on XML

After have seen the different ontology languages, we will go to studding the different ontology development methodology and tools which use those languages to valid the ontology.

2.2.5. Ontology Development Methodologies and Tools

2.2.5.1. *Ontology Methodologies*

Several research groups are looking for adequate ontology development methods. However, the variables are so many that it may well be impossible to come up with a single method that is adequate for all situations. Probably, the best solution will be a choice among different possibilities or a composition of different ontology development methods. We will summarize various ontology construction methods to understand the difficulties involved in the ontology elicitation, modeling, and construction processes. None of this method is the most adequate. Each has its use, depending on the application's specificity as we will see in the rest of this section.

Firstly, Uschold and King Ontology Development Method [Uschold and King, 95]:

The research group at Edinburgh University proposed the first ontology construction method. It is composed of four distinct stages:

1. *Identify purpose and scope of the ontology*: Define why the ontology is being built and for what it is going to be used. An ontology may be designed with the intention of knowledge sharing, knowledge reusing, or as part of an existing knowledge base.
2. *Build the ontology*:
 - a. *Capture*: Define concepts and relationships textually. The textual descriptions should not use the traditional dictionary approach. By doing so, the relationships between key concepts will be made explicit.
 - b. *Code*: Formalize the concepts and relationships defined in previous step.
 - c. *Integrate*: Question the possibility of reusing existing ontologies. This activity can, and should, be made in parallel with the others.
3. *Evaluate the ontology*: Use technical criteria to verify the specification, using competency questions and real-world validations.
4. *Document the ontology*: Describe the ontology construction process. The final format may vary according to the type of ontology in question. Users may determine their own conventions, such as representing class names in capital letters and relationships in italics.

The unique feature of this methodology is that it strongly recommends the utility of an informal ontology which is easily understood by many people and works as a useful specification of a formalized ontology.

Toronto Virtual Enterprise (TOVE) Methodology [Gruninger and Fox, 95]: TOVE methodology was developed intended to help enterprise process modeling at Toronto University. The authors used motivating scenarios to describe problems and examples that were not addressed by existing ontologies, as proposed by Uschold and Kings. It is composed of the following core steps:

1. *Description of motivating scenarios:* Make motivating scenarios informally in order to formalize the requirement specification of the ontology.
2. *Formulation of informal competency questions:* Based on the motivating scenarios, competency questions are elaborated, for which the ontology to be built must provide valid answers.
3. *Specification of ontology terms using a formal representation:* Define a set of concepts from the competency questions. These concepts are the basis of a formal specification, developed using a knowledge representation language, such as first-order logic or the Knowledge Interchange Format (KIF).
4. *Formulation of formal competency questions:* Describe the competency questions using a formal language.
5. *Axiom specification:* Formally describe rules that capture the semantics associated with the ontology concepts and relations.
6. *Verification of ontology completeness:* Establish conditions to characterize the ontology as complete, based on the formal competency questions.

This approach has a major problem for it supposes that the concepts of ontology can be derived from motivating scenarios alone. In fact, the scenario technique is best used to observe dynamical aspects of a given environment, rather than to identify static entities [Mizoguchi, 04].

Ontology Development 101 [Noy and McGuinness, 01]: This method was proposed by Noy and McGuinness as a guide to help users create their first ontology. The authors summarize their experiences with the development of the Protégé2000, Ontolingua and Chimaera tools (section 2.2.5.2). In order to model ontology, [Noy and McGuinness, 01] suggest the following step-by-step approach:

1. *Determine the domain and scope of the ontology:* For establishing ontology by defining its domain and scope. There are some of questions which the ontology should provide answers on them. Some of those questions:
 - a) What is the domain that the ontology will cover?
 - b) For what we are going to use the ontology?
 - c) For what types of questions the information in the ontology should provide answers?
 - d) Who will use and maintain the ontology?

The answers on these questions help who will use and maintain the ontology during the ontology-design process.

2. *Consider reusing existing ontologies:* Reusing existing ontologies may be a requirement if our system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies. So, it is very important task to try to refine and extend existing sources for a particular domain and task.
 - a) *Enumerate important terms in the ontology:* It is useful to write down a list of all terms either to make statements about them or to explain them to a user.
 - b) *Define the classes and the class hierarchy:* (Top-Down, Bottom-Up, Combination) are some possible approaches which help in developing a class hierarchy [Uschold and Gruninger, 96]
 - c) *Define the properties of classes – slots:* The classes alone are enough information to answer the proposed questions in Step 1. For that, once we have defined some of the classes, we must describe the internal structure of concepts.
 - d) *Define the facets of the slots:* Slots can have different facets describing the value type, allowed values and cardinality.
 - e) *Create instances of a class:* The last step is creating individual instances of classes in the hierarchy; this creating requires: choosing a class, creating an individual instance of that class, and filling in the slot values.

For this thesis, we apply the Ontology Development 101 methodology. We begin the construction of our ontology by the competency questions which are suitable to our

transport domain. Then, we define the formalism (i.e., define: classes, class hierarchy, etc...) of our ontology. Finally, we use Protégé2000 to implement our ontology, as we will see in the third chapter.

2.2.5.2. *Ontology Tools*

Incorporating the methodologies and languages, there have been developed many environments for ontology development; notable among these tools are: OntoLingua, WebOnto, Protégé, OilEd, and OntoEdi. Most of these tools provide an integrated environment to build and edit ontologies, check for errors and inconsistencies (using a reasoner), browse multiple ontologies, share and reuse existing data by establishing mappings among different ontological entities. However, these tools are influenced by traditional Knowledge Representation (KR) which is based ontology engineering methodologies, with steep-learning methods, making it heavy to use for casual web ontology development.

In this section, we present the most popular ontology development tools:

- **Ontolingua [Farquhar et al., 97]:** It was created by the Knowledge Systems Laboratory (KSL) at Stanford University. The system consists of a server and a representation language. The Ontolingua Server is a set of tools and services that support the building of shared ontologies between distributed groups. The ontology server architecture provides access to a library of ontologies, translators to languages and an editor to create and to browse ontologies. Remote editors can browse and edit ontologies, and can access any of the ontologies in the ontology library using the OKBC¹² (Open Knowledge Based Connectivity) protocol.
- **WebOnto [Domingue, 98]:** WebOnto is a tool developed by the Knowledge Media Institute (KMI) of the Open University (England). It supports the collaborative browsing, creation and editing of ontologies, which are represented in the Operational knowledge Modelling Language (OCML). OCML can be translated to Ontolingua as well as translated to/from (RDF(S)) (section 2.2.4).
- **Protégé-2000 [Noy et al., 00]:** Protégé-2000 developed by the Stanford Medical Informatics (SMI) group at Stanford University for knowledge acquisition. Protégé-2000 has thousands of users all over the world who use the system for projects ranging from modelling cancer-protocol guidelines to modelling nuclear-power stations. Protégé-2000 is freely available for download under the Mozilla open-source license.

¹² <http://www.ai.sri.com/~okbc/>

The core of the Protégé-2000 is the ontology editor which provides a graphical and interactive ontology-design and knowledge-base-development environment. It allows the user to perform knowledge-management tasks. Ontology developers can access relevant information quickly whenever they need it, and can use direct manipulation to navigate and manage ontology. Tree controls allow quick and simple navigation through a class hierarchy. The knowledge model of Protégé-2000 is OKBC-compatible.

Protégé-2000 is the most ontology-editing environment with its **scalability** and **extensibility**. One of the major advantages of the Protégé-2000 architecture is: the system is an open, modular fashion. The architecture allows the development and integration of plug-ins. Plug-ins are additional modules that extend the Protégé system's core. The Protégé plug-ins Library contains contributions from developers all over the world. Most plug-ins falls into one of the three categories:

1. **Back-ends** that enable users to store and import knowledge bases in various formats.
2. **Slot Widgets**, which are used to display and edit slot values or their combination in a domain-specific and task-specific ways.
3. **Tab Plug-ins**, which are knowledge-based applications usually tightly linked with Protégé knowledge bases.

For this thesis, we design our transport ontology by Protégé-2000, thanks to it because we have also used its plug-ins like (BeanGenerator and OntoViz) where the former automatically generated java classes comply with the JADE specifications. In fact, the intelligent software agents (section 3.2) can profit from this mechanism since the resulting java source files can be accessed easily from any java program. The later provide a convenient graphical visualization of ontology models.

2.3. Multi-Agent System (MAS) Overview

2.3.1. Agent Definitions and Architecture

Twenty years ago, Distributed Artificial Intelligent (DAI), and in particular Multi-Agent System (MAS), emerged in the field of AI. Nowadays, MAS are not a simply research topic, but also an important subject of academic teaching and of industrial and commercial applications.

When we talk about agents we mean agents in a Multi-Agent System, where more than one agent is present, and where agents can interact. The term “agent” can be understood differently depending on the domain of research

“Agents are often described as entities with attributes considered useful in a particular domain” [Flores, 99].

This is the case with *Intelligent Agents*, where agents are seen as entities that emulate mental processes or simulate rational behaviour, [Flores, 99]; intelligent agents interact by passing a message among each other. They use techniques from the artificial intelligence area such as learning, reasoning or negotiation and decision support.

Another definition for a software agent is used:

“An agent is a computer system capable of flexible autonomous action in a dynamic, unpredictable and open environment.” [Luck, 03]

Software Agents are based on a software program, it has two types:

- *Simulation Agents*, which help to simulate a discrete system, which cannot be described by differential equations or which are too complex e.g. simulating traffic in the city.
- *Mobile Agents*, which can move from one place to another. There are 2 types of mobility: strong and weak. The *strong mobility* means a migration of an agent with its execution state and its variables values from one computer to another. The *weak mobility* is when an agent migrates and carries only the code and variables values. [Laclavik, 06]

Finally, *Robotic Agents* are based on hardware. Many combinations between robotic and software agents or intelligent and mobile agents exist.



Figure 2.12. Agent Classification Architecture

As we have seen that various definitions from different disciplines have been proposed for the term Agent (Figure 2.12). For the **Multi-Agent System (MAS)**, we find that MAS from the point of view of the DAI:

“MAS is a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity”, [Flores, 99].

In another general meaning, and it is now used for all types of systems.

“MAS is a system composed of several agents with the capability of mutual interaction. A competitive or cooperative interaction occurs when two or more agents are brought into a dynamic relationship through a set of reciprocal actions” [Ferber, 99].

Finally, the agents used in MAS appreciate the following several properties:

- **Autonomy.** Each agent works by itself and it doesn't need the direct intervention of humans.
- **Reactivity.** Each agent can perceive and respond to the relevant changes timely.
- **Pro-activity.** Agents are able to exhibit goal-directed behaviors by taking the initiative, but not simply act in response to their environments.
- **Co-operability.** Each agent can communicate with other relevant agents in order to realize common objectives.
- **Mobility.** Agents are able to travel through computer networks. An agent in one host may create another agent or transport from host to host during execution.

2.3.2. Agent Communication Languages

As we have mentioned previously, today, the MAS technologies is being used in a large range of important industrial applications area. These applications needs have a one thing in common which is the gent must be able to 'talk' to each other to decide what action must to take and how this action can be coordinated to other's actions.

The language used by the agents for this exchange is called the 'Agent Communication Language (ACL)' [Chaib-Draa and Dignum, 02]. An ACL stems from the need to coordinate the actions of an agent with that of other agent. It can be used to share the information and knowledge among agents in distributed computing environments. One of the main objectives of an ACL design is to model a

suitable framework that allows heterogeneous agents to interact, to communicate with meaningful statements that convey information about their environment or knowledge [Kone et al., 00]. In addition, in the process of solving a complex task, an agent may need to cooperate in a concise language with other agents able to add their contribution. This property gives multi-agent systems the image, as [Colombetti, 98] stated, of ‘a social community’ whose constituents rely on one another to perform tasks on behalf of their users.

ACLs are high-level languages whose primitives and structures are used to support all the kinds of collaboration, negotiation, and information transfer required in multi-agent interactions. Several ACLs (KQML, ARCOL, LOGOS and FIPA-ACL) have been implemented. But the most used ACL are KQML and FIPA-ACL. We will study in the follow.

2.3.2.1. *KQML Language*

Indeed, the first attempt to standardized ACL came from the ARPA knowledge sharing project (section 2.2.4) and produced Knowledge Query and Manipulating Language (KQML). KQML is a general-purpose language that supports communication between several agents with a set of reserved primitives called *performatives* [Finin et al., 95]. In the context of ARPA project, three working groups with complementary objectives compose this work: the Interlingua group designed the Knowledge Interchange Format (KIF) as a common language for describing a message content, the Shared and Reusable Knowledge Base group described the content of sharable knowledge bases, and the the External Interface group produced the KQML language and looks at interactions of system components at run time.

KQML includes many performatives of speech act which agent use to assert facts, request information, or subscribe to services. A simple scenario of conversation in KQML message has the following syntax, (Figure 2.13):

```
( Tell
  : sender A
  : receiver B
  : content "snowing"
)
```

Figure 2.13. KQML Message Example

In this example, the agent A tells the agent B that the proposition “it is snowing” is true.

The semantic of the KQML language is based on the idea that each agent has its own knowledge base (KB). For that, the main advantage of KQML is its ability to support a wide range of agent architecture with its extensible set of performatives.

2.3.2.2. *FIPA-ACL*

The Foundation for Intelligent Physical Agents (FIPA)¹³ is an international organization that aims to develop a set of generic agent standards with the contribution of several parties involved in agent technology. FIPA assigns tasks (ontologies, semantics, architectures, gateways and compliance) to technical committees.

In particular, the FIPA standard for ACL attempts to identify the practical components of inter-agent communication and cooperation and define a concise formal semantics and supporting communication protocols. In fact, the main FIPA standard specification [FIPA, 99] is composed of seven sub-specifications: agent's management, agent's communication, agent's interaction, personal travel assistance, personal assistance, audio-visual entertainment and broadcasting, and network management and provisioning.

FIPA-ACL is also based on speech act theory and messages which are also considered as communicative acts which objective is to perform some action by virtue of being sent. A simple scenario of conversation in FIPA-ACL message where this example related to a call for papers for a conference, the example has the following syntax, (Figure 2.14): Agent A sends a call for papers to a number of agents R.

```
(CFP
  : sender A
  : receiver R
  : reply-with call-proposal
  : language fipa-sl
  : content ((action R (submit (paper; conf)) ) true)
  : ontology conference
  : protocol FIPA-Contract-Net
)
```

Figure 2.14. FIPA-ACL Message Exemple

¹³<http://www.fipa.org>

2.3.3. Agent Knowledge Model

As already mentioned, there are several agent models. The main focus in literature is the agent's communication with environment and other agents. The internal knowledge model is left for an agent creator. FIPA does not cover this area of agent systems either. FIPA specifications just describe how agents should communicate and how they can share, translate or communicate ontologies.

The support for ontologies or agent knowledge modeling provided by Java Agent Development Framework (JADE) ¹⁴ is designed to automatically perform all the conversion and check operations as in Figure 2.15. Ontology elements and its relations and properties are described as a real java object. This is powerful for its manipulation, when developing an agent code. Instances of ontology classes can be passed in ACL messages in the form of FIPA-SL language¹⁵. FIPA-SL is based on predicate logic.

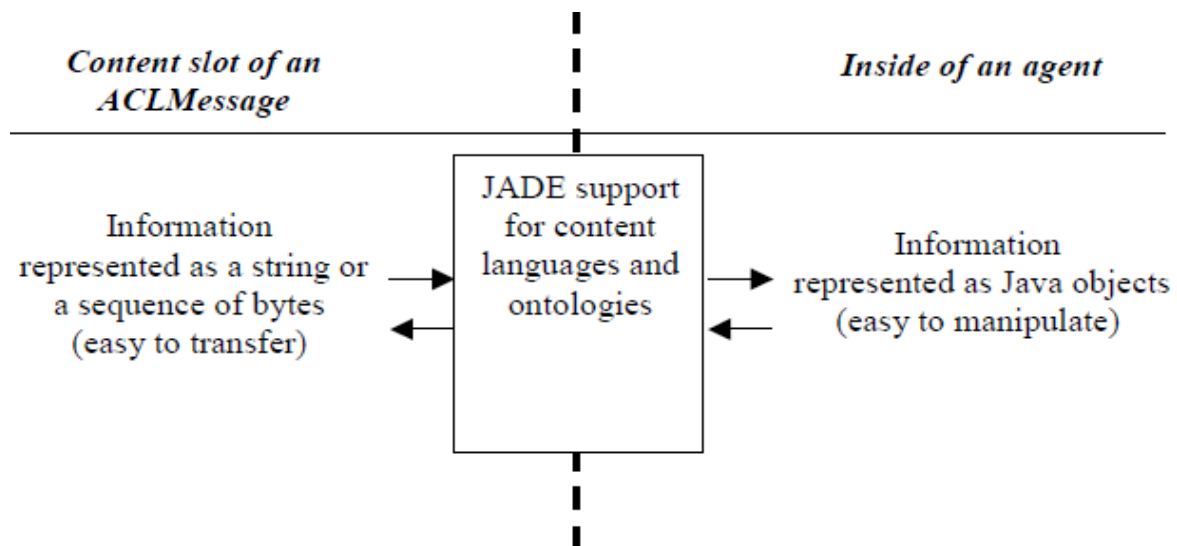


Figure 2.15. The Conversion Performed by the JADE Support for Content Languages and Ontologies

JADE proposes the use of the Content Reference Model (CRM) [Caire and Cabanillas, 04], which is a classification of all possible elements that occur in the discourse domain, to support ontologies. The UML diagram of the CRM presents in the Figure 2.16. The model has the main important elements which are: Predicate, Concept and AgentAction. These are the types which the JADE ontology uses.

¹⁴<http://jade.tilab.com/>

¹⁵<http://www.fipa.org/specs/fipa00008/SC00008I.html>

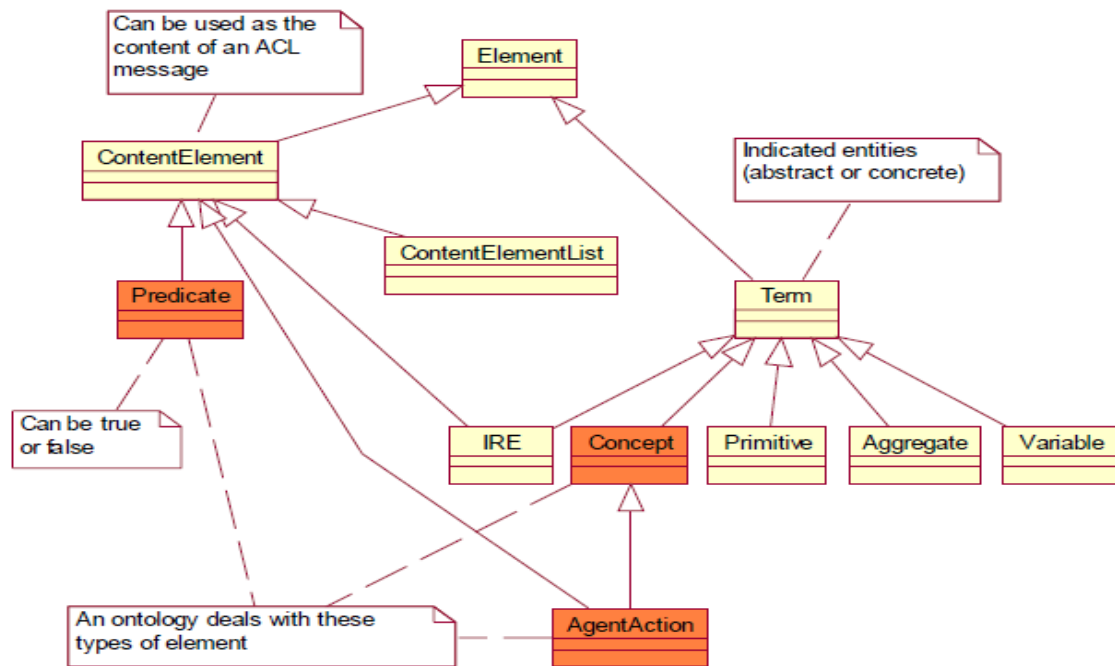


Figure 2.16. Content Reference Model

In detail, the types of elements an ontology deals with are defined as follows:

- *Concepts* are expressions that indicate entities that "exist" and that agents talk and reason about.
- *Predicates* are expressions that say something about the status of the world and usually evaluate to true or false.
- *AgentActions* are expressions that indicate something that can be executed by some agent.

The agent model can be built by protégé ontology editor and then exported to JADE ontology model by protégé bean generator plug-in (section 2.2.5.2).

2.4. Using Optimization Solution for The Multi-Agent Systems in Transport Domain

In the previous chapter, we have discussed a state of the art about the existing multimodal transport information systems. We have illustrated that an information system for multi-modal transports has different composition and integration techniques of information; which are based on a modeling methodology and resolution approach. Therefore, we present in this section, the existing optimization techniques for multi-agent system in the transport domain.

In this section we present a theoretical overview of the optimization algorithms; we start by optimization definitions (2.4.1). Then, we present some optimization methodologies (section 2.4.2). We illustrate the metaheuristics problems in (section 2.4.3). We address the using of the evolutionary algorithms in the transport domain (section 2.4.4). (Section 2.4.5), we present the interest of using of and optimization algorithms in multi-Agent system. Finally, towards a distributed architecture based on optimizing agents in the transport domain will be explained in section 2.2.5.

2.4.1. Optimization Definitions

Optimization is the mathematical discipline which is concerned with finding the maxima and minima of functions f , possibly subject to constraints, called objective function or cost function or optimization criteria function. The need to optimization based on the need to provide the user with a solution that can have the best response on their requirements. The search for the optimum of the function f is modified by a composition of variables, called **decision parameters**.

A mathematical optimization problem, or just optimization problem, has the form:

Minimize $f(\vec{x})$ as $\vec{g}(\vec{x}) \leq 0$ and $\vec{h}(\vec{x}) = 0$ with:

$$\left\{ \begin{array}{l} \vec{x} \in \mathbb{R}^n : n \text{ decision variables} \\ \vec{g}(\vec{x}) \in \mathbb{R}^m : m \text{ inequality constraints} \\ \vec{h}(\vec{x}) \in \mathbb{R}^p : p \text{ equality constraints} \end{array} \right.$$

The constraints of the optimization problem define a restricted search area of the optimal solution. Two types of inequality constraints are defined as follow, (Figure 2.17):

- Constraints of type $B_{i\inf} \leq x_i \leq B_{i\sup}$: the values of \vec{x} that satisfy these constraints define a **research space**. This space is illustrated in Figure 2.15.a, for $n = 2$;
- Constraints of type $c(\vec{x}) \leq 0$ or $c(\vec{x}) \geq 0$: the values of \vec{x} that satisfy these constraints define a **space of realizable values**. space is illustrated in Figure 2.15.b, for $n = 2$;

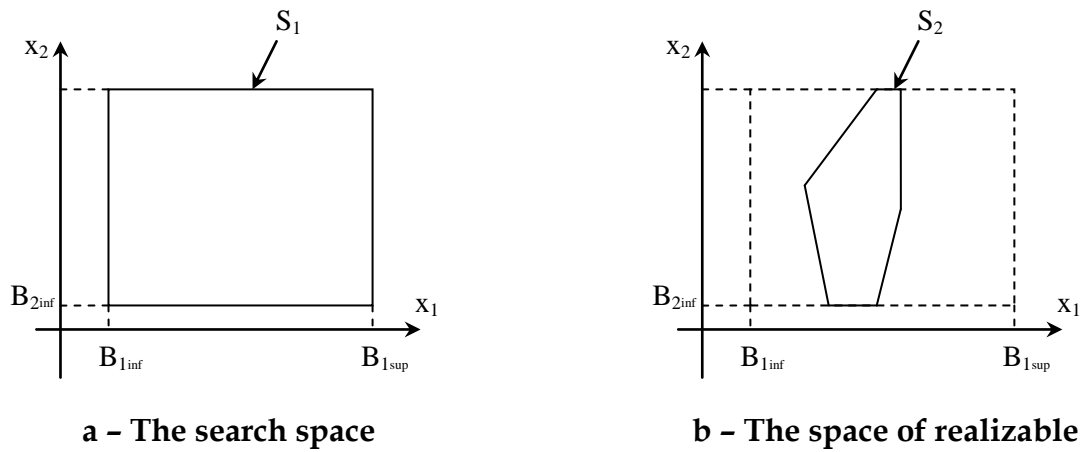


Figure 2.17. The Different Research Spaces

If the objective function f is minimized, then a point \vec{x} is, (Figure 2.18):

- A **global minimum** of the function f , if $f(\vec{x}^*) < f(\vec{x}) \forall \vec{x}$ when $\vec{x}^* \neq \vec{x}$. The point M_3 of the Figure 2.18, corresponds to this definition.
- A **strong local minimum** of the function f , if $f(\vec{x}^*) < f(\vec{x}) \forall \vec{x} \in V(\vec{x}^*)$ when $\vec{x}^* \neq \vec{x}$ and $V(\vec{x}^*)$ define a neighborhood of \vec{x}^* . The points M_2 and M_4 of the Figure 2.18, correspond to this definition.
- A **weak local minimum** of the function f , if $f(\vec{x}^*) \leq f(\vec{x}) \forall \vec{x} \in V(\vec{x}^*)$ when $\vec{x}^* \neq \vec{x}$ and $V(\vec{x}^*)$ define a neighborhood of \vec{x}^* . The point M_1 of the Figure 2.18 corresponds to this definition.

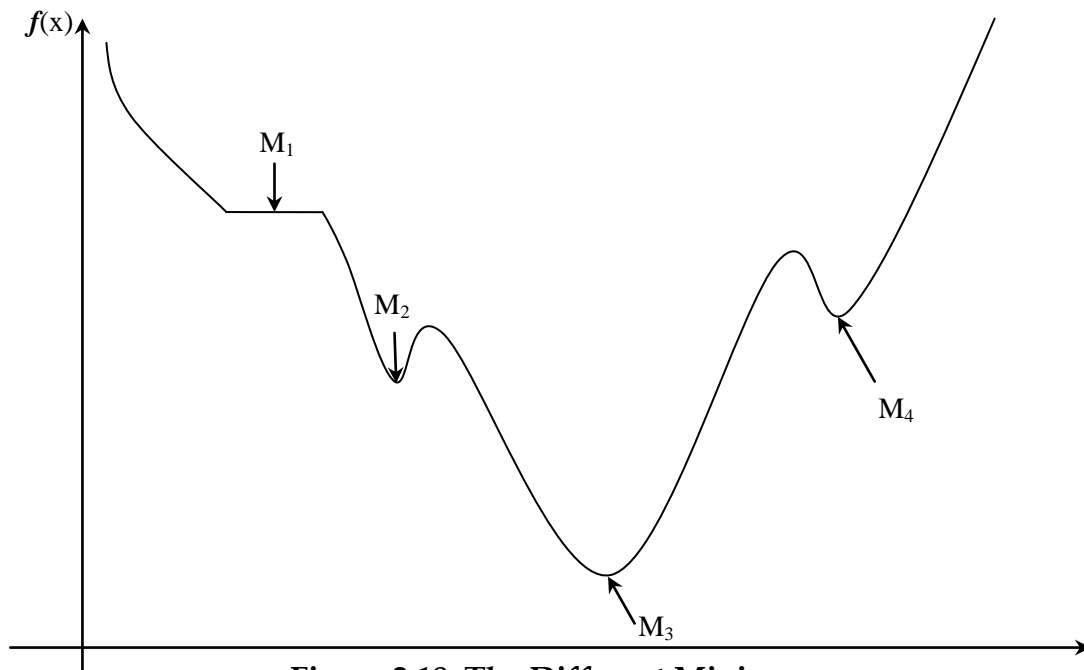


Figure 2.18. The Different Minimum

2.4.2. Optimization Methodologies

2.4.2.1. Classification of The Optimization Problems

There are many optimization algorithms available to the engineer. Many methods are appropriate only for certain types of problems. Thus, it is important to be able to recognize the characteristics of a problem in order to identify an appropriate technique solution. Within each class of problems, there are different minimization methods, varying in computational requirements, convergence properties, and so on. Optimization problems are classified according to the mathematical characteristics of the decision variables, the objective function, and the problem formulation. The classifications of optimization problem are summarized in Table 2.3.

Problem Characteristic			Problem Type
Decision Variable	Number	1	Univariate
		>1	Multivariate
	Type	Continuous real numbers	Continuous
		Integers	Integer or Discrete
		Both continuous real numbers and integers	Mixed Integer
Objective Function	Type	Linear functions of the control variables	Linear
		Quadratic functions of the control variables	Quadratic
		Other nonlinear functions of the control variables	Nonlinear
Problem formulation	Type	Subject to constraints	Constrained
		Not subject to constraints	Unconstrained

Table 2.3. Optimisation Problem Classifications.

2.4.2.2. The Methods for Solving Multi-Objective Optimization Problems

Multi-objective Optimization also known as **multi-criteria** or **multi-attribute** optimization is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints.

Multi-objective optimization problems can be found in various fields: product and process design, finance, aircraft design, the oil and gas industry and automobile design, or wherever optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Maximizing profit and minimizing the cost of a product; maximizing performance and minimizing fuel consumption of a vehicle; and minimizing weight while maximizing the strength of a particular component are examples of multi-objective optimization problems.

Mathematically, a multi-objective optimization problem tries to minimizing a number of objective functions grouped in $\vec{f}(\vec{x})$ when $\vec{g}(\vec{x}) \leq 0$ and $\vec{h}(\vec{x}) = 0$ with:

$$\begin{aligned} \vec{x} &\in \mathbb{R}^n && : n \text{ decision variables} \\ \vec{f}(\vec{x}) &\in \mathbb{R}^k && : k \text{ objectif fonctions} \\ \vec{g}(\vec{x}) &\in \mathbb{R}^m && : m \text{ inequality constraints} \\ \vec{h}(\vec{x}) &\in \mathbb{R}^p && : p \text{ equality constraints} \end{aligned}$$

Several solutions can address a multi-criteria optimization problem, because the goals are often contradictory, i.e., that decrease of an objective leads to increase the other. These solutions are not **optimal** because they do not optimize all the objectives of the problem. In this case, it is interesting to adopt the concept of **compromise**. Compromise solutions optimize a number of goals while degrading the performance on other objectives.

Several methods exist for solving multi-objective optimization problems. These methods are classified into five groups [Collette et al., 02]:

- **The scalar methods:** Also called the naive approach, this method of resolution is most obvious because it aims to convert the multi-objective problem, which need to solve, to an optimization problem with one object, where there are many methods of resolution it. The easiest process way is to get a new objective function based on the summation of all the objective functions, by amplifying each function by a weighting factor.
- **The interactive methods:** They are progressive methods which allow searching only one solution. During the optimization, these methods allow the user to determine their preferences towards choosing between objectives.
- **The fuzzy methods:** These are methods that involve fuzzy logic for solving multi-objective optimization problems. Unlike classical logic, where everything is described in terms of TRUE or FALSE. Fuzzy logic can deal with uncertainty and imprecision of a human knowledge, as well as, with

progressive transitions between states. This logic is based on fuzzy sets which were developed by the automation scientist L. Zadeh.

- **The decision making methods:** These are the only methods for solving multi-objective optimization problems which work only on discrete sets of points which are not based on dominance relations. Those methods generate a set of solutions through established an order relation between the different elements.
- **The metaheuristic methods:** The metaheuristic are a large family of optimization methods designed to solve difficult optimization problems. The optimization problems are concerned both the combinatorial optimization problems with discrete variables as the global optimization problems with continuous variables. These methods are characterized by a high level of abstraction, which allows them to adapt to a wide range of different problems, from a simple local search algorithms for global search one.

2.4.3. The Metaheuristics

In this thesis, we are interested for the metaheuristics, for that, we will illustrate their features in this section. These algorithms are used to solve many complex problems. The metaheuristics describe a method for how we can be solved a difficult problem by applying a heuristic strategy.

Heuristic strategies can also be designed to develop algorithms for optimization problems. An approximation algorithm built on heuristic principles is therefore often denoted a *heuristic procedure* or, in the optimization literature, simply a *heuristic*. While the word *heuristic* means to find or to discover. In optimization, it is so much used to describe how to find as how to *search* for good solutions [Reeves and Beasley, 95].

The metaheuristics do not require knowledge about the problem which we want to solve. It can provide solutions to these problems by setting one or more function (s) and objective (s).

The metaheuristics generate solutions to optimization problems using techniques inspired by natural evolution, where they look for to reproduce the characteristics of diversification and of evolution. The genetic and evolutionary algorithms are inspired by the biology in the evolution individuals and competition among them where the fittest individuals dominating over the weaker ones. (Theory of Darwin: "survival of the fittest.").The best individuals give then descents which would inherit genes of the current population.

The algorithm of colonies of ants takes his sources of the ethnology. It is about adaptation of the behaviour of the ants which search for their food. This algorithm is used to solve the problems of research for the shortest road.

The physic phenomena also enriched the metaheuristics through the simulated annealing issue which based on the observation of the processes in the metallurgy and aims to minimize the function.

1. The Evolutionary Metaheuristics:

The evolutionary algorithms apply the principle of the evolution of the individuals; they are affected by mechanisms of encoding, of memorization, of diversification as well as of a stop criterion.

We detail in what follows the genetic algorithm (GA) which is a part of the most used evolutionary metaheuristics.

The Genetic Algorithms (GAs):

The Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems.

GAs simulate the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

GAs are based on an analogy with the genetic structure and behaviour of chromosomes within a population of individuals using the following foundations:

- Individuals in a population compete for resources and mates.
- Those individuals most successful in each “competition” will produce more offspring than those individuals that perform poorly.
- Genes from “good” individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
- Thus each successive generation will become more suited to their environment.

The most common type of genetic algorithm works like this: a *population* is created with a group of individuals created randomly. The *individuals* in the population are

then evaluated. The *evaluation function* is provided by the programmer and gives the individuals a score based on how well they perform at the given task. Two individuals are then selected based on their fitness, the higher of the fitness, and the higher of the chance of being selected. These individuals then “reproduce” to create one or more offspring, after which the offspring are mutated randomly. This continues until a suitable solution has been found or a certain number of generations have passed, depending on the needs of the programmer. We will explain the different steps of GA in the follow:

- **Selection:**

While there are many different types of selection, we illustrate the most common type - *roulette wheel selection*. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness. Two individuals are then chosen randomly based on these probabilities and produce offspring. Pseudo-code for a roulette wheel selection algorithm is shown below, (Figure 2.19).

```
for all members of population
    sum += fitness of this individual
end for

for all members of population
    probability = sum of probabilities + (fitness / sum)
    sum of probabilities += probability
end for

loop until new population is full
    do this twice
        number = Random between 0 and 1
        for all members of population
            if number > probability but less than next probability
                then you have been selected
            end for
        end
        create offspring
    end loop
```

Figure 2.19. The Pseudo-Code of Roulette Wheel Selection.

While this code is very general and will obviously not compile, it illustrates the basic structure of a selection algorithm. Besides, you should write the code yourself, you learn better that way.

- **Crossover**

So, now, we have selected the individuals, and we know that we are supposed to somehow produce offspring with them, but how should we go about doing it?

The most common solution is something called *crossover*, and while there are many different kinds of crossover, the most common type is *single point crossover*. In single point crossover, we choose a locus at which we swap the remaining alleles from one parent to the other. This is complex and is best understood visually.

Parent 1: **1011010|010100110**

Parent 2: **0011010|110110101**

Child 1: **1011010|110110101**

Child 2: **0011010|010100110**

As we can see, the children take one section of the chromosome from each parent. The point at which the chromosome is broken depends on the randomly selected crossover point. This particular method is called *single point crossover* because only one crossover point exists. Sometimes only child 1 or child 2 is created, but oftentimes both offspring are created and put into the new population. Crossover does not always occur. Sometimes, based on a set probability, no crossover occurs and the parents are copied directly to the new population. The probability of crossover occurring is usually 60% to 70%.

- **Mutation**

After selection and crossover, we have a new population full of individuals. Some are directly copied, and others are produced by crossover. In order to ensure that the individuals are not all exactly the same, The GAs allow for a small chance of mutation. They loop through all the alleles of all the individuals, and if that allele is selected for mutation, they can either change it by a small amount or replace it with a new value. The probability of mutation is usually between 1 and 2 tenths of a percent. A visual for mutation is shown below.

Before: **1101101001101110**

After: **110110**0**001101110**

As we can easily see, mutation is fairly simple. We just change the selected alleles based on what we feel is necessary and move on. Mutation is, however, vital to ensuring genetic diversity within the population.

2. Local Search based the Metaheuristics:

The basic principle of local search is that solutions are successively changed by performing moves which alter solutions *locally*. Valid transformations are defined by neighborhoods which give for a solution all neighboring solutions that can be reached by one move.

Moves must be evaluated by some *heuristic measure* to guide the search. Often one uses the implied change of the objective function value, which may provide reasonable information about the (local) advantage of moves. Following a greedy strategy, *steepest descent (SD)* corresponds to selecting and performing in each iteration the best move until the search stops at a local optimum. As the solution quality of local optima may be unsatisfactory. Mechanisms are developed to guide the search to overcome local optimality. A simple strategy called iterated local search is to iterate/restart the local search process after a local optimum has been obtained, which requires some perturbation scheme to generate a new initial solution.

A variable way of handling neighborhoods is still a topic within local search. Consider an arbitrary neighborhood structure N , which defines for any solution s a set of neighbor solutions $N_1(s)$ as a neighborhood of depth $d = 1$. In a straightforward way, a neighborhood $N_{d+1}(s)$ of depth $d + 1$ is defined as the set $N_d(s) \cup \{s_- \mid \exists s_- \in N_d(s): s_- \in N_1(s_-)\}$. In general, a large d might be unreasonable, as the neighborhood size may grow exponentially.

However, depths of two or three may be appropriate. Furthermore, temporarily increasing the neighborhood depth has been found to be a reasonable mechanism to overcome *basins of attraction*, e.g., when a large number of neighbors with equal quality exist.

Finally, this local search is performed by using various methods, the most basic: *Simulated Annealing* method and *Tabu Search* method.

❖ **Simulated Annealing(SA):**

Simulated annealing (SA) extends basic local search by allowing moves to inferior solutions, [Kirkpatrick et al., 83], [Dowsland, 93]. The basic algorithm of SA may be described as follows: Successively, a candidate move is randomly selected; this move is accepted if it leads to a solution with a better objective function value than the current solution, otherwise the move is accepted with a probability that depends on the deterioration Δ of the objective function value. The probability of acceptance is computed as $e^{-\Delta/T}$, using a temperature T as control parameter.

Various prepositions describe a robust concretization of this general SA procedure. Following [Johnson et al, 98], the value of T is initially high, which allows many inferior moves to be accepted, and is gradually reduced through multiplication by a parameter *cooling Factor* according to a geometric cooling schedule. At each temperature $sizeFactor \times |N|$ move candidates are tested ($|N|$ represent the current neighborhood size and *sizeFactor* an appropriate parameter). The starting temperature is determined as follows: Given a parameter *initialAcceptanceFraction* and based on an abbreviated trial run, the starting temperature is set so that the fraction of accepted moves is approximately *initialAcceptanceFraction*. A further parameter, *frozenAcceptanceFraction* is used to decide whether the annealing process is *frozen* and should be terminated. Every time a temperature is completed with less than *frozenAcceptanceFraction* of the candidate moves accepted, a counter is increased by one. The procedure is terminated when no new best solution is found for a certain value of this counter.

For a different concretization of SA see [Ingber, 96]. An interesting variant of SA is to strategically reheat the process, i.e., to perform a non-monotonic acceptance function. Successful applications are provided, e.g., in [Osman,95]. *Threshold accepting*, [Dueck and Scheuer, 90], is a modification of SA with the essential difference between the two methods being the acceptance rules. Threshold accepting accepts every move that leads to a new solution which is 'not much worse than the older one.

❖ Tabu Search :

The basic paradigm of *tabu search* (TS) is to use information about the search history to guide local search approaches to overcome local optimality [Glover and Laguna, 97]. In general, this is done by a dynamic transformation of the local neighborhood. Based on some sort of memory certain moves may be forbidden, which are set tabu. As for SA, the search may lead to performing deteriorating moves when no improving moves exist or all improving moves of the current neighborhood are set tabu. At each iteration a best admissible neighbor may be selected. A neighbor, respectively a corresponding move, is called admissible, if it is not tabu or if an aspiration criterion is fulfilled.

Various TS methods are described which differ especially in the way in which the tabu criteria are defined. An aspiration criterion may override a possibly unreasonable tabu status of a move.

The most commonly used TS method is based on a *recency-based* memory that stores moves, more exactly move attributes, of the recent past. The basic idea of such

approaches is to prohibit an appropriately defined inversion of performed moves for a given period.

Strict TS use the idea of preventing cycling to formerly traversed solutions. The goal is to provide necessity and sufficiency with respect to the idea of not revisiting any solution. Accordingly, a move is classified as tabu if it leads to a neighbor that has already been visited during the previous part of the search, [Glover, 90].

Reactive TS aims at the automatic adaptation of the tabu list length of static TS [Battiti, 96]. The idea is to increase the tabu list length when the tabu memory indicates that the search is revisiting formerly traversed solutions. A possible specification can be described as follows: Starting with a tabu list length L of 1 it is increased every time a solution has been repeated. If there has been no repetition for some iterations, we decrease it appropriately. To accomplish the detection of a repetition of a solution, one may apply a trajectory based memory using hash codes as for strict TS.

2.4.4. The Evolutionary Algorithms in the Transport Domain

Through their various advantages in solving optimization problems, the AGs knew a real development in the general scheduling domain, and in particular in the scheduling of the production systems. Indeed, [Pierreval 03] presents a study of different evolutionary approaches in the scheduling of production, Design of production facilities, and the assembly systems.

For the air traffic control, [Hansen, 03], address this domain by applying the evolutionary approaches. In fact, [Ciesielski, 98] proposed an evolutionary approach for scheduling the landing time of aircraft with limited number of the on a limited number of tracks. According to Ciesielski, the corresponding coding illustrates, for each aircraft, the track where it should land (Number of periods of 30 seconds) following the current time.

The Figure 2.20 shows an example for this coding. We can deduce that the aircraft 1 will land at 12:00 on track 0, the aircraft 2 will land at 12:03 on track 1, the schedule landing of the aircraft 3 is 12:04:30 on track 0, etc. The hourly scheduling of landing of aircraft which used by Ciesielski is made at a real time.

1		2		3		4		...	
0	0	6	1	9	0	12	0

Figure2.20. Example of Coding of the Scheduling Landings.

Ngamchai, [Ngamchai, 00], presents a genetic algorithm to design lines of a bus network with correspondence nodes location. In fact, Ngamchai has presented an individual as a collection of paths. Each path represents a road and the number of paths in each individual represents the number of vehicles.

In, [Jianming et al., 05], proposed a transit network optimization model. Jianming illustrated an optimization model of headways for all the transit routes in the optimized network. Since all the two models can be boiled down to the NP-hard problem, two kinds of evolutionary algorithms. Indeed, the ant colony algorithm and improved genetic algorithm are introduced to solve the problems respectively.

Finally, the traveling salesperson problem (TSP) is a well known important optimization problem. The goal is to find the shortest tour that visits each city in a given list exactly once and then returns to the starting city. Formally, the TSP can be stated as follows. The distances between n cities are stored in a distance matrix D with elements d_{ij} where $i, j = 1 \dots n$ and the diagonal elements d_{ij} are zero. A *tour* can be represented by a cyclic permutation π of $\{1, 2, \dots, n\}$ where $\pi(i)$ represents the city that follows city i on the tour. The traveling salesperson problem is then the optimization problem to find a permutation π that minimizes the length of the tour denoted by:

$$\sum_{i=1}^n d_{i,\pi(i)}$$

For this minimization task, the tour length of $(n - 1)!$, permutation vectors have to be compared. This results in a problem which is very hard to solve and in fact known to be NPcomplete [Johnson and Papadimitriou, 85]. However, solving TSPs is an important part of applications in many areas including vehicle routing, computer wiring, machine sequencing and scheduling, frequency assignment in communication networks ([Lenstra and Kan, 1975], [Punnen, 02]).

2.4.5. Multi-Agent System and Optimization

There has been considerable recent interest in the analysis of large-scale networks, such as the Internet, which consist of multiple agents with different objectives. For such networks, it is essential to design resource allocation methods that can operate in a decentralized manner with limited local information and rapidly converge to an approximately optimal operating point. Most existing approaches use the distributed optimization frameworks under the assumption that each agent has an objective function that depends only on the resource allocated to that agent. In many practical situations however, individual cost functions or performance measures depend on the entire resource allocation vector.

[Nedic and Ozdaglar, 09] was interested in a distributed computation model for a multi-agent system, where each agent processes his/her local information and shares the information with his/her neighbours. To describe such a multi-agent system, we need to specify two models: an information exchange model describing the evolution of the agents' information in time and an optimization model specifying overall system objective that agents are cooperatively minimizing by individually minimizing their own local objectives.

In [Davidsson et al., 08], has compared the strengths and weakness of agent-based approaches and mathematical optimization techniques for sources allocation. The comparison indicated that properties of the tow approaches were complementary and that it could be advantageous to combine them. In a case study, the hybrid approaches tested thought as follow:

- In the first approach, optimization was embedded in the agents to improve their ability to make good decision.
- In the second approach, optimization was used for creating long-term course plans which were redefined dynamically by agents.

Soft agent technology and optimization can be combined in various ways. For example, the usage of the optimization algorithm inside the multi-agent system based simulation tool for the purpose of realistic modelling. The idea is that is some cases decision makers can be represented more accurately by using optimization.

Another possibility is to let software agent optimize some system. It must be stressed that a system sometimes can be optimized also without the use of advance optimization algorithm. This is the case of the [Karageorgos et al, 03] approach which explain the application of virtual collaboration networks. In this approach, agents were used for optimizing by using agent-based negotiation, based on the nested contract net protocol.

Finally, an approach provided by [Dorer and Clisti, 05], which utilized agent technology for solving real world dynamic product planning problems, where the orders were allocated to trucks. The problem was solved by means of neighborhood search algorithm and auction-based negotiation for gradually improving the solution.

2.4.6. Towards a Distributed Architecture Based on Optimizing Agents in the Transport Domain

In the domain of transport, several models are based on MASs for solving the following problems:

- Simulation of automobile traffic [Mandiau et al., 02b] ;
- Regulation of multimodal transport networks [Fayech, 03] [OuldSidi, 06];
- Optimization, research and composition of multimodal routes, [Zidi, 06] [Kamoun, 07] [Zgaya, 07] ;
- Monitoring of automated vehicles [Durfee et al.,91] ;
- Air traffic control [Ljungberg et al., 92];
- Systems management support of military aircraft [Chaib-draa, 95];
- Urban traffic management [Chaib-draa, 96];

Indeed, in MAS, two modes of communication exist: an explicit mode of communication based on protocols for consisting of high-level semantic concepts, and a stigmergy mode of communication which is based on the environment. First mode is essential for communication between agents, but not always sufficient to coordinate the activity of decentralized MAS. The agents involved in solving the problem of communications must find a good agreement between resolution efficiency and the resources consumers, including the time of resolution and computing of resources.

We have already integrated within our team, the optimization methods in SMAs to solve transportation problems such as ant colony algorithms for reconfiguration problems of transport networks [Zidi, 06] or genetic algorithms for traffic control [Fayech, 03], [Kamoun, 07] [Zgaya, 07] .

The problem of regulation of multimodal transport networks requires not only the reorganization of racing vehicles, but also the monitoring of transport networks and diagnosis of incidents. For That, Fayech [Fayech, 03] proposed a decision support system based on an evolutionary approach to the temporal regulation or spatio-temporal, in unusual situations.

Zidi, [Zidi, 06] proposed a system to solve the problem of aid the mobility in the multi-modal transport networks which based on graphical modelling. He used a multi-criteria method of route search based on hybridization between their modified Dijkstra algorithm and a genetic algorithm to find a minimum population of paths.

Kamoun [Kamoun, 07] presented a cooperative information system to aid the urban mobility. He applied the tow level of optimization method: the first optimization permitted, compared with the use's requests, to limit the domain of research and to

define a set of information agents. Then, the second optimization is to search the best global route, corresponding to the same use query by interviewing different agents.

Finally, Zgaya [Zgaya, 07] designed multimodal transport information system to optimize the management of the data flow of the users' requests, which can be simultaneous and numerous. The system uses the mobile agents (MA) to retrieve information from multimodal transport network. She proposed an optimization solution which operates in two levels. The first one aims to optimize the number of MAs in order to explore the whole Multimodal Network (MN) respecting initial routes, which are provisional. Final routes are deduced every time a set of customers' simultaneous requests. His system optimizes the selection of nodes to answer to formulated requests. Selected nodes are chosen according to the total resulted cost and response delay corresponding to generated responses.

2.5. The Role of Ontologies in Multi-Agents Systems (MAS)

After having seen the role of optimization algorithms in MAS, now, we will illustrate the role of ontologies in MAS. In fact, research in agent communication languages, such as KQML and FIPA ACL, has embraced the notion of ontology to enable effective information exchange between agents. Recently, ontologies and agents are two research areas that have become intertwined. Ontologies have started to be developed aiming at agent-based applications. Agents have benefited by the use of ontologies in heavily information-based processes.

Within MAS, agents are characterized by different views of the world that are explicitly defined by ontologies. These views of what the agent recognizes to be the concepts describing the application domain which is associated with the agent together with their relationships and constraints [Falasconi et al., 96]. Interoperability between agents is achieved through the reconciliation of these views of the world by a commitment to common domain ontologies that permit agents to interoperate and cooperate while maintaining their autonomy. Indeed, the use of a common domain ontology guarantees the consistency (an expression has the same meaning for all the agents) and the compatibility (a concept is designed, for the same expression, for any agent) of the information present in the system.

But, in open systems, agents are associated with knowledge sources which are diverse in nature and have been developed for different purposes. Knowledge sources embedded in a dynamic environment can join and leave the system at any time. For that, in all types of agent's communication, the ability to share information is often prevented because the meaning of information can be hardly affected by context in which it is viewed and interpreted [Ciocoiu et al., 01], and the ability to

share information may be hard due to the impossibility to have a unique ontology for each application domain.

Usually, each application (or each agent) has its specific, private ontology and it may not fully understand other agent's ontology. Even in similar domains there are both syntactic and semantic differences between ontologies (section 2.6). Different people have a different vision of the world, and consequently people or agents may use different terms for the same meaning or may use the same term to mean different things. Successful exchange of information means that the agents understand each other. The interoperability problem happens when we have heterogeneous and distributed systems.

Finally, ontologies are widely used, not only in agent-based applications. Nevertheless, they provide specific benefits for agent applications. [Bermejo-Alonso, 06] summarized these benefits as follows:

1. Ontologies clarify the structure of knowledge: By performing an ontological analysis of a domain allows defining an effective vocabulary, assumptions and the underlying conceptualisation.
2. Ontologies help in knowledge scalability: knowledge analysis can result in large knowledge bases. Ontologies help to encode and manage in a scalable way.
3. Ontologies allow knowledge sharing and reuse: by associating terms with concepts and relationships in the ontology as well as syntax for encoding knowledge in them, ontologies allow further users and agents to share and reuse such knowledge.
4. Ontologies increase the robustness of an agent-based system: agents can draw on ontological relationships and commitments to reason about novel or unforeseen events in their domain.
5. Ontologies, that have as focus the domain of software engineering of agent-based systems, do help development teams and software processes, and may even render useful during exploitation phases as a foundation of cognitive understanding and integration of agents including cognitive self-reflection capabilities
6. Ontologies provide a foundation for interoperability among agents.

After have presenting the benefits of ontology for agent's application. In the next section, we illustrate the different semantic problems arise from heterogeneous of the schemas and ontologies supporting the different MAS.

2.6. Semantic Heterogeneity and Interoperability in MAS

MAS are heterogeneous if they have different characteristics. For that, the difficult problems in design of MAS, which facilitate *interoperation* and *mediation* among information sources and their consumers, arise from the presence of semantic heterogeneity among the schemas and ontologies supporting the different services.

In determining, whether MAS are heterogeneous one can focus on different characteristics, yielding different types of heterogeneity that have to be dealt in the context of interoperability. These types are usually classified to be of *syntactical*, *structural* and *semantical* nature, [Bouquet et al., 04], [Obrst, 03]:

- **Syntactic level:** At this level, all forms of heterogeneity depend on the choice of the formalism used for representing information. Therefore, the different formats can be interoperated at a syntactic level; this is typically achieved through a translation function.
- **Structural level:** At this level, all mismatches related to differences in the arrangement of concepts and their relationships.
- **Semantic level:** At this level, all discrepancies have to do with the fact that the same real world is represented using different denotations or structures, and so can be interpreted differently.

Agents may be heterogeneous in many degrees. In principal, it is better to assume that agents are heterogeneous in every aspect than to assume that agents share some common features. To solve the agent heterogeneity, agents need increasingly more explicit, machine interpretable semantics. Semantic interoperability solutions aim to provide a knowledge-level interoperability that enable cooperation between communities with the ability to bridge semantic conflicts that are arising from differences in implicit meanings, perspectives, and assumptions.

Heterogeneity can be regarded as an advantage as well as a disadvantage by system designers. On one hand, heterogeneity is positive because it is closely related to system efficiency. On the other hand, heterogeneity in data and knowledge systems is considered as a problem since it is an important barrier for their interoperation. Thus, the lack of standards is the main obstacle to the exchange of data between heterogeneous systems [Visser et al., 97].

In point of fact, agents may use different ontologies to represent their views of a domain which can be leading to possible ontology mismatches. [Wache et al., 01] present three different way of how we can employ the ontologies. In general, three different directions can be identified: *single ontology approach*, *multiple ontologies*

approach and *hybrid ontologies approach*. We summarized the different approaches as follow:

- **Single Ontology Approach (Figure 2.21.)** uses a one global ontology providing a shared vocabulary for the specification of the semantics. All information sources are related to a one global ontology. The global ontology can also be a combination of several specialized ontologies. Single ontology approach can be applied to integration problems where all information sources to be integrated provide nearly the same view on a domain.

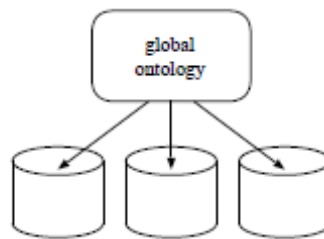


Figure 2.21. Single Ontology Approach

- **Multiple Ontologies Approach (Figure 1.22.).** In this approach, each information source is described by its own ontology. In principle, the “source ontology” can be a combination of several other ontologies but it cannot be assumed that the different “source ontologies” share the same vocabulary.

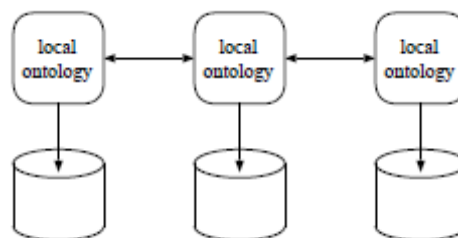


Figure 2.22. Multiple Ontologies Approach

- **Hybrid Ontology Approach (Figure 1.23.)** is similar to multiple ontology approach where the semantics of each source is described by its own ontology. But in order to make the source ontologies comparable to each other they are built upon one global shared vocabulary. The shared vocabulary contains basic terms (the primitives) of a domain. In hybrid approaches the interesting point is how the local ontologies are described, i.e. how the terms of the source ontology are described by the primitives of the shared vocabulary.

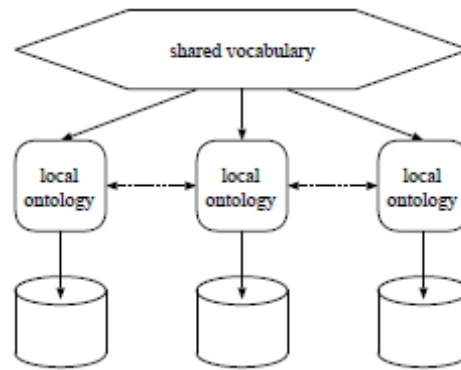


Figure 2.23. Hybrid Ontologies Approach

In this thesis we are dealing with the hybrid ontology approach where each information provider, register in our system, explores its own ontology. We chose this approach because the overall system architecture is an open MAS and we developed a global ontology (shared vocabulary) to make the different local ontologies comparable with each other.

2.7. Ontologies and Combinations Problems

In previous section we have seen the role of ontology in the multi-agent systems. Where in open multi-agent systems, communication problems that arise from heterogeneous ontologies should be solved, rather than avoided. In order to achieve a better understanding of the mismatches, we will summarize those heterogeneous ontologies problems. Firstly, we explain the meaning of terminology in section (2.7.1). Then, we will give an overview of the mismatches which might occur between different ontologies, based on the work by [Klein, 01], in Section (2.7.2). Then, we survey the ontology mapping, merging, and alignment and we will present a number of representative approaches for ontology mapping, merging, and alignment in Section (2.7.3).

2.7.1. Terminology

Before we present the problems that play an important role in the ontology combinations problems, we explain some of the terminology which used to describe the operations between ontologies [Klein, 01]. Those terminologies used to avoid misunderstandings; we present the main definitions of the terms used throughout this thesis.

- **Combining:** Using two or more different ontologies for a task in which their mutual relation is relevant.

- **Merging, Integrating:** Creating a new ontology from two or more existing ontologies with overlapping parts, which can be either virtual or physical.
- **Aligning:** Bring two or more ontologies into mutual agreement, making them consistent and coherent.
- **Mapping:** Relating similar (according to some metric) concepts or relations from different sources to each other by an equivalence relation. Mappings result in a virtual integration.
- **Articulation:** The points of linkage between two aligned ontologies.
- **Translating:** Changing the representation formalism of ontology while preserving the semantics.
- **Transforming:** Changing the semantics of ontology slightly to make it suitable for purposes other than the original one.
- **Version:** The result of a change that may exist next to the original.
- **Versioning:** A method to keep the relation between newly created ontologies, the existing ones, and the data that conforms to them consistent.

2.7.2. Ontology Mismatching

What are the types of differences between the ontologies that can affect agents? According to [Klein, 02], there are different categorizes of mismatches (heterogeneity), as we will see in the next sections. Figure (2.24):

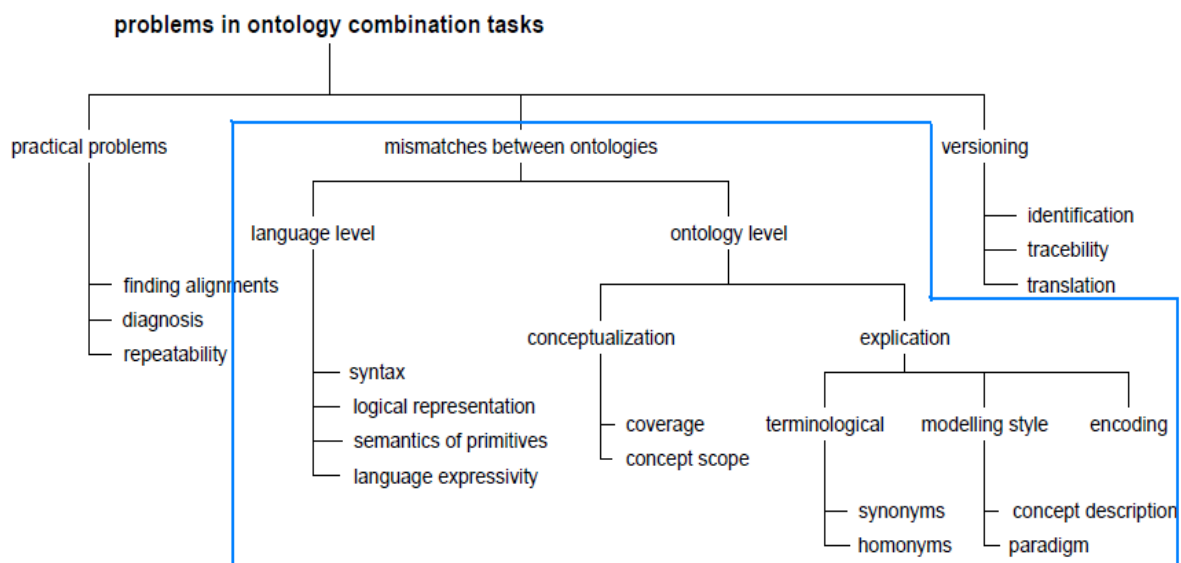


Figure 2.24. Ontology Mismatching

2.7.2.1. *Language Level Mismatches*

Mismatches at the language level occur when ontologies written in different ontology languages are combined. We distinguish four types of mismatches that can occur:

- **Syntax:** This mismatch occurs when different ontology languages often use different syntaxes. This difference is probably the simplest kind of mismatch. However, this mismatch often doesn't come alone, but is coupled with other differences at the language level.
- **Logical representation:** The mismatch at this level is the difference in representation of logical notions. The point here is not whether something can be expressed but which language constructs should be used to express something. Also, this mismatch is not about the representation of *concepts*, but about the representation of *logical notions*. This type of mismatch is still relatively easy solvable.
- **Semantics of primitives:** A more possible difference at the meta-model level is the semantics of language constructs. Although, sometimes the same name is used for a language construct in two languages, the semantics may differ.
- **Language expressivity** The mismatch at the meta-model level with the most impact is the difference in expressivity between two languages. This difference implies that some languages are able to express things that are not expressible in other languages. This type of mismatch has probably the most impact.

The list of differences at the language level can be seen as more or less compatible with the broad term “language heterogeneity” of [Visser et al., 97]. Indeed, if the ontologies are not represented in the same languages, a *translation* between sources ontologies to the same language is required.

2.7.2.2. *Ontology Level Mismatches*

Mismatches at the ontology level happen when two or more ontologies that describe overlapping domains are combined. These mismatches may occur when the ontologies are written in the same language, as well as when they use different languages. In fact, several types of mismatches at the model level can be observed. [Visser et al., 97] make a useful distinction between mismatches in the *conceptualization* and *explication* of ontologies. [Wiederhold, 94] mentions the problems with *synonym terms* and *homonym terms*. [Chalupsky, 00] lists four types of

mismatches in ontologies which are, *inference system bias, modelling conventions, coverage and granularity and paradigms*.

[Klein, 02] relate the different types of mismatches that are distinguished by the authors as follow:

- **Conceptualization mismatches:** where the detection and reconciliation of conceptualization differences usually requires the knowledge of a domain expert.
 - *Concept scope:* Two classes seem to represent the same concept, but do not have exactly the same instances, although these intersect. This is called a class mismatch.
 - *Model coverage and granularity:* This is a mismatch in the things that are contained in the ontology. There are three dimensions for model coverage. A first dimension is the extent of the model. A second dimension is the granularity of the model. Finally, there is the perspective of the ontology which determines what aspects of a domain are described. Models can be different in each of these dimensions. For example, an ontology about public transport might or might not include taxis (difference in extent), might distinguish many different types of trains or not (difference in granularity), and could describe technical aspects or functional aspects (difference in perspective).
- **Explication mismatches:** which are mismatches in the way a conceptualization is specified:
 - *Paradigm:* Different paradigms can be used to represent concepts such as time, action, plans, etc. The use of different “top-level” ontology is also an example of this kind of mismatch.
 - *Concept description:* This type of differences is called modelling conventions. Several choices can be made for the modelling of concepts in the ontology. For example, a distinctions between two classes can be modeled using a qualifying attribute or by introducing a separate class.
- **Terminological mismatches :** which mention the problems when two concepts are equivalent:
 - *Synonym terms:* Concepts can be represented by different names. A trivial example is the use of the term “car” in one ontology and the term “automobile” in another ontology. This type of problem is called

term mismatch. Usually these problems coincide with semantic problems and require a lot of human effort.

- *Homonym terms* The meaning of a term is different in another context. For example, the term “conductor” has a different meaning in a music domain than in an electric engineering domain. This is called concept mismatch.
- *Encoding Values* in the ontologies may be encoded in different formats. For example, a date may be represented as “dd/mm/yyyy” or as “mm-dd-yy”. There are many mismatches of this type, but these are all very easy to solve. In most cases, a transformation step or wrapper is sufficient to eliminate all those differences

The overview above illustrates that there are many aspects in which ontologies can differ. In principle, all these difference can occur between different ontology versions, although some mismatches are more likely to happen than others.

2.7.3. Ontology Mapping, Merging and Alignment

Because ontologies are shared specifications, the same ontologies can be used for the annotation of multiple data sources, not only Web pages, but also collections of XML documents, relational databases, etc. The use of such shared terminologies enables a certain degree of inter-operation between these data sources. It can be expected that many between these ontologies have to be reconciled. The reconciliation of these differences is called *Ontology Mediation*.

Ontology mediation is a broad field of research which is concerned with determining and overcoming differences between ontologies in order to allow the reuse of such ontologies, and the data annotated using these ontologies, throughout different heterogeneous applications [Bruijn et al., 06].

Ontology mediation can be subdivided into three areas: *ontology mapping*, which is mostly concerned with the representation of correspondences between ontologies; *ontology alignment*, which is concerned with the (semi-)automatic discovery of correspondences between ontologies; and *ontology merging*, which is concerned with creating a single new ontology, based on a number of source ontologies.

2.7.3.1. Ontology Mapping

An ontology mapping is a (declarative) specification of the semantic overlap between two ontologies; it is the output of the mapping process (see Figure 2.25). The correspondences between different entities of the two ontologies are typically

expressed using some axioms formulated in a specific mapping language. The three main phases for any mapping process are:

- Mapping discovery
- Mapping representation
- Mapping exploitation/execution.

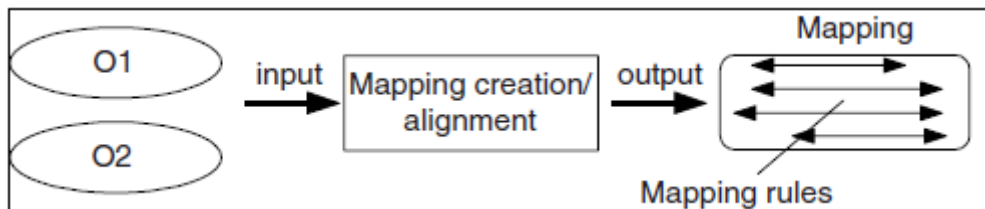


Figure 2.25. Ontology Mapping & Alignment

2.7.3.2. *Ontology Alignment*

Ontology alignment is the process of discovering similarities between two source ontologies. The result of a matching operation is a specification of similarities between two ontologies. Ontology alignment is generally described as the application of the so-called Match operator [Rahm & Bernstein, 01]. The input of the operator is a number of ontology and the output is a specification of the correspondences between the ontologies.

2.7.3.3. *Ontology Merging*

Ontology merging is the creation of one ontology from two or more source ontologies. The new ontology will unify and in general replace the original source ontologies. We distinguish two distinct approaches in ontology merging.

- **First approach:** In this approach, the input of the merging process is a collection of ontologies and the outcome is one new, merged, ontology which captures the original ontologies (see Figure 2.26.(a)). A prominent example of this approach is PROMPT [Noy & Musen, 00b], which is an algorithm and a tool for interactively merging ontologies.
- **Second approach:** In this approach, the original ontologies are not replaced, but rather a 'view', called bridge ontology, is created which imports the original ontologies and specifies the correspondences using bridge axioms. OntoMerge [Dou et al., 02] is an example of this approach. Figure 2.26.(b)

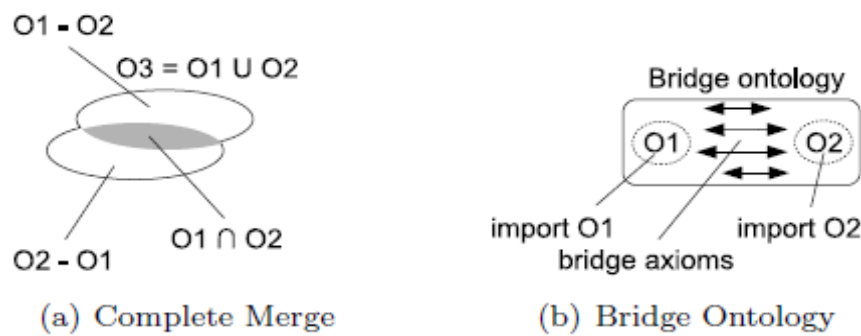


Figure 2.26. Ontology Merging

2.7.3.4. *Ontology Mapping Systems*

In this section we survey a number existing approaches for ontology mapping that combine several of the techniques presented before according to specific aims or domains features, with a focus on the mapping representation aspect.

PROMPT [Noy & Musen, 00a] is a semi-automatic algorithm and an interactive tool for ontology merging and alignment. The central element of PROMPT is the algorithm which defines a number of steps for the interactive merging process; firstly, it identifies merge candidates based on class-name similarities. Then, the result is presented to the user as a list of potential merge operations. The user chooses one of the suggested operations from the list or specifies a merge operation directly. After that, PROMPT performs the requested action and automatically executes additional changes derived from the action. Finally, PROMPT creates a new list of suggested actions for the user, based on the new structure of the ontology, determines conflicts introduced by the last action, finds possible solutions to these conflicts and displays these to the user.

PROMPT is a product of the Stanford Medical Informatics (SMI) Lab and is available as a plug in for Protégé-2000(section 2.6.2), the ontology editor by the same lab. The knowledge model is frame based and Open Knowledge Base Connectivity (OKBC) compatible. PROMPT is based on an extremely general knowledge model and therefore can be applied across various platforms. Their formative evaluation showed that a human expert followed 90% of the suggestions that PROMPT generated and that 74% of the total knowledge-base operations invoked by the user were suggested by PROMPT.

Anchor-PROMPT [Noy & Musen, 00b] is also a product of (SMI) Lab and is OKBC compatible. It augments the earlier PROMPT algorithm. The main difference between PROMPT and Anchor-PROMPT is that, in the latter, anchors (or related concepts) are used to establish a link between common terms in the source

ontologies. The user can input the set of anchors or these terms can be automatically identified through lexical matching (based on the use of a lexicon to identify similar words). Then it refines the input relations based on the ontology structures and user feedback. Finally, based on the frequency counts and user feedback, the algorithm determines new matching candidates.

Anchor-PROMPT is also a plug-in to Protégé-2000 and can import and export ontologies in a wide variety of ontology languages. In the evaluation tests on merging ontologies developed independently by different group of researchers, 75% of the results produced by Anchor- PROMPT were deemed correct by experts. The Anchor- PROMPT algorithm produces good results only if ontology developers link the concepts in a similar fashion even though different names are assigned to them. Information related to conflict resolution is not provided.

Chimaera [McGuinness et al., 2000] is a merging and diagnostic Web-based browser ontology environment to support users in creating and maintaining distributed ontologies on the Web. Chimaera takes knowledge base source files as its input prior to merging them into a new or existing knowledge base. The source files can be in a wide variety of different source languages thus providing syntactic interoperability. In contrast to Prompt , the Chimaera environment supports the creation and editing of disjoint partition information, allows bringing together of ontologies built using different formalisms like Knowledge Interchange Format and OKBC. Chimaera only addresses the merging of child concepts, parent concepts and attributes of concepts. The merging and evaluation consists of a name resolution list generation and taxonomy resolution list generation.

Quick Ontology Mapping (QOM) [Ehrig & Staab, 04b] is a successor of the NOM system. QOM is an efficient method for identifying mappings between two ontologies because it has lower run-time complexity of the identification of similarities between two ontologies. QOM based on the ontology features and it uses heuristics (e.g., similar labels) to lower the number of candidate mappings. The actual similarity computation is done by using a wide range of similarity functions, such as string similarity. Several of such similarity measures are computed, which are all input to the similarity aggregation function, which combines the individual similarity measures. QOM first iterates to find mappings based on lexical knowledge and then iterates to find mappings based on knowledge structures. The output of one iteration can be used as part of the input in a subsequent iteration of QOM in order to refine the result. After a number of iterations, the actual table of correspondences between the ontologies is obtained.

	Automation	String-Based	Constraint-based	Taxonomy-based	Linguistic-Resources
PROMPT	Semi	Heuristics			
Anchor-PROMPT	Semi	Heuristics		Heuristics	
Chimaera	Semi	Heuristics			
QOM	Full	Heuristics	heuristics	Heuristics	Heuristics

Table.2.4. Summary of Ontology Mapping Systems

The most popular ontology matching techniques are string-based, constraint-based, Taxonomy-based as well as linguistic resources techniques. Table 2.4, shows that heuristic is widely implemented for carrying out string-based, constraint-based, and linguistic resources probabilistic reasoning approach, also play a part in the execution of taxonomy-based technique, whereas semantic reasoning is the dedicated approach used to execute the model.

2.8. Ontology Negotiation Protocol between the Agents Community

2.8.1. Negotiation Process Overview

Negotiation plays a fundamental role in electronic commerce activities, by allowing the participants to interact and take decisions for mutual benefit. Recently there has been a growing interest in conducting negotiations over Internet, and constructing large scale agent communities based on emergent Web Service architectures.

In multi-agent systems, agents need to interact in order to fulfil their objectives or improve their performance where negotiation is a key form of interaction that enables groups of agents to arrive at a mutual agreement regarding some belief, goal or plan. The negotiation process has different types of interaction mechanisms suit different types of environments and application such as protocols in the style of the contract net, and argumentation. Agents need mechanisms that facilitate information exchange, *coordination* (in which agents arrange their individual activities in a coherent manner), *cooperation* (in which agents work together to achieve a common objective), and so on. We present the following definition of negotiation according to the [Walton and Krabbe, 95]:

“Negotiation is a form of interaction in which a group of agents, with conflicting interests and a desire to cooperate try to come to a mutually acceptable agreement on the division of scarce resources.”

From the point of view of Jennings, negotiation word has been used in a variety of ways, though in general it refers to communication processes that further coordination and cooperation. So, Jennings et al define it:

“Negotiations can be used to resolve conflicts in a wide variety of multi-agent domains” [Jennings et al., 00].

When building an autonomous agent which is capable of flexible and sophisticated negotiation, several protocols for strategic negotiation are considered. Evaluation of the results of multi-agent protocols is not an easy task. Nevertheless, there are certain parameters that can be used to evaluate different protocols, [Kraus, 01]:

- **Negotiation Time:** Negotiations which end without delay are preferred over negotiations which are time-consuming. It is assumed that a delay in reaching an agreement causes an increase in the cost of communication and computation time spent on the negotiation.
- **Efficiency:** An efficient outcome of the negotiations is preferred. In other words, an outcome that increases the number of agents which will be satisfied by the negotiation results and the agents' satisfaction levels from the negotiation results.
- **Simplicity:** Negotiation processes that are simple and efficient are better than complex processes. Being a “simple strategy” means that it is feasible to build it into an automated agent. A “simple strategy” also presumes that an agent will be able to compute the strategy in a reasonable amount of time.
- **Stability:** A set of negotiation strategies are stable if, it is beneficial to an agent to follow its strategy too. Negotiation protocols which have stable strategies are more useful in multi-agent environments than protocols which are unstable.
- **Money transfer:** Money transfer may be used to resolve conflicts. For example, a server may “sell” a data item to another server when relocating this item. This can be done by providing the agents with a monetary system and with a mechanism for secure payments. Since maintaining such a monetary system requires resources and efforts, negotiation protocols that do not require money transfers are preferred.

2.8.2. Ontology Negotiation in Multi-Agent Systems

In systems composed of multiple autonomous agents, negotiation is a key form of interaction that enables groups of agents to arrive at a mutual agreement regarding some belief, goal or plan, as we mentioned previously.

FIPA has analyzed the interoperability problem in heterogeneous MAS and has proposed an Ontology Agent (OA) for MAS platforms to assist the community of agents in the mapping of ontologies. Furthermore, The FIPA Ontology Service Specification classifies this domain dependent task as very complex and possibly not always achievable, because ontology services are important to allow for negotiation to take place. However, no mechanism is provided to match terms between different ontologies. Only in recent years has the problem of handling different ontologies negotiation in MAS been addressed. In this Section we provide a brief summary of the main contributions in this domain, which are:

[Bailin and Truszkowski, 02], this approach describe an ontology negotiation protocol (ONP) between agents supporting intelligent information management, and a framework for implementing agents that use this protocol. This ONP allows Web-based information agents to resolve mismatches in real time without human intervention. The system employs the WordNet lexical database as a source of extending each ontology's concept repertoire. However, the heart of the process lies in the exchanges between agents when WordNet by itself does not allow the agents to interpret each other's concepts. The exchanges of these data are structured by the rules of the ONP where it allows each agent to ask for clarification of previous messages and for confirmation or correction of attempted interpretations. Additionally, they define an API tool to provide functionalities to support ontology evolution. The terms exchanged between the standard agents consist either of queries or answers to queries. Both contents can be viewed as keywords describing the document that is either desired (query) or found (answer).

[Wiesman and Roos, 04], in this approach the authors proposed a domain independent method for handling interoperability problems by learning a mapping between ontologies. The learning method is based on exchanging instances of concepts that are defined in the ontologies. The method is based on identifying pairs of instances of concepts denoting the same entity in the world using information retrieval techniques, followed by proposing and evaluating mappings between the ontologies using the pairs of instances. The important benefits of the method are that no domain knowledge is required, and the structures of ontologies between which a mapping must be established, play no role.

[Malucelli et al., 06], this approach is focused on the resolution of negotiation conflicts in a B2B domain. The authors define a set of services for tackling with the interoperability problems which arise during inter-agent communication. The most important service provided is the resolution of ontological conflicts. They propose a methodology to assess the similarities between the concepts represented in the different ontologies without the need to build a priori a shared ontology. This approach uses a mediator agent. This agent is responsible for the resolution of all

negotiation conflicts that occur within the MAS communication. They used the Protégé ontology editor and OWL ontology language to create and store the ontologies. Each agent has its own private ontology and is ignorant of the ontologies of the other agents. The mapping between ontologies is established by comparing, for each pair of concepts, the attributes (grouped by data type), the relation has-part and the descriptions of the concepts. The comparison includes both syntactic and semantic measurements

[Van Diggelen et al., 07], in this approach the authors proposed the ontology negotiation in heterogeneous multi-agent system which called ANEMONE (An Effective Minimal Ontology Negotiation Environment) system. The first introduced of this system in [van Diggelen et al, 06]. The purposed ANEMONE established an effective communication by using a minimal common ontology. The communication protocol detects when communication is ineffective and applies techniques for ontology exchange to build a common ontology of minimal size. The communication mechanism in ANEMONE system consists of three layers. The upper layer of the protocol is the Normal Communication Protocol (NCP) which deals with the kind of social interaction that agents normally exhibit when no ontology problems exist in the system. If the agents fail to understand each other, the agents switch to the middle layer in the protocol which is the Concept Definition Protocol (CDP). In this layer, the agents explain the meaning of a concept to each other by exchanging concept definitions. If the communication difficulties are so severe that the agents do not even understand each other's concept definitions, the agents switch to the lowest layer in the protocol, which is the Concept Explication Protocol (CEP). Finally, the authors have applied ANEMONE to the domain of news articles.

2.9. Summary

As we presented in this Chapter, although there is not a unique definition for ontology, there is a large accord about the need to use ontologies in different domains. Several authors' identified different types of ontologies, in this thesis; we have listed and discussed the most representative. The design principles, which should be followed when building ontologies, were also discussed. Finally, several development tools and languages for ontology creation have been summarised.

We have founded too that MAS are typically an open systems, which means that the ontologies rather than being defined once and for all, are expected to expand as new needs arise. Thus, the problem of using different ontologies in MAS has only recently been addressed. We reported the most representative work in this domain as well as the most ontology mismatch problem.

Our goal is to design and optimize an information system to support urban mobility which provides users with the services requested in respecting the two criteria: the information cost and the required response time. It is a multi-objective problem has to be solved within a system widely distributed and dynamic which connects with information providers of multimodal transport networks. Therefore, our research has been conducted in order to provide services modelling method to facilitate the information retrieval from different system providers, and the negotiation process between agents in a transport system when there is the perturbation case. In order to try to solve those problems and to give users all needed information even if some information providers are no longer available. We proposed an ontology solution with some matching techniques and mechanisms which we aim to improve the Quality of Services (QoS) of the response time with the best cost. We explain our approach in details in Chapters 3 and 4.

Chapter 3. Our Proposal: Modelling of Services to Aide Urban Mobility by Using Flexible Ontology in The Transport Domain

3.1. Introduction

The real world has become too complex to implement entirely within an information system such as Intelligent Transportation System (ITS). A travel planning technique is an essential issue related to decision support tool in ITS. Thus, the traveller from the transportation system seeks assistance when he looks the information's about moving from one place to another. In majority of times, this moving deals with different modes of transport (plain, train, metro, bus, etc...) to achieve user requirements. Besides to these transport services, it is useful to support the user by the services related to transport (lodgment reservations, cultural events, weather, tourist information, etc). All these information come from different systems, so it should be planed and scheduled to present to the traveler and aid him to take the best choice. In this manner, user travel planning services use semantic technologies and ontologies used to satisfy user demands and predict their requirements.

In this rest of this chapter, we give some examples about the ontology application in the transport domain in section 3.2. Then, we address the design and the development phases of our travel ontology (purpose, conceptualization, formalization, and validation.) to aid the modeling system tasks for planning user's travels in section 3.3. We illustrate the services modelling by using our travel ontology in section 3.4. Finally, the conclusion of the third chapter will be in the section 3.5.

3.2. Some Examples of Ontology Applications in Intelligent Transport Systems (ITS)

Ontologies have been used in many domains and studies (ex, medical domain, tourism domain, etc ...), thanks to their capacity to sharing of knowledge bases, knowledge organization, and interoperability among different systems.

In tourism domain, there already exist different taxonomies and catalogues which are designed and used internally by tourism agents to help them to manage heterogeneous tourism data. Efforts are made to generate global standards to

facilitate inter and intra tourism data exchange. In this section we present several publicly available formal tourism ontologies which show the current status of the efforts for problem specific tourism ontologies.

Harmonise Ontology¹⁶. The Harmonise Ontology was created within the EU Project Harmonise. It is specialized to address interoperability problems in the area of tourism (e-tourism) focusing on data exchange. The aim of Harmonise ontology is to allow tourism information systems to cooperate without the need to modify their software or their data organization. In Harmonise project each organization sees the world using its own concepts and data schemas. Harmonise is based on mapping different tourism ontologies by using a mediating ontology. This central Harmonise ontology is represented in RDF and contains concepts and properties describing tourism concepts, mainly dealing with accommodation and events.

Mondeca Tourism Ontology¹⁷ includes important concepts of the tourism domain which are defined in the WTO (World Tourism Organization) thesaurus. The WTO Thesaurus includes information and definitions of the topic tourism and leisure activities. Mondeca ontology helps enterprises to integrate and interlink heterogeneous information by mapping it to explicit knowledge references. It also improves the way information is retrieved, analyzed, and reused by producing consistent, precise, and relevant metadata as well as supplying the relevant context. The dimensions which are defined within the Mondeca ontology are tourism object profiling, tourism and cultural objects, tourism packages and tourism multimedia content. The used ontology language is OWL and the ontology itself contains about 1000 concepts.

OnTour Ontology¹⁸ is a lightweight system based on the RDF data which created especially for the tourism domain and was developed by DERI (Digital Enterprise Research Institute). The OnTour ontology includes the tourism concepts (location, accommodation...), and the concepts that describe leisure activities and geographic data. They used ontology language is OWL-DL to build their ontology.

3.3. Modelling System Tasks by Using Travel Ontology for Planning Users Travels

Our system aims to support the transport passengers in planning their travels.

¹⁶<http://www.harmon-ten.org>

¹⁷<http://www.world-tourism.org>

¹⁸<http://ontour.deri.org/>

In this context, the user defines (the departure city, the arrival city, the date and etc...) of his travel. With this information, our system looks for all trips possibilities in the different servers by using his travel ontology. Indeed, our system takes into account possible disturbance (crash, bottlenecks, etc.) ,through the Extended and distributed Transport Multimodal Network (ETMN) which contains several heterogeneous data sources including the different proposed transport services, in order to satisfy user requests in all cases. So, we propose using the transport ontology that will improve the communication process between the mobile agent systems.

To define our travel ontology, we represent the necessary knowledge to address this goal. In fact, there are various methodologies to develop an ontology (section 2.2.5). All of them consider basically the following steps: definition of the ontology purpose, conceptualization, formalization, and validation. We will discuss the different steps in the rest of this chapter.

3.3.1. Ontology Purpose

In order to achieve an efficient interoperability between heterogeneous information systems, we propose the travel ontology solution, because the ontologies play an important role in resolving semantic heterogeneity by providing a shared comprehension of a given domain of interest. In addition, this solution facilitates, automates the communications and the information retrieval between the different transport information providers (servers) in our EMTN. This purpose helps too in modelling the different travel services which help the passenger to decide (which travel?, what mode of transport?, and what are the travel related services (ex: lodgement, tourism services , etc...)?) is preferred to reserve his travel. Finally, ontologies offer many benefits to multi-agent systems (MAS) like: interoperability, reusability, support for MAS development activities (such as system analysis and agent knowledge modelling) and support for MAS operation (such as agent communication, negotiation and reasoning).

The Figure (3.1) illustrates the general view of our system which supports the user travel planning. Indeed, the users define their travels (departure city, arrival city, departure date, etc...). With this information, our system supported by the travel ontology generates the queries and looks for all the travel possibilities in different servers. Our system collected also all the relevant geographic information and other travels services. All this information is used to create several instances in our travel ontology. Thus, our system uses those instances to optimize user queries and propose the best possible travels with best choices of travel services (cost, price) to the users for their travels.

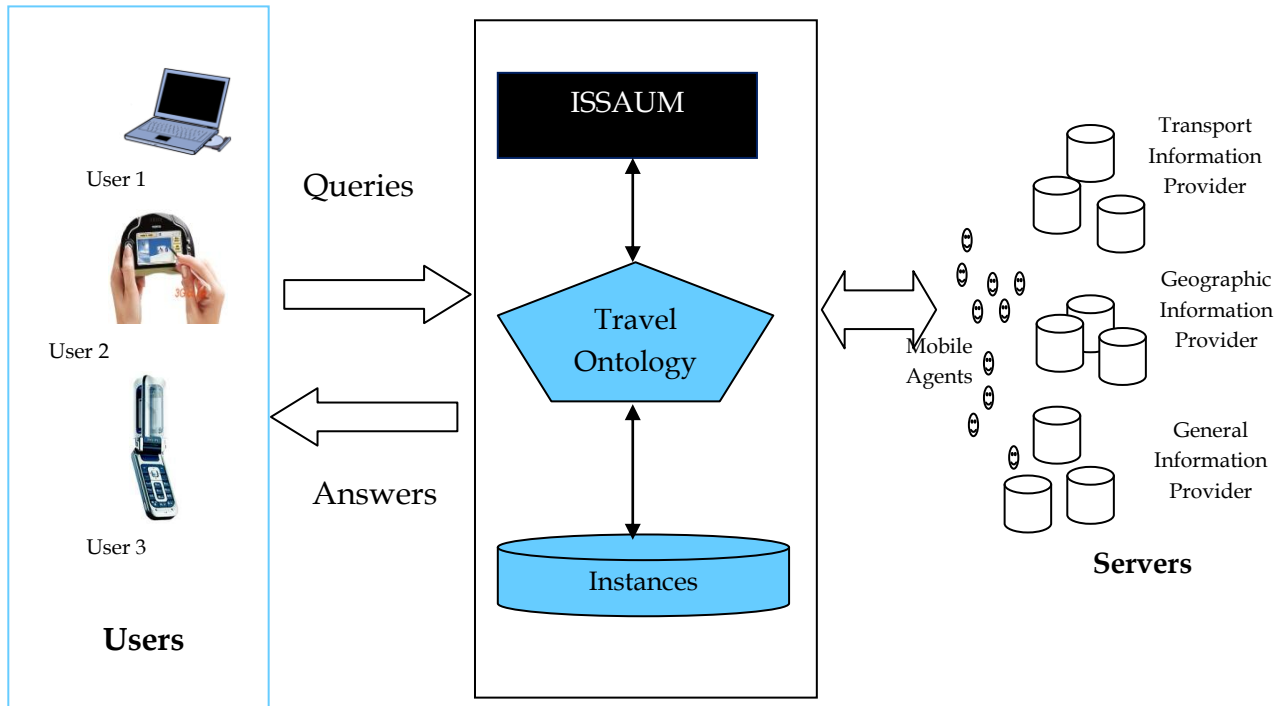


Figure 3.1. General View of System Using the Ontology to Support Traveller's Planning

3.3.2. Conceptualization

As we motioned previously, the second step for modelling our travel ontology is the *conceptualization* step which is the longest step and it defines by: 1) determine the domain and the scope of the ontology, which sketch by the list of competency questions, that a knowledge base and the ontology should be able to answer?. 2) Then, enumerate the important terms and their properities which we would like to talk about or to explain them to a user. We can do this step by putting all the terms description in a glossary for all concepts and attributes. 3) Finally, defining of all ontology's concepts, attributes, relations and constraints.

3.3.2.1. Competency Questions

We will start the development of our travel ontology by defining, in this part, the competency questions which the ontology must be able to answer. We can then propose the following set of competency questions for the travel ontology with respect to the agent behaviours.

- i. What are the agent actions possible to begin the communications process tour?

- ii. What are the events generated when we performed the communication between agents when the later search the travels and their services?
- iii. How can an agent organize a travel (flight, train trip, ship trip)?
- iv. What is the means of transport, which he has to use?
- v. Is there a traffic perturbation in the EMTN?
- vi. What is the cheapest trip selected?
- vii. What kinds of lodgements' reservations are available to the passenger?

3.3.2.2. *Description Glossary*

According to the needs of users of our system, we can define the glossary of our travel ontology. This glossary will describe all the terms (the most relevant concepts).Table 3.1.

Concepts	Definition
ACTOR	Actor is used for representing the software agent in the system.
ACTION	Typical actions which can be performed by software agents are defined to represent the types of inter-agent communication
EVENT	The events which can be take place in the system.
DOnto	Stands for all the domains in the agent environment where DOnto gives the flexibility to the negotiation process which can capture the valid knowledge for different domains (e.g. Transport domain, Geographic domain, etc).
TRAVEL	A trip from an origin place to a destination place using a specific transport mode.
CITY	A geographic place which may be specified as the origin or destination of a trip.
STOP-CITY	A point where passengers can change their travel city, or from one transport

	vehicle to another to continue their trip.
FLIGHT	The trip which can be done by plain.
TRAIN-TRIP	The trip which can be done by train.
SHIP-TRIP	The trip which can be done by ship.
MEANS-OF-TRANSPORT	A set of transport mode which can be contain the public transport too. The travel characterized according to the means of transport (bus, tram, metro, train, plain, ship).
LODGMET	Is used to reserve a lodgement like (hotel or bed&breakfast).
PERTURBATION	The perturbation problems which can be to take place in the transportation networks.

Table 3.1. Description Glossary

3.3.2.3. *Define : the Concepts, the Attributes, the Relations and the Constraint of Our Travel Ontology*

To answer the *first and second competency questions* concern the agent actions and their events (*i. What are the agent actions possible to begin the communications process tour? , and, ii. What are the events generated when we performed the communication between agents when the later search the travels and their services?*).

We think that the best solution is to design an agent architecture based knowledge model system in applying the agent actions like (Propose, Agree, Refuse, etc...) which use the Agent Ontology, [Saad et al., 08]. The Agent architecture idea comes from the pellucid project ¹⁹.We extended and used it in our MTIS project.

Our architecture is modelled as a workflow of basic agent behaviour (our model is based on events) (Figure 3.2). The idea is taken partially from the JADE ontology model (predicates, concepts and agentAction) (see section 2.3). Where this graph can be understand as a formal representation of ontology described by graph. Black boxes represent ontology classes, black arrows stand for relations between classes, mostly inheritance relations, expressed by words "Is-a". Property relations are represented by blue arrows with name of property and cardinality that is mostly

¹⁹ <http://www.sadiel.es/Europa/pellucid/2002>

multiple “*”. Red boxes denote ontology individuals and red arrows relations of an individual to ontology class with associated letters “io”. Such graphs can be generated by using Ontoviz plug-in for Protégé (section 2.2.5).

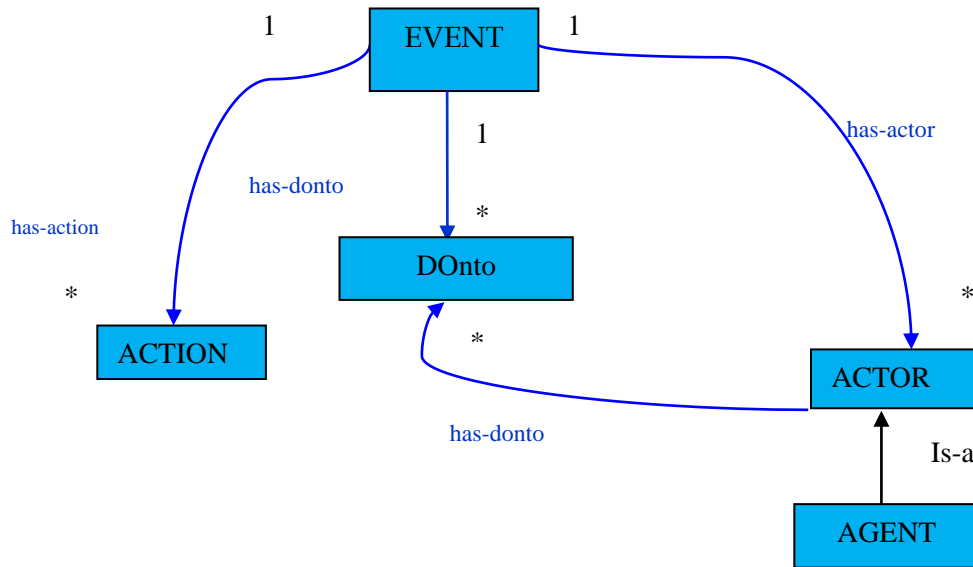


Figure 3.2. Agent Architecture

Our model is based on: ACTIONS, ACTORS, DOnTo and EVENTS. In all cases EVENTS are generated based on performed communication, and on received event actor's model. In our proposal, we aim that our DOnTo gives the flexibility to the negotiation ontologies which can capture the valid knowledge for different domains (e.g. Transport domain, Geographic domain, etc). Figur 3.3.

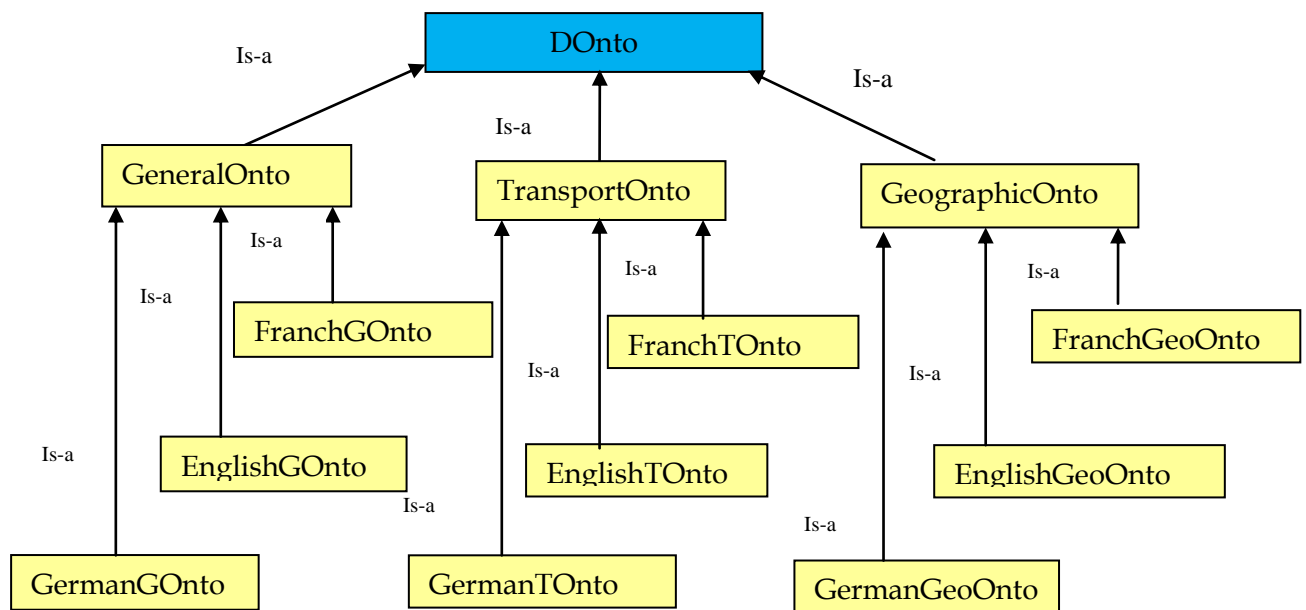


Figure 3.3. Domain Ontology

Typical actions which can be performed by software agents are defined. They represent types of inter-agent communication such as (Agree (total, Partial), Propos, Conform, Cancel, etc...) message. When communication between agents is performed, events of such kind are generated (Figure 3.4).

$$\{aAgree, aPropos, aConform, aCancel\} \in ACTION$$

When actions such as creating, translating, merging or alignment of DOnTo are performed in the system, events containing of those kinds of action are stored and evaluated in the system

$$\{ACreate, ATranslate, ARetriev, AMerging, AAligneme\} \in ACTION$$

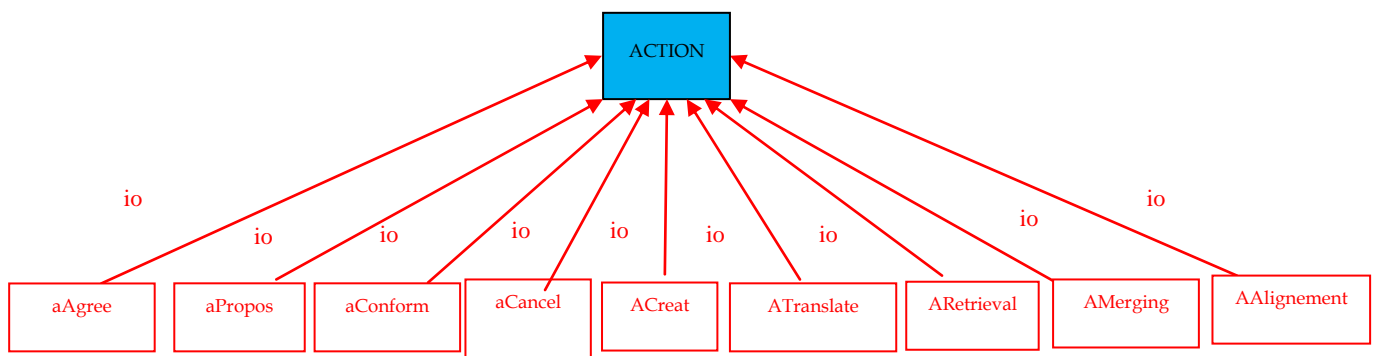
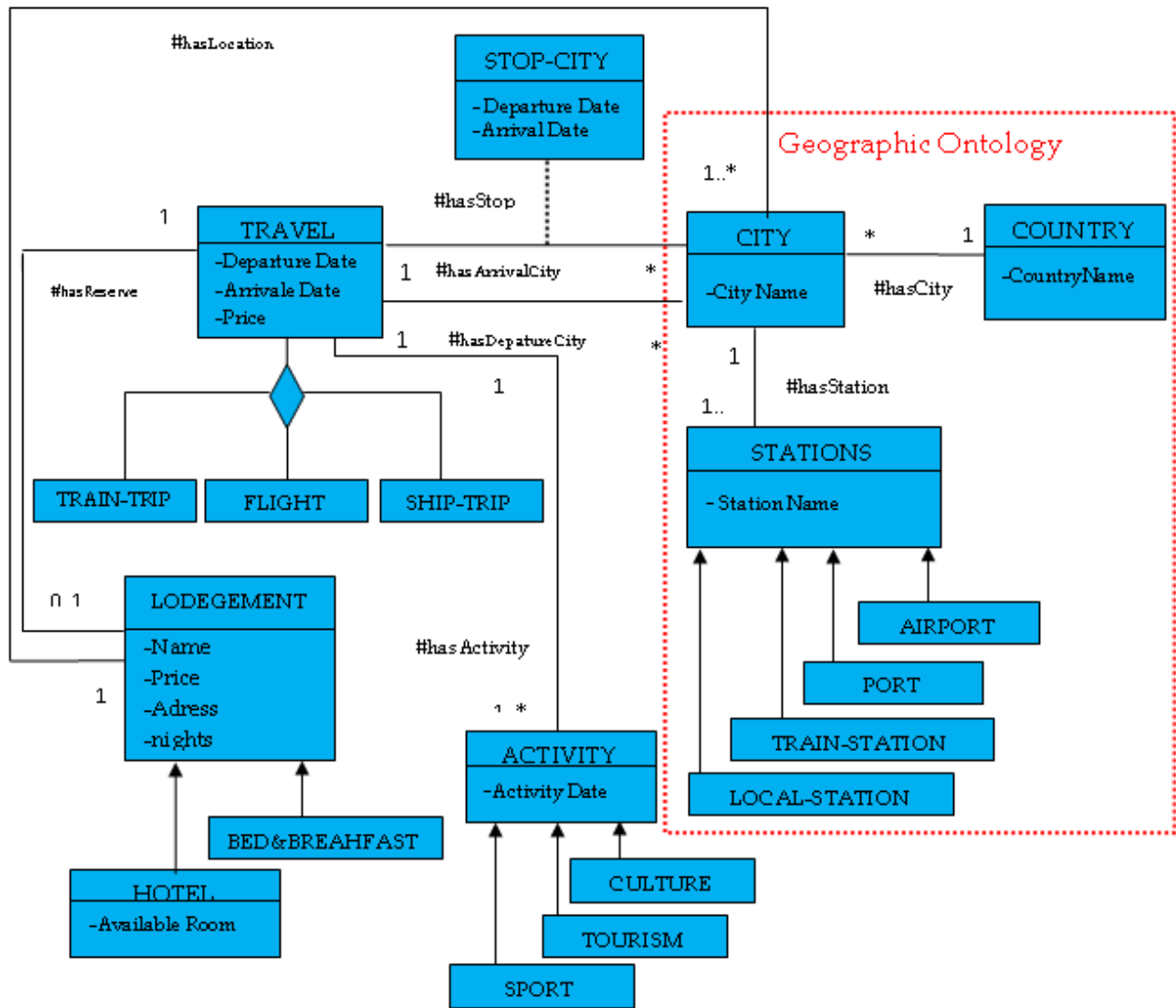


Figure 3.4. Action Model

To answer the (third, sixth and seventh) competency questions (*iii*. How can an agent organize a travel (flight, train trip, ship trip)?, *vi*. What is the cheapest trip selected?, and *vii*. What kinds of lodgements' reservations are available to the passengers?).

We found that we can be defining two ontology to response at this question: the *first ontology* includes all terms based on the travel information offered in tourist agencies and being important for travellers, such as (departure date, arrival date, the different type of the trips and the travel services (ex: lodgement reservations)). Whereas, the *second ontology* will be the geography ontology which defines the concepts and relations from the real world: geographical terms, locations, languages, and all other concepts that are in a way related to tourism (ex: city, country, stations and etc concepts). Each *travel* is a trip (flight, train trip, etc...) from an origin city to a destination one with or without a stop. Then, a *travel* is associated to a *stop-city* that defines the depart time and arrival time based on that should consider in a planning for a travel. Figure 3.5 shows a diagram, using UML notation, of this part of the ontology related to the competency questions (*iii*, *vi*, *vii*).



Figur 3.5. Travel & Geographic Ontology

To answer the *(fourth and fifth) competency questions* (iv. What is the means of transport, which he has to use?, and, v. Is there a traffic perturbation in the EMTN?). The different means of transport which we used in our transport ontology are: plain, train, ship and public transport (metro, tram, train and bus). A *travel* can have different means of transport to realize different types of trips. In the fifth competency questions, the transport ontology can answer the possible disturbance through the EMTN (crash, bottlenecks, etc.) in order to satisfy user requests in all cases by adding the perturbation concept which contains tow attributes (Actual Time and Scheduled Time). Figure 3.6 shows the link between the different means of transport and the problem of perturbation.

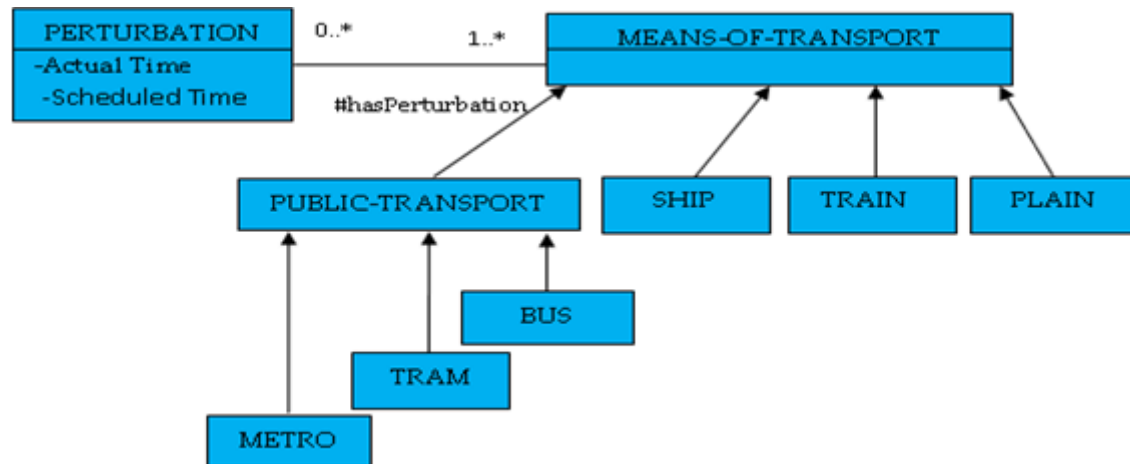


Figure 3.6. Transport Ontology

Finally, Figure 3.7 shows all the concepts, attributes and the relations between the concepts which form our Travel Ontology. For reasons of simplicity the concept properties were not shown.

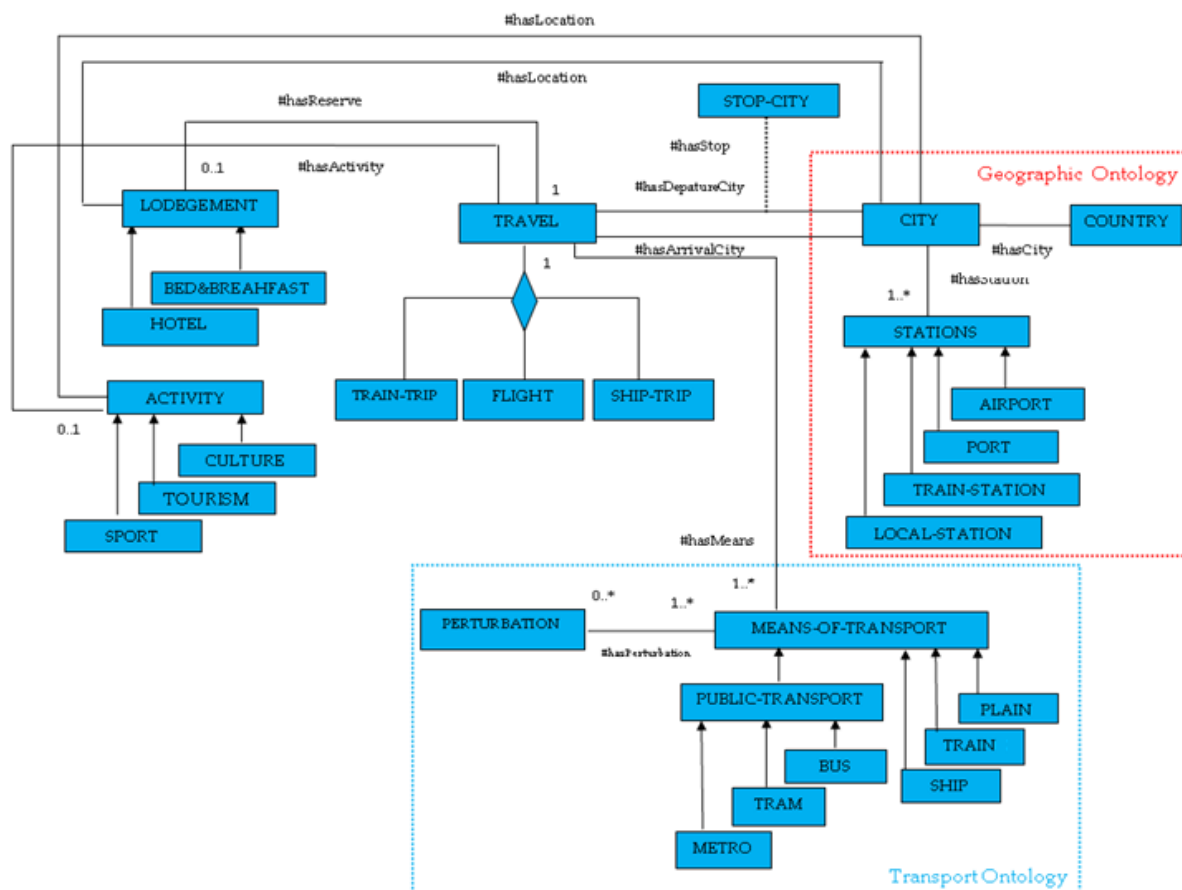


Figure 3.7. Travel Ontology

3.3.3. Formalization

After have studying the conceptualization step, now, we are continuing the modelling of our travel ontology where we arrive to the formalization step. In fact, we describe our knowledge model by using the Description Logic (DL) [Horrocks, 04]. Where, C is a concept (class); P is a role (property); x_i is an individual/nominal. Table 3.2 and Table 3.3 show conversion tables between OWL-DL definitions and DL. Here, we use such formalism to describe our ontology models (agent architecture and travel ontology model).

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male	$C_1(x) \wedge \dots \wedge C_n(x)$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer	$C_1(x) \vee \dots \vee C_n(x)$
complementOf	$\neg C$	\neg Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} \sqcup {mary}	$x = x_1 \vee \dots \vee x = x_n$
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor	$\forall y.P(x, y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer	$\exists y.P(x, y) \wedge C(y)$
maxCardinality	$\leq_n P$	≤ 1 hasChild	$\exists^{\leq n} y.P(x, y)$
minCardinality	$\geq_n P$	≥ 2 hasChild	$\exists^{\geq n} y.P(x, y)$

Table 3.2. OWL Constructor

OWL Syntax	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
transitiveProperty	$P^+ \sqsubseteq P$	ancestor ⁺ \sqsubseteq ancestor

Table 3.3. OWL Axioms

3.3.3.1. Agent Architecture

Firstly, the *EVENT* class represents events in the system. *EVENT* individual $\{event\}$ is $\{action\}$ taken by $\{actor\}$ on particular $\{donto\}$. Properties of *EVENT* class are: *hasDonto*, *hasAction*, and *hasActor*.

$$\begin{aligned}
 &D_{Onto} \sqcup ACTOR \sqcup ACTION \sqsubseteq EVENT \\
 &\{event\} \in EVENT \\
 &EVENT \equiv \exists has-action.ACTION(\{action\}) \sqcap \\
 &\quad \exists has-actor.ACTOR(\{actor\}) \sqcap \\
 &\quad \exists hasDonto.D_{Onto}(\{donto\})
 \end{aligned}$$

The *DOnto* class stands for all the domains in the agent environment (figure 3.3).

$$\begin{aligned} DOnto &\sqsubseteq EVENT \\ \{donto\} &\in DOnto \end{aligned}$$

ACTOR class denotes actors in the environment. Actor individuals can take an action {action} which are individuals of ACTION class.

$$\begin{aligned} \{actor\} &\in ACTOR \\ \{action\} &\in ACTION \end{aligned}$$

Special type of ACTOR is *Agent*. AGENT is used for software agent representation in the system.

$$\begin{aligned} AGENT &\sqsubseteq ACTOR \\ \{agent\} &\in AGENT \end{aligned}$$

3.3.3.2. Travel Ontology

Concerning the Travel Ontology (figure3.7), we have different ontology concepts. We will begin by the class TRAVEL which represents all the trips in the system. TRAVEL individual {travel} is trip {flight, train-trip, ship-trip} from {departure-city} to {arrival-city}. Properties of TRAVEL class are: *has-arrival-city*.TRAVEL, *has-depature-city*.TRAVEL, and *Is-a*.TRAVEL. The concepts which denote to the trip: FLIGHT; TRAIN-TRIP and SHIP-TRIP are subclasses of TRAVEL class.

$$\begin{aligned} FLIGHT \sqcup TRAIN-TRIP \sqcup SHIP-TRIP &\sqsubseteq TRAVEL \\ \{travel\} &\in TRAVEL \end{aligned}$$

$$\begin{aligned} TRAVEL &\equiv TRAVEL \sqcap (\exists \text{ is-a.FLIGHT} \sqcup \exists \text{ is-a.TRAIN-TRIP} \sqcup \exists \text{ is-a.SHIP-TRIP}) \\ &\sqcap \exists \text{ hasDepatureCity.CITY}\{city\} \\ &\sqcap \exists \text{ hasArrivaleCity.CITY}\{city\} \end{aligned}$$

CITY is a class from geographic ontology which has an attribute CityName. An individual de CITY is {city} and it is a property *hasCity* with the class COUNTRY. Every CITY has at least one station.

$$\begin{aligned} \{city\} &\in CITY \\ COUNTRY &\sqsubseteq \exists \text{ hasCity.CITY}\{city\} \end{aligned}$$

$$CITY \equiv CITY \sqcap \geq 1 \text{ hasStation.STATION}$$

STATION is class which composed of all the possible stations in the city.

$$LOCAL-SATATION \sqcup TRAIN-STATION \sqcup PORT \sqcup AIRPORT \sqsubseteq STATION$$

$$\{station\} \in STATION$$

COUNTRY class represents the countries and it has the individual {country}.
Example:

$$\{France\} \sqcup \{German\} \sqcup \dots \sqcup \{England\} \in Country$$

Each travel is served by at least one means of the transport. The class MEANS-OF-TRANSPORT denotes the transport multimodality which the passenger can be use. In our transport ontology, the class MEANS-OF-TRANSPORT is composed of plane, train, ship and public transport means. The later, in turn, consists of the bus, metro, and Tram way.

$$\begin{aligned} TRAVEL &\equiv TRAVEL \geq 1 \text{ hasMeans. MEANS-OF-TRANSPORT } (\{mean-of-transport\}) \\ PLAN \sqcup TRAIN \sqcup SHIP \sqcup PUBLIC-TRANSPORT &\sqsubseteq MEANS-OF-TRANSPORT \\ \{plan\} &\in PLAN \\ \{train\} &\in TRAIN \\ \{ship\} &\in SHIP \\ \{metro\} \sqcup \{tram\} \sqcup \{bus\} &\in PUBLIC-TRANSPORT \end{aligned}$$

Then, every model of transport is associated only by the concept MEANS-OF-TRANSPORT.

If the perturbation occurred in the transportation networks, the related time information update in the scheduled time for each means of transport. The class PERTURBATION with his time attributes {ActualeTime , SchudeledTime} represents the disturbance in the ETMN.

$$\begin{aligned} \exists \text{hasPertubation. PERTURBATION} &\sqsubseteq MEANS-OF-TRANSPORT \\ \{perturbation\} &\in PERTURBATION \end{aligned}$$

Some of travel has a perturbation in a mean of transport. The axiom which presents this travel perturbation is:

$$TRAVEL \sqsubseteq \exists \text{hasPertubation. MEANS-OF-TRANSPORT} (\{perturbation\})$$

Some travels offer the service to make a reservation to a lodgement for some numbers of nights. The class LODEGEMENT consists of tow type of lodgement (HOTEL and BED&BREAKFAST). To represent this axiom, as in follow:

$$\begin{aligned} HOTEL \sqcup BED\&BREAKFAST &\sqsubseteq LODEGEMENT \\ TRAVEL &\sqsubseteq \exists \text{hasReserved. LODEGEMENT } (\{lodgment\}) \\ \{lodgment\} &\in LODEGEMENT \end{aligned}$$

Some travels have a stop in a certain city in a determined date. The STOP-CITY class represents as an associated class which related to tow concepts (TRAVEL and CITY) and contains tow attributes (departureDate and arrivalDate). The STOP-CITY properties are (hasDepatureDate, hasArrivaleDate).

$$\begin{aligned} TRAVEL \sqsubseteq & \exists hasStop. STOP-CITY(\{city\}) \sqcap \\ & (\exists hasDepatureDate. TRAVEL(\{departureDate\}) \sqcup \\ & \exists hasArrivaleDate. TRAVEL(\{arrivalDate\})) \\ & \{city\} \in CITY \\ & \{departureDate\} \in DEPARTURE-DATE \\ & \{arrivalDate\} \in ARRIVALE-DATE \end{aligned}$$

Finally, there are another constraints (axioms) for the concepts and relations that we can be used, such as:

Each FLIGHT is a TRAVEL having at least one PLAIN , and the PLAIN is a MEANS-OF-TRANSPORT

$$\begin{aligned} FLIGHT \sqsubseteq & TRAVEL \sqcap \\ & \exists is-a. TRAVEL(\{flight\}) \sqcap \\ & \geq 1 hasMeans. MEANS-OF-TRANSPORT (\{plain\}) \end{aligned}$$

TRAIN-TRIP is a TRAVEL having a TRAIN, and the TRAIN is a MEANS-OF-TRANSPORT

$$\begin{aligned} TRAIN-TRIP \sqsubseteq & TRAVEL \sqcap \\ & \exists is-a. TRAVEL(\{train-trip\}) \sqcap \\ & \geq 1 hasMeans. MEANS-OF-TRANSPORT (\{train\}) \end{aligned}$$

Each LODGEMENT has an address which is related with the geography ontology by the property has-location. LODGEMENT

$$\begin{aligned} LODGEMENT \sqsubseteq & LODGEMENT \sqcap \\ & \exists hasLocation. CITY(\{city\}) \end{aligned}$$

Finally, we will discuss the validation step of our ontology in the fifth chapter.

3.4. Services Modelling by Using Travel Ontology and The Dynamic Data Archiving

In our work, to market its data, an information provider must solicit the system in order to register or update the services that it offers. A service is characterized by a cost, a response time and a data size. A service is also characterized by a time relevance that allows saving information locally for a certain time to reduce the

transmission of data if that is possible. We found therefore our system requires the existence of two local databases, as we see in the follow:

3.4.1. The Data Base Registration (DBR)

Each provider of the services, wanting to offer its services through our system, must register all its services in this database. For this purpose, we use the reference as the index for the services. In addition, a supplier must register the label of each service proposed (the suitable terms in his ontology), the estimated response time, cost and size of data corresponding. It must also mention the address of his ontology. The same service may be proposed by several suppliers with different (labels, costs, response times and sizes). Thus, those providers use different terms with the same service meaning.

Our system builds on the hybrid ontology approach (see section 2.5). It registers in his DBR the address of local ontology for each information source, and it uses his global travel ontology as a reference for the local ontologies. Thus, the advantage of this approach is: the system can simply integrate the services, and, the addition and the removal of sources can be easily supported.

3.4.2. The Data Base Archiving (DBA)

This database plays the role of "buffer zone" contain static data to a certain degree. In other words, we classified the data on distributed network according to their degree of stability, by a method using the indicators for classification. The aim of this method is to avoid redundant search of the same data not yet changed in the short and long term.

The information, which corresponds to a multimodal transportation services and / or a related service, is represented by the data located in distributed nodes that represent the data of information providers on the EMTN. Therefore, according to the chosen system architecture and the optimization approach adopted (see chapter 4), the mobile agents should visit these nodes to collect the necessary data that satisfy the system users. In previous work [Ben Khaled et al., 05], the information was classified according to four types of indicators: the location indicator, the customization indicator, the time indicator and the indicator of updating and according to the values attributed to these indicators. The information may be in the following categories: Dynamic, event-driven, localized, non-localized, general or personalized. We are only interested in the classification of information according to the time indicators and updated one, in order to classify information in the class static, dynamic or event-driven for dynamic data archiving.

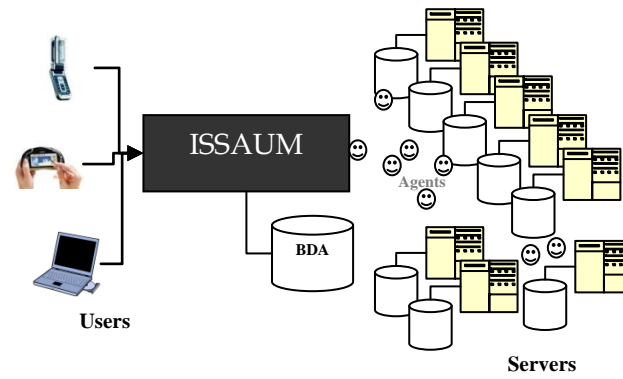


Figure 3.8. The Dynamic Data Archiving

When an information provider offers a service, it must include both the value of its time indicator (IndT) and the value of updating indicator (IndU). The time indicator of information represents its degree of stability, while the updating indicator represents its period of validity. Indeed, when the value of time indicator is low, this mean the information is dynamic and when the validity period ends, the information is no longer valid in this case it must be restored. Thus, the static information is periodically archived in the BDA (Figure 3.8). The data are stored with their time indicator and updating them respectively. The significance of the values assigned to a time indicator is given in Table 3.4.

Indicators		
IndT	IndU	Significance
0	It does not exist	The information is very dynamic
1	$\leq x$ hours	The information can change in less than x hours
2	$> x$ hours	The information can change in more than x hours
3	It does not exist	The information is static

Table 3.4. Time and Updating Indicators for Information Classification

When the information is very dynamic, it is not archived in the ADB and the agents must each time get it. This explains why, for this type of information, there is not the updating indicator. On the other hand, when the information is completely static, there is no need of updating indicator because the information is static. In practice, static data does not remain forever and can be changed in an exceptional manner. In this case, the information provider must inform the concerned agent of the system, which is the responsible of the BDA. This procedure is required to ensure the relevance of the information which is simply the responsibility of its suppliers. Finally, they return to the system, the mobile agents are the responsible for the restoration of outdated information.

3.4.3. Query Service

3.4.3.1. *Ontology Query Languages*

Based on the ontology languages described in section (2.2.4), several query languages and systems have been already developed. Each ontology language provides different expressive power and also computational complexity for reasoning. Ontology query languages were developed to query the information defined by these ontology languages and reasoning systems.

We explain in the follow the most important query systems which have been developed along with the development of these ontology languages:

- **RDF Data Query Language (RDQL) and Jena2:** Jena2 is a framework built by HP Labs. It provides multiple reasoners for RDF, RDFS, and OWL. It also provides a flexible query language, called RDQL [Seaborne, 02]. The development of RDQL is to provide a data-oriented query model. This means that RDQL only queries the information held in the models. RDQL provides no reasoning mechanisms. The reasoning is provided by user selected reasoner bound to the model containing the original ontology information. Provided with a proper reasoner, RDQL can process ontology in various languages including OWL. All variables in the input query are must-bind variables. An RDQL query has the following form (Figure 3.9.):

```
SELECT ?givenName
WHERE (?y <http://www.w3.org/2001/vcard-rdf/3.0#Family> "Smith") ,
      (?y <http://www.w3.org/2001/vcard-rdf/3.0#Given>
       ?givenName)
```

Figuer 3.9. RDQL Query Example

In this query, we want to find a node in the graph ?y, which has the vCard property "Family" with the value "Smith". ?y also has another property, the vCard given name, which we want to put into a variable ?givenName.

- Another system is **OWL Query Language (OWL-QL)** [Fikes et al., 03], is a well designed language for querying over knowledge represented in a repository. OWL-QL is an updated version of DAML Query Language (DQL). It is intended to be a candidate for query-answering dialogues among

answer agents and query agents. Then information receivers and information providers can transfer queries and answers via the Internet.

OWL-QL provides a formal description of the semantic relationships among queries, answers, and knowledge bases used to produce answers. An OWL-QL query contains a collection of OWL sentences in which some URI references are considered to be variables. This collection is called query pattern. For example, a client could ask "Who owns a red car?", with a query having the query pattern shown in follow, Figure 3.10.

Query: ("Who owns a red car?")

Query Pattern: {(owns ?p ?c) (type ?c Car) (has-color ?c Red)}

Must-Bind Variables List: (?p)

May-Bind Variables List: ()

Don't-Bind Variables List: ()

Answer Pattern: {(owns ?p "a red car")}

Answer KB Pattern: ...

Answer: ("Joe owns a red car?")

Answer Pattern Instance: {(owns Joe "a red car")}

Query: ...

Server: ...

Figuer 3.10. OWL-QL Query Example

Unfortunately, the executable package of OWL-QL is not available right now. So, we could not compare it with the other query systems.

- Finally, the **Simple Protocol And RDF Query Language (SPARQL)** [Seaborne et al., 05], is a Server-Client-based RDF query language. It has SQL syntax and is influenced by RDQL. SPARQL supports disjunction in the query and thus can process more complex query than RDQL. SPARQL also provides optional variable binding and result size control mechanisms for real world usage. An example of a SELECT query follows, Figure 3.11:

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?mbox

WHERE { ?x foaf:name ?name .

```
?x foaf:mbox ?mbox . }
```

Figuer 3.11. SPARQL Query Example

The first line defines namespace prefix, the last two lines use the prefix to express a RDF graph to be matched. Identifiers beginning with question mark ? identify variables. In this query, we are looking for resource ?x participating in triples with predicates *foaf:name* and *foaf:mbox* and want the subjects of these triples. Syntactic shortcuts of TURTLE can be used in the matching part.

The SPARQL specification is already implemented in some software packages and it seems that it will become the main RDF querying language for the semantic web. The specification of protocol for a SPARQL web service is available as well - SPARQL then serves as a RDF data access protocol.

We use in our service query the SPARQL querying system, as we explain in next section.

3.4.3.2. The Functionality of Query Service

After we have illustrated the manner of registration and archiving the information provider's data in our system. Now, we explain the query service of the user's requests. User simultaneous queries are submitted to the system only on the reference ontology by using the query service. Thus, our system can query heterogeneous and distributed information sources simultaneously and combine the obtained results in order to get information that may not be available directly. In order to solve the heterogeneity problem between information sources, ontologies are used to describe the semantics of the information sources and to make their contents explicit. Then the local ontologies (travel, transport, geographic, etc ...) are mapped to a global ontology using the mapping service (see section 4.3).

As we mention previously, the global ontology describes the semantics of the whole travel domain. User's queries are submitted to system that analyses the queries; decompose them into sub-queries which are redelivered to the relevant data provider services.

Our reference ontology (travel ontology) describes a specified knowledge for travel domain. It represents the global model for local ontology models and is supposed to cover all the local domains (transport, travel, geographic, tourism, etc...), i.e. each concept, role and attribute in any local ontology has a corresponding concept, role and attribute in the reference ontology. The mapping service contains information

about the mappings between the reference ontology and the local ones. The mapping itself is stored in the data provider service. This service only associates each concept in the referential ontology with a list of local ontologies which are linked to this concept. We will explain the method to estimate the similarity between ontologies components later (see section 4.3).

When a query is submitted to the system, it is analyzed by this query service and decomposed into a set of sub queries. In fact, queries are expressed in SPARQL language. Therefore, a query is composed of a set of triple patterns. Each triple pattern corresponds to a concept or a property in our travel ontology reference. For each local ontology, a sub-query is established by selecting from the global query the triple patterns that are relevant to this local ontology (according to the mapping service).

When an SPARQL query is received, it is translated to an SQL query using the mappings between the database and the local ontology. Then, the agent must verify the IndT for each identified service. If the corresponding information dose not very dynamic ($\text{IndT} \neq 0$); then the agent must check whether this information has already been stored locally in the BDA. If so, the agent must check the value of the indicator IndU. If the information is still valid, then the agent sends data directly to the agent available which is responsible of collects the responses and recomposes them in one coherent response which will be sent to system users. The system sends the mobile agents to collect the information from data provider services, in three cases:

1. If the information is very dynamic ($\text{IndT} = 0$).
2. If the information is not very dynamic ($\text{IndT} \neq 0$), and does not yet exist in the BDA.
3. If the information is not very dynamic ($\text{IndT} \neq 0$), and it exists in the BD. But it is no longer relevant.

In the two latter cases, an update of the data must be take place at the BDA.

For collecting the date, each sub-query is then redelivered to the appropriate data provider service by the mobile agents. In other words, each data provider service will receive only a subset of query triple patterns which are covered by its local ontology. From his part, the data provider service when it received an SPARQL query. It translated it to an SQL query using the mappings between the database and the local ontology. The SQL query is executed in the database and its result is encapsulated as an SPARQL response and returned to our system by mobile agent. The system then collects the responses returned from data provider services and recomposes them in one coherent response which will be sent to system users.

Finally, the query service and the dynamic data archiving model which are provided by our system. It is illustrated by an activity diagram (Figure 3.12).

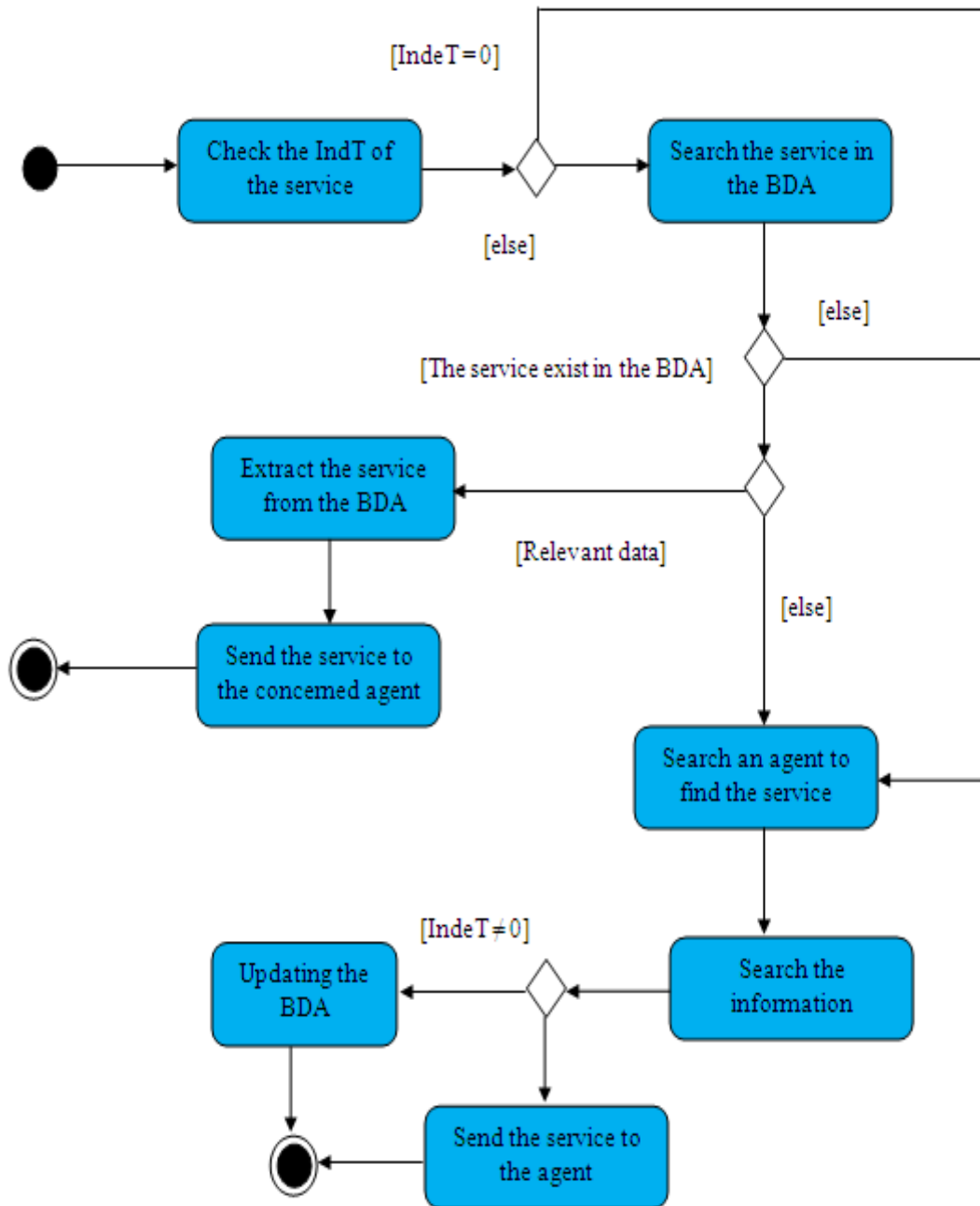


Figure 3.12. Query Service and the Dynamic Data Archiving Model

3.5. Summary

We presented in this chapter transportation ontology defined from the analysis of the main concepts of the public transportation domain. Our aim is to assist the user request to choose the best way to go from one point to another by using our travel ontology.

In this chapter, we tried to detail the different development steps of our travel ontology, from the definition of the concepts till the specification and formalization of axioms related to the transport domain by using the description logic. We describe also how this ontology can be used to support user travel planning by using the query service and the dynamic data archiving methods. We explained the agent collecting way of the instances of the concepts from information providers, in order to provide our system user all the necessary information for a travel planning with best time and cost.

Finally, thanks to great adaptability of our travel ontology. It can be applying to achieve efficient interoperability of information systems. Indeed, ontologies play an important role in resolving semantic heterogeneity, so, we propose a general multi-agent system that uses ontologies for explicit description of the semantics of information sources, and mapping services to facilitate the communication between the different agents in our system architecture which we will illustrate in the next chapter.

Chapter 4. Optimize Solution Based on Agent to Aid Urban Mobility by Using a Flexible Transport Ontology

4.1. Introduction

The problem of optimization of distributed data and the services composition of multi-sources in the transport domain follow the continuing growth of distributed information in the large scale networks. The distributed information systems are directly involved in this optimization problem which requires the selection and the search of services (divers and numerous), because a service can be offered by several information providers, in competition, with costs, formats and different response times. In this context, Multimodal transport network customers need to be oriented during their travels. A travel support tool can be offered by a Multimodal Information System (MIS) which allows them to express their demands and provides them the appropriate responses to improve their travel conditions.

Our goal is to design and implement an Information System of Services to Aid the Urban Mobility (ISSAUM) which can be take, at the same time, a real-time decision support for transport customers and a trip planning tool. In our previous work [Zgaya, 07], we aimed to design, optimize and implement a Multimodal Transport Information System (MTIS) to optimize the management of the data flow of the users' requests, which can be simultaneous and numerous . In (MTIS), Zgaya has developed a negotiation protocol intended for the transport area which permits the agents to negotiate in the case of perturbations. This protocol uses messages to exchange the information where those messages are exchanged between the Scheduler Agents (SAs), representing the initiators of the negotiation, and the Intelligent Collector Agents (ICAs), representing the participants of the negotiation. This protocol has studied before only the cases of the simple messages.

For that, we propose an approach that will improve the protocol of the negotiation of the multi-agents where we present an ontology solution based on the knowledge management system for semantic heterogeneity. Our approach aims to make the agents able to understand each other when using these ontologies and by applying the mapping services to resolved the misunderstanding problem.

Finally, we detailed the reassignment process by using Dynamic Reassigned Tasks (DRT) algorithm which based on the current state of travelling mobile agents in their correspondent routes called Workplans. Our goal is to give users all needed information even if some information providers are no longer available. Thus we improve the Quality of Services (QoS) of the response time with the best cost.

In this chapter, we illustrate firstly our multi-agent system and the optimize solutions (section 4.2). The proposed ontology mapping negotiation model explained in section 4.3, where in this section, we present the organization of our architecture as follow: the first layer contains the *Negotiation Layer (NL)*. The second layer represents the *Semantic Layer (SEL)*, and the third layer is the *Knowledge Management Systems Layer (KMSL)* uses ontology in purpose of automatic classifying and using of the news ontologies and meta-ontologies. The different negotiation protocols explain in the section 4.4. We present the dynamic reassigned tasks (DRT) algorithm with ontology mapping approach in the section 4.5. Finally, the summary of this chapter will be in the section 4.6.

4.2. Multi-Agent System Architecture and The Optimize Solutions

4.2.1. Problem Statement

The main concern of our Information System of Services to Aid the Urban Mobility (ISSAUM) is to satisfy transport customers, respecting delays of responses (due dates) and minimizing their costs, i.e. improve the Quality of Services (QoS). This problem becomes difficult to resolve if user requests are numerous and simultaneous, formulated by a set of customers via different devices (laptop, desktop, mobile, PDA, etc...) at the same time. For example, if there is a user connect to our ISSAUM at the instant $t=11.00$ am. The user looks for a **travel from Lille to Paris today at 14.00 pm** and he Looks for a **hotel of a good (Quality /Cost) in Paris for two nights** and he would like to **make a reservation**.

We therefore propose to handle user's requests, which have received as simultaneous "block", through a stage of decomposition of these simultaneous requests into a set of independent sub-requests called tasks (Figure 4.1), according to our travel ontology which has proposed in the previous section. In our example, we have two different tasks, table 4.1:

Tasks	Descriptions
T1	Travel from Lille to Paris today at 14.00 pm
T2	Looking for a hotel of a good (Quality /Cost) in Paris for 2 nights

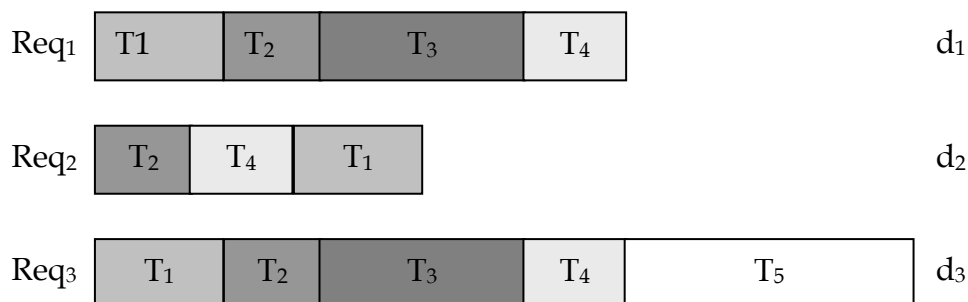
Table 4.1. Table of Tasks

Each task corresponds to a service which can be proposed, differently, by several information providers, in competition, with different cost, response delay, (different size and format) of the data (i.e. knowledge representing formats), and different services labels. A single task can correspond to transport services (sub-route, well-known geographical zone...) or to relate services (lodgement, cultural event, weather forecast, etc...). In our example, according to our travel ontology, we find that those tasks correspond to the following concepts, table 4.2 :

Tasks	Associated Concepts	Services Labels
T1	TRAVEL,CITY,COUNTRY, DepartureDate=today at 14.00	Travel Service
	TRAIN,CITY,COUNTRY	Transport Service
	TRAIN,CITY,COUNTRY,TRAIN-STATION	Geographic Service
T2	LODEGEMENT,HOTEL,CITY, COUNTRY, Nights=2	Travel Service (Lodgement)

Table 4.2. Table of Associated Concepts

The user's requests decomposed as follow:

**Figure 4.1. Requests Decomposition**

The information providers (called nodes), correspond to distributed information servers on the Extended and distributed Transport Multimodal Network (ETMN) which contains several heterogeneous data sources including the different proposed services to transport users. An information provider, which aims to propose some services through the ISSAUM, has firstly to register its information system, by assuming the responsibility for the legal and qualitative aspects of the correspondent

data. After the decomposition process, information providers (distant nodes), which propose services to the correspondent identified tasks, are recognized (Figure 4.2).

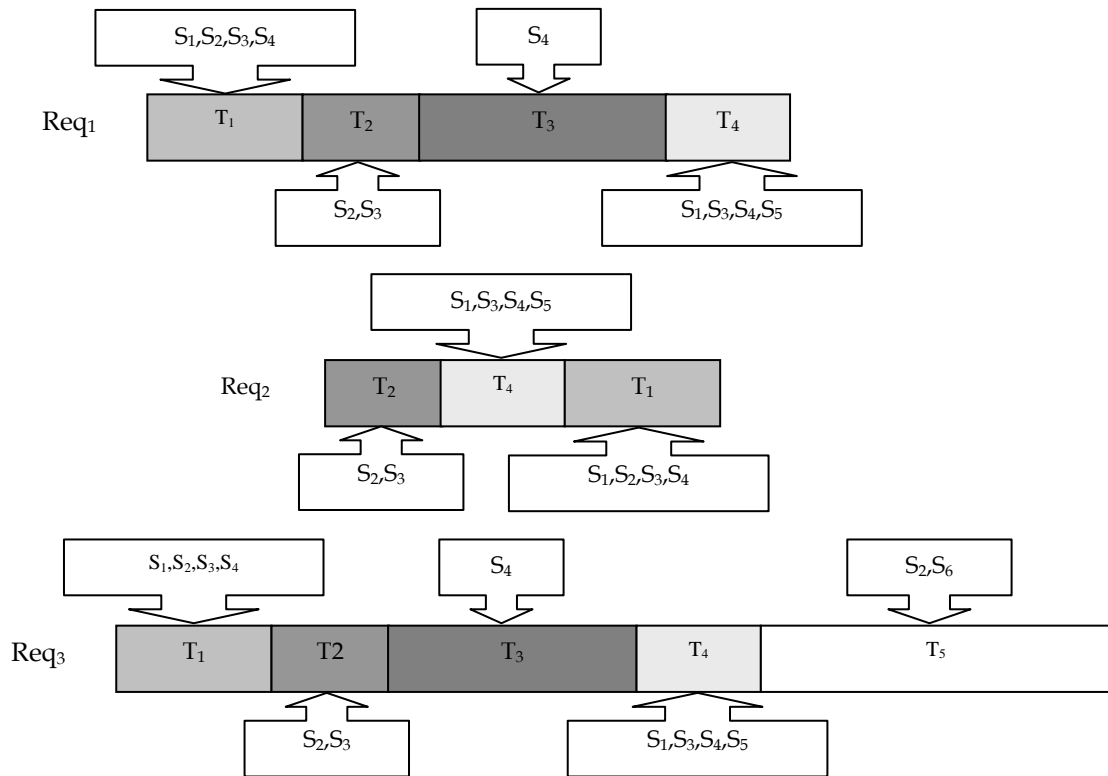


Figure 4.2. Nodes Identification

Finally, nodes must be assigned in order to satisfy all connected users knowing that a user is satisfied if his request was answered rapidly with a reasonable cost (Figure 4.3).

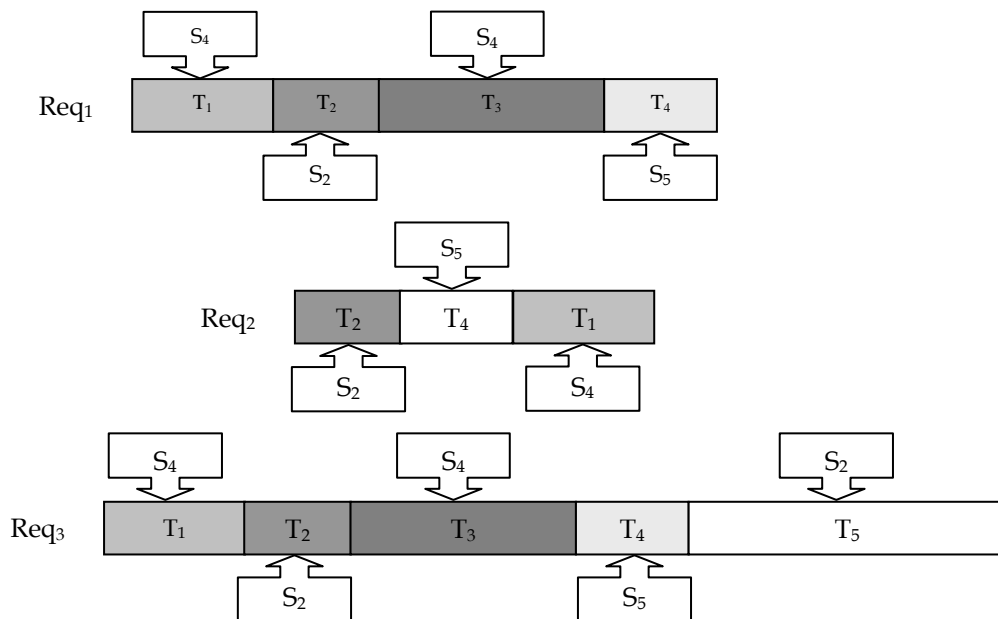


Figure 4.3 Nodes Assignment

We have designed the problem i which described previously by:

- R requests, waiting for responses at the same instant t . The set of these requests is noted R_t ,
- Each request $req_w \in R_t$ is decomposed into a set of independent tasks, noted $I_{t,w}$,
- Each request req_w has a due date d_w initially known (Figure 4.1), an ending date D_w and a total cost C_w ,
- The set of independent I tasks representing all proposed services on the Multimodal Network (MN) is noted I_t ,
- The realization of each task $T_i \in I_t$ requires a resource, or node, selected from a set of J available nodes, noted $S = \{S_1, \dots, S_J\}$,
- The set of independent I' tasks ($I' \subseteq I$) composing R_t is noted I'_t ($I'_t \subseteq I_t$),
- The set of J' nodes ($J' \subseteq J$) selected from S , to perform I'_t is noted S' ($S' \subseteq S$),
- There is a predefined set of processing time; for a given node S_j and a given task T_i , the processing time of T_i using S_j is defined and noted by P_{ij} ,
- There is a predefined set of information cost; for a given node S_j and a given task T_i , the cost of the information to collect from S_j , corresponding to the service referenced by T_i , is defined and noted by Co_{ij} ,
- The size of the data to collect to ensure a service is defined; for a given node S_j and a given task T_i , the data size is defined and noted by Q_{ij} ,
- We have partial flexibility, the realization of each task T_i requires a node selected from a set of nodes which propose the same service performing the task T_i , with different cost, processing time and data size. Therefore, a service is described by:
 - A processing time P_{ij} of the task T_i on the node S_j . It corresponds to the estimated time to perform the task T_i by means of the resources of S_j ,
 - The cost of the service Co_{ij} corresponding to the task T_i on the node S_j ,
 - The data size Q_{ij} corresponding to the size of the information to collect from S_j to response to T_i .

A same task may be performed differently on several nodes, namely with different processing time, different cost and different response formats. These three characteristics $(P_{ij}, Co_{ij}, Q_{ij})$ represent successively the first, second and last term of each element of a service table. Table 4.3 below shows an example of different proposed services. We notice that if a provider does not offer a response to a task (partial flexibility); the correspondent term in the table above is $(0,0,0)$. Otherwise we have $P_{ij} \neq 0$, $Co_{ij} \neq 0$ and $Q_{ij} \neq 0$.

	S_1	S_2	S_3	S_4
T_1	(0,0,0)	(2,5,3)	(4,3,3)	(2,5,3)
T_2	(2,4,5)	(1,5,2)	(4,5,1)	(3,8,3)
T_3	(1,7,3)	(0,0,0)	(2,5,3)	(4,2,2)
T_4	(3,2,1)	(0,0,0)	(0,0,0)	(0,0,0)
T_5	(2,3,1)	(1,1,3)	(4,5,2)	(4,5,3)

Table 4.3. Example of Available Services (Processing Time, Cost and Data Size)

4.2.2. MAS Architecture

In this section we will explain the general architecture of our system and the agent's behaviours.

4.2.2.1. General System

To resolve the problem described in last section, we propose a system based on the coordination of five kinds of software agents [Green et al., 97], [Davics, 91], figure 3.4. The Interface Agents (IAs), the Identifier agents (IdAs), the Scheduler Agents (SAs), the Intelligent Collector agents (ICAs), the Fusion Agents (FAs), and the Translation Agents (TAs).

The IA agent plays the role of the interface between the user and the system where the agent IdA manages the simultaneous requests formulated by the users, decomposing them into a set of independent tasks. The decomposition process includes the identification of the requests similarities by using our travel ontology, in order to formulate a set of autonomous and independent tasks which are waiting for responses at the same time t . Each task represents a service which can be proposed

by different Mobile Network (MN) nodes, with different cost, processing time and data size. To response to tasks, needed data is available through the MN and their collect correspond to the (ICA agents and TA agents) jobs. Therefore, the SA agent must optimize the assignments of nodes to tasks, minimizing total cost and processing time, in order to respect due dates. The optimize solution will be detailed through sections (4.2.3). We called the agents (IA, IdA, SA, ICA, TA and FA) created at the instant t , the agent society P_t (Figure 4.4).

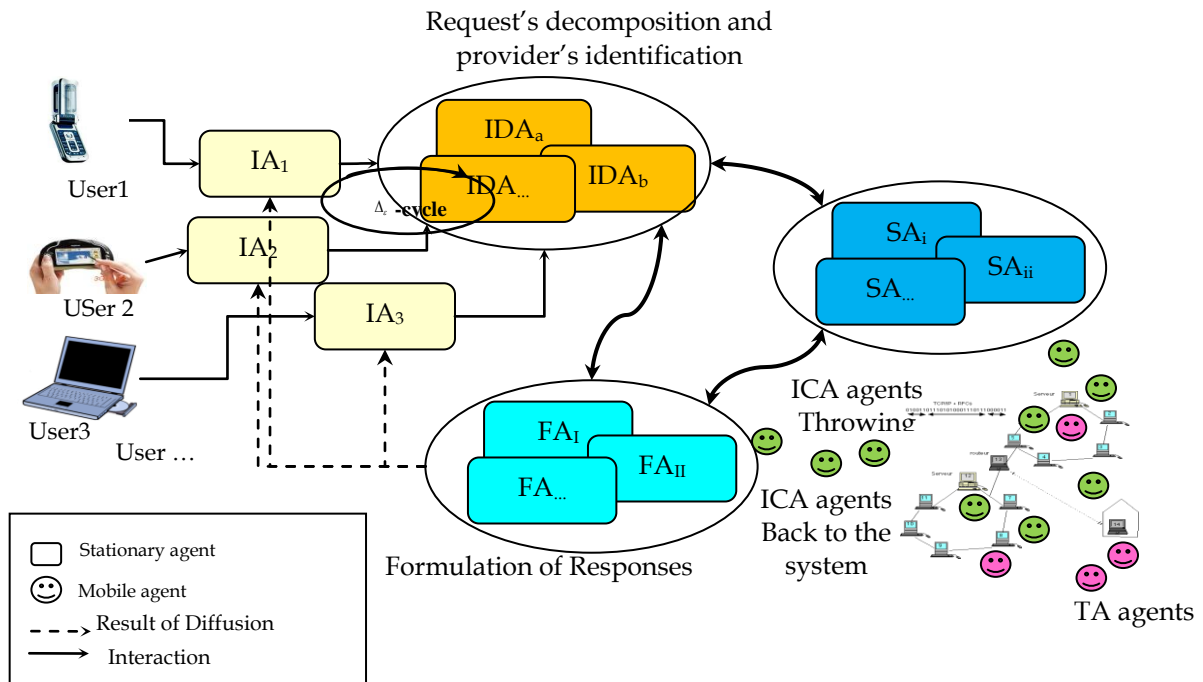


Figure 4.4 System Architecture

4.2.2.2. Agents Behaviours

The behaviour of various agents in the system is described in what follows:

4.2.2.3. Interface Agents (IAs)

These agents interact with system users, allowing them to choose appropriate form of responses to their demands so IA agents manage requests and then display results. When a MN customer access to the MIS, an agent IA deals with the formulation of his request and then sends it to an available identifier agent. This one relates to the same platform to which several users can be simultaneously connected, thus it can receive several requests formulated at the same time. An identifier agent has to identify and to choose nodes which propose services corresponding to the requests of the users. Figure 4.5.

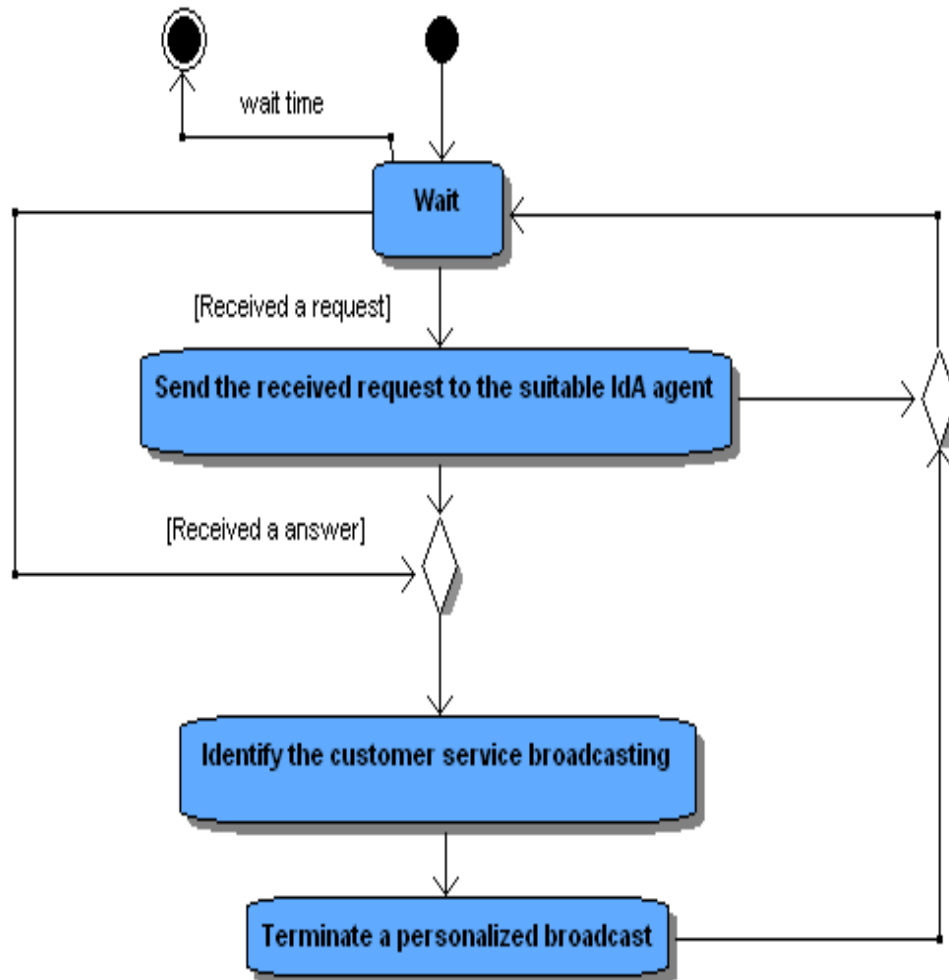


Figure 4.5. Interface Agent Behaviour

4.2.2.4. Identifier agents (IdAs)

These agents decompose received requests into sub-requests, corresponding for example, to sub-routes or to well-known geographical zones, according to their ontologies (see section 3.2). Sub-requests are elementary independent tasks to be performed by the available set of nodes (information providers) on the MN. Each node must login to the system registering all proposed services. A service corresponds to the response to a defined task with fixed cost, processing time and data (size and format). Therefore, the agent IdA decomposes the set of simultaneous available requests into a set of independent tasks recognizing possible similarities, in order to avoid a redundant data research. The decomposition process occurs during the identification of the information providers. Finally, the agent IdA transmits all generated data to available scheduler agents which must optimize the choice of MN nodes, taking into account some system constraints. Figure 4.6.

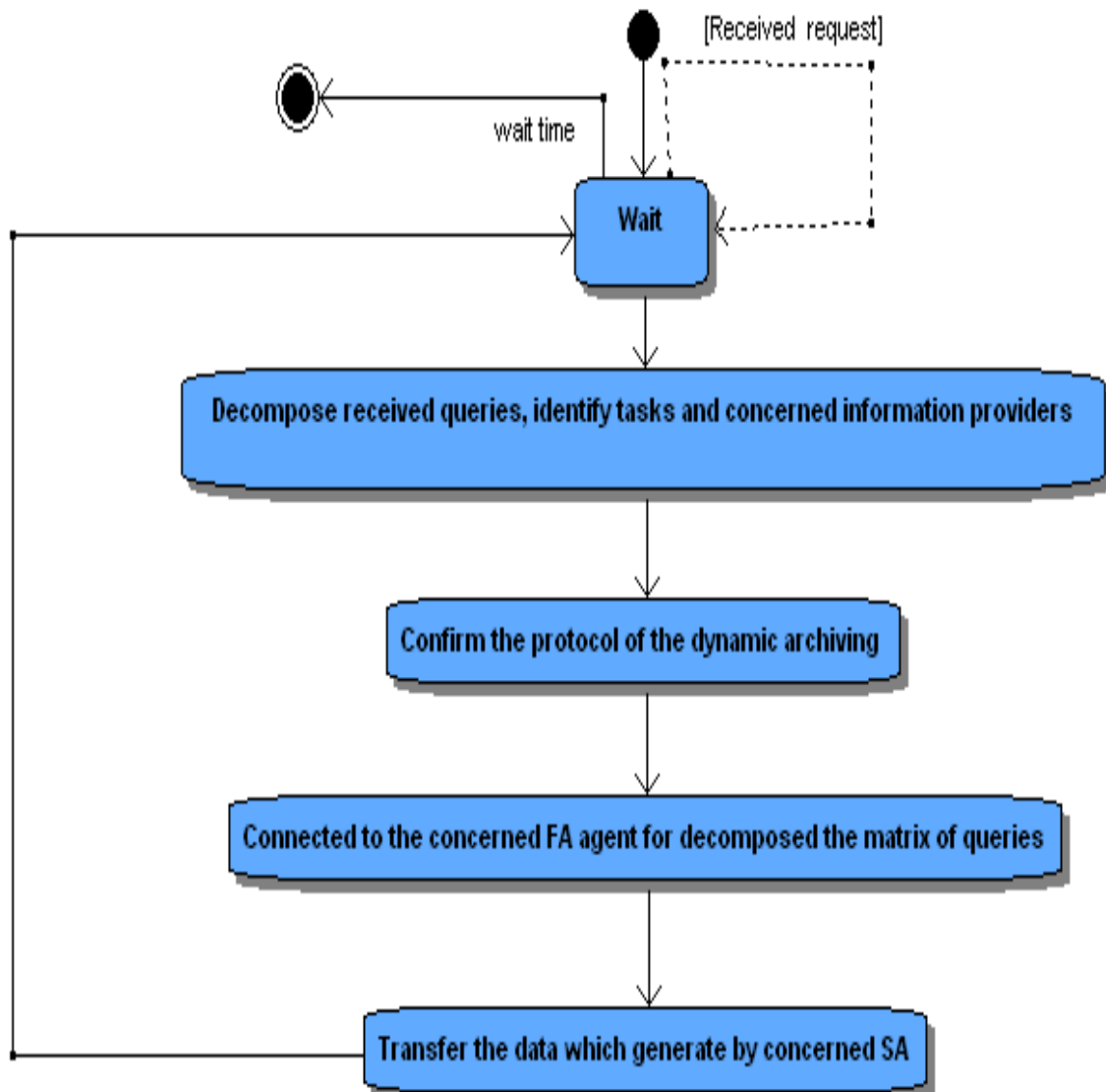


Figure 4.6. Identifier Agent Behaviour

4.2.2.5. Scheduler Agents (SAs)

Several nodes may propose the same service with different cost and processing time and data (size and format). The agent SA has to assign nodes to tasks minimizing total cost and processing time (QoS) in order to respect due dates (data constraint). Selected set of nodes corresponds to the sequence of nodes building Workplans (routes) of the data collector agents. Figure 4.7, the agent SA has firstly to find an effective number of collector agents then he has to optimize the assignments of nodes to different tasks. Some network errors can occur during the MAs moving through network nodes (bottleneck, failure, crash...) , this case called the perturbation case , where the SA agents began the negotiation process with the ICA agents to reassigned the tasks to novels nodes by using their ontologies, then , the

SA agent decide the optimize assignments of nodes. This behaviour will be developed later in section (4.3).

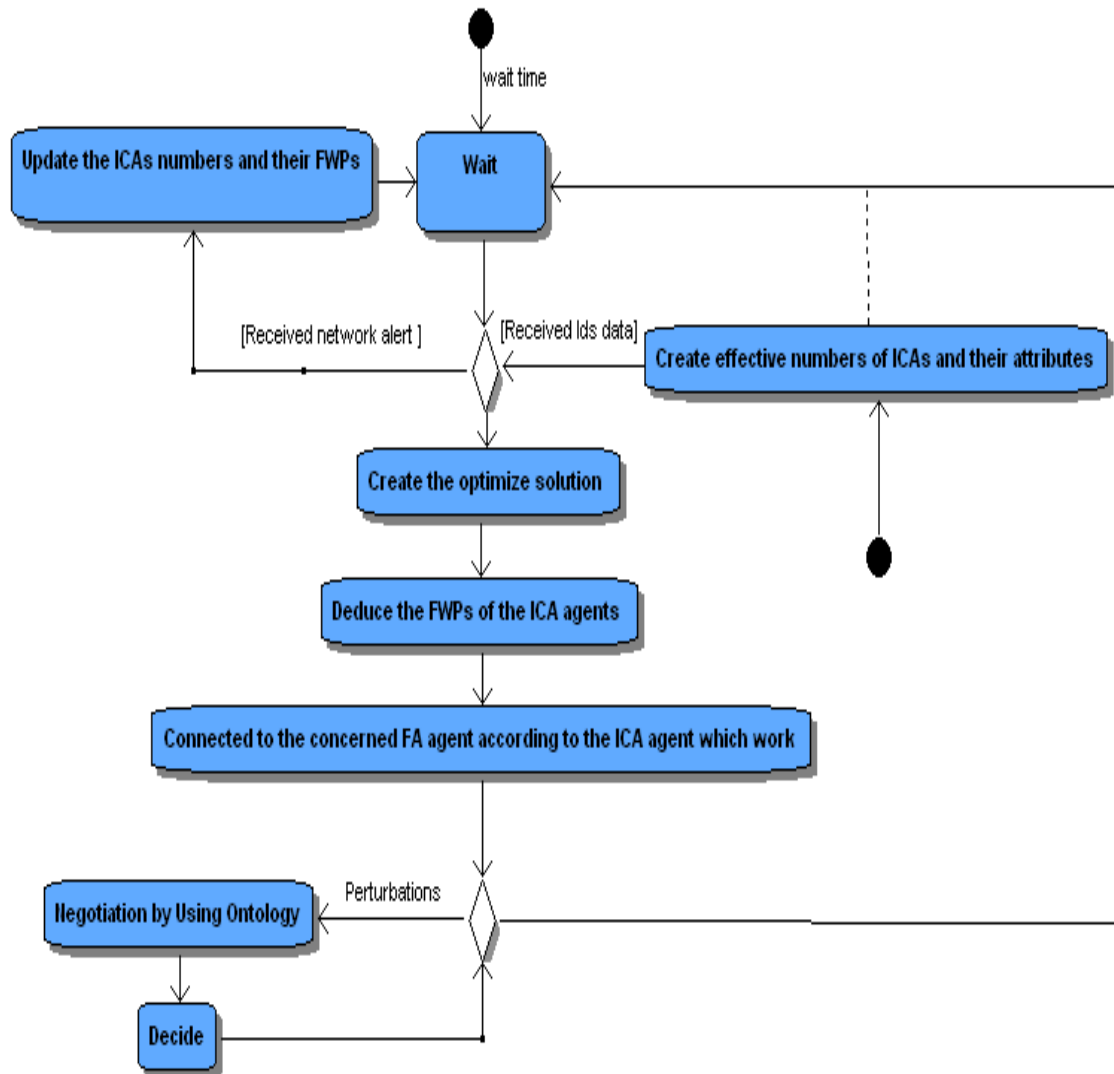


Figure 4.7. Scheduler Agent Behaviour

4.2.2.6. Intelligent Collector agents (ICAs)

An agent ICA is a mobile software agent which can move from a node to another through a network in order to collect needed data. This special kind of agent is composed of data, code and a state (section 2.3). Collected data should not exceed a capacity threshold in order to avoid overloading the MA. Therefore, the agent SA must take into account this aspect when assigning nodes to tasks. Thus, upon receive his FWP; an ICA will visit the network according the plan that has been assigned to him. The intelligent behaviour of the ICA explained by his ability to change his route in a real time while moving depending on the availability of network nodes in the

perturbation case. For this, the ICAs must interact with the SA agent to begin the negotiation tour according to the adopted negotiation protocol (see section 4.3). An ICA agent must propose a plan for rebuilding its remaining route from its current position, based on its priorities, preferences, constraints and ontologies. The SA agent, from his part, must negotiate with all corresponding active ICAs to reassign novel nodes to the services which have already assigned to them. But, in some case, the misunderstanding may be take place between those agents. So, the ICA agents have to connect with TA agents, and the later will do the necessary. When ICAs come back to the system, they must transmit collected data to available fusion agents and update the archiving data base. Figure 4.8.

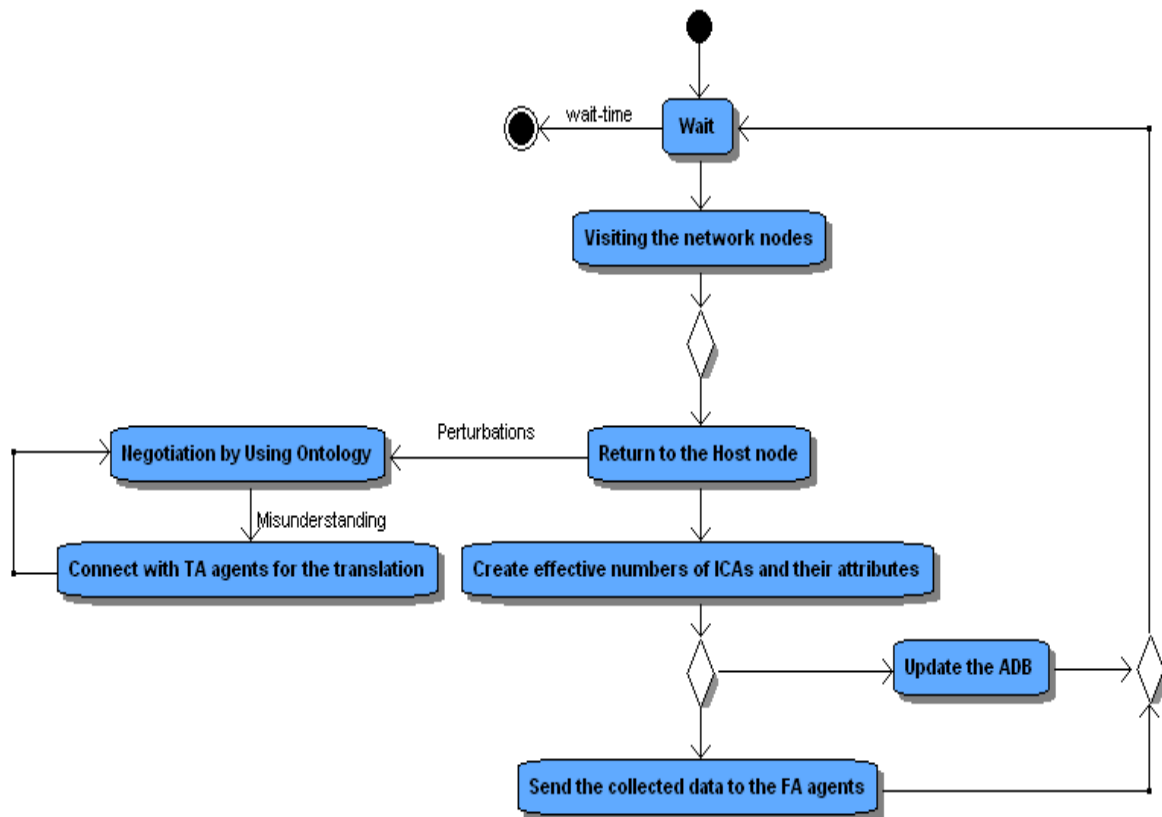


Figure 3.8. Intelligent Collector Agent Behaviour

4.2.2.7. Translation Agents (TAs):

TA responsible for providing the translation services that support the negotiation agents (i.e. SA agents and ICT agents). Thus, it helps solving the interoperability problems. TA uses a dictionary (or a lexical database, in our system, we use EuroWordNet) to obtain the set of synonyms terms of each term from the source ontology. The task of TA consists of applying methodology to detect semantic similarities between two concepts in the conversion between different ontologies. Figure 4.9.

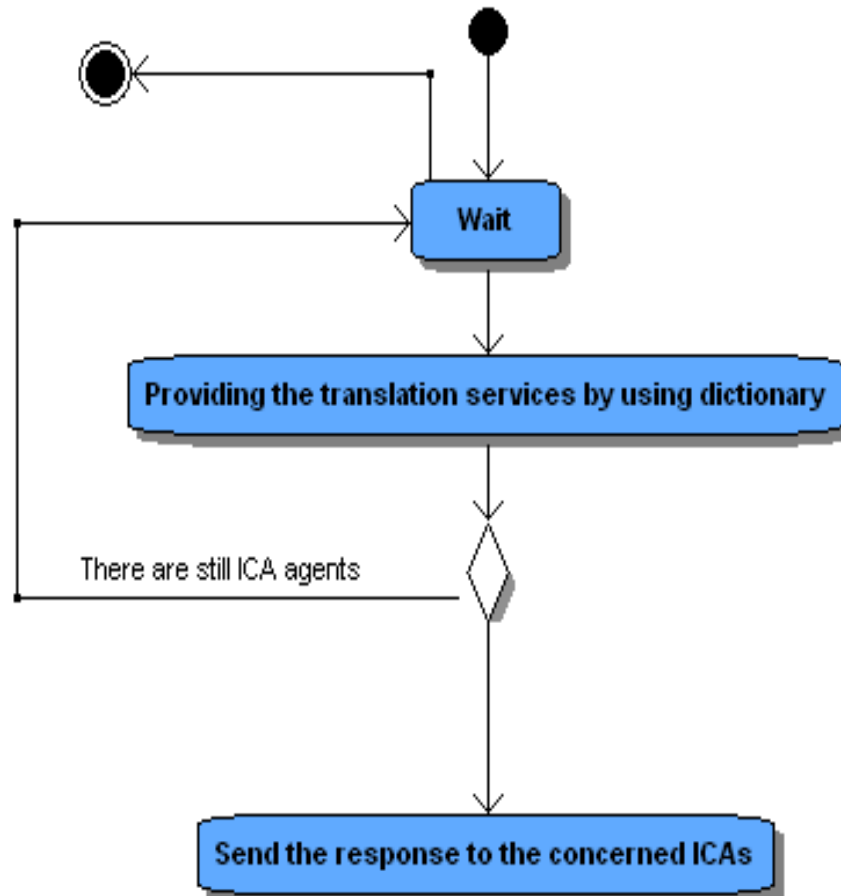


Figure 4.9 Translation Agent Behaviour

4.2.2.8. *Fusion Agents (FAs)*

These agents have to fusion correctly collected data in order to compose responses to simultaneous requests. The fusion procedure progresses according to the collected data availability. Each new answer component must be complementary to the already merged ones. Providers are already selected and tasks are supposed independent. Therefore, there is no possible conflict. A response to a request may be complete if a full answer is ready because all concerned components are available. It can be partial if at least a task composing the request was not treated, for example, because of an unavailable service. Finally, a response can be null if no component is available. If an answer is partial, the correspondent result is transmitted to the concerned user through the agent IA which deals with request reformulation, with or without the intervention of the user. Figure 4.10.

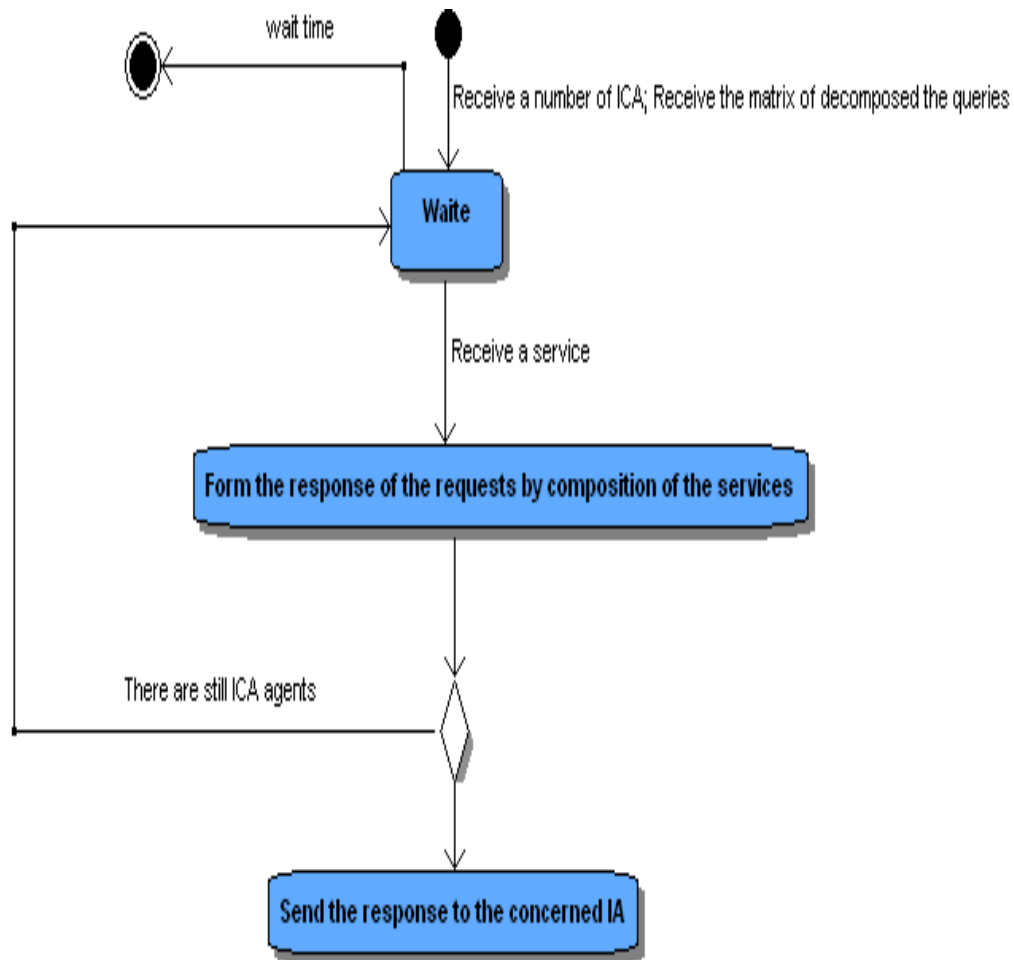


Figure 4.10. Fusion Agent Behaviour

The behaviour of the society P_t which represent the interacting between the intelligent agents, (consisting of an IA agent, an IDA agent, a SA agent, a number of ICA agents, a TA agent and finally a FA agent), is illustrated by the sequence diagram in figure 4.11.

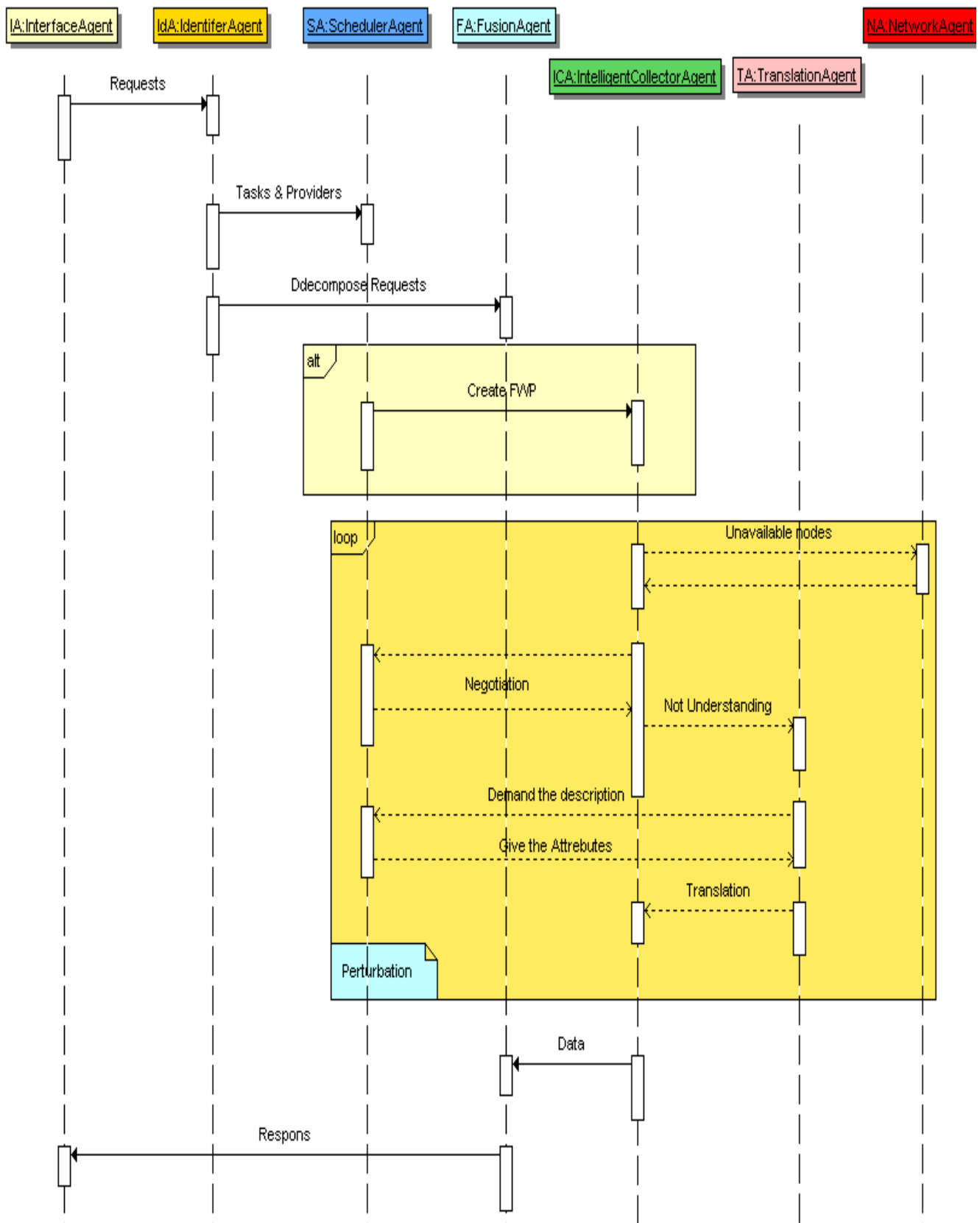


Figure 4.11. The Sequence Diagram of Agent Society

4.2.3. The Optimize Solutions by Scheduler Agents SAs

Since his creation, the SA agent calculates an actual number of ICA agents that created at the same time, and then he gives everyone an Initial Workplan (IWp) which updates whenever the network status varies considerably. When the IdA agent, from the same society P_t , gives him a number of tasks thus the SA agent has to begin the optimization process. The SA agent has to optimize the assignments of nodes to the exiting tasks, by minimizing total cost and processing time to respect due dates. To solve this assignment problem, we proposed a two level optimization solution, expressing the complex behaviour of an agent SA, which was already studied and implemented in previous works [Zgaya et al., 05a], [Zgaya et al., 05b] :

- The first level aims to find an effective number of ICA agents, building their initial Workplans in order to explore the ETMN completely [Zgaya et al., 05a].
- The second level represents the data flow optimization corresponding to the nodes selection in order to increase the number of satisfied users [Zgaya et al., 05b].

4.2.3.1. The WorkPlans Algorithm

Latency is the needed time for a data packet to cross a network connection, from sender to receiver. Therefore, using ICA agents can decrease considerably network traffic because they do not require simultaneous connection among different nodes. In a previous work [Zgaya et al., 05a], we proposed a *WorkPlan* design scheme which aims to find a suitable number of ICA agents minimizing their navigation time in order to explore all MN nodes, taking into account network latency. Our dynamic algorithm tries to find the next node from the current node where the agent resides. In other words, this algorithm looks for the next node for a part calculating new routing time. A node is selected if the new routing time does not exceed the threshold δ . Otherwise, a WorkPlan is ready to be assigned to an ICA and the algorithm ends if each MN node belongs to a routing path. We have to build our algorithm based on those definitions. Table 4.4.

Variable	Description
M	CA agents number
CA_1, \dots, CA_m	CA agents identifiers
H	Home node
Wk_i	Nodes sequence representing a CA_i agent Workplan: (S_{i1}, \dots, S_{ip}) with $1 \leq p \leq J$

$T(Wk_i)$	Routing time for Wk_i
$Qte_{k,u}$	The size of the data transported by CA_k from node S_u
$Tr(Qte_{k,u}, S_u, S_{u+1})$	Transmission time for $Qte_{k,u}$ from node S_u to node S_{u+1}
CT_i	Processing time on node S_i
Qt_i	Data quantity on node S_i
$L_s(S_i, S_j)$	Shortest latency between nodes S_i and S_j

Table 4.4. Notations

Algorithm 1. Workplans Algorithm

Step 1 : Sort nodes

$m:=0;$

for $i:=1$ to J

$Wk_i:=S_i;$

Sort nodes by decreasing order, according to their correspondent routing time:

$T(Wk_i=S_i);$

Let $R_d=(S_{d1}, \dots, S_{dn})$ be the resulting sequence, so

$\delta=T(Wk_1=S_{d1});$

Step 2 : Build Workplans

for $j:=1$ to J

{

$Wk_j = \emptyset;$

If \exists "unassigned" S_{dk} where k is minimum

{

Select it;

$m := m+1;$

$Union(Wk_j, S_{dk});$

Mark S_{dk} as "assigned";

```

    } // end if
else terminate; // end step2
while (true)
{
    Sort nodes according to  $L(S_{dk}, S_{dl})$  with
     $1 < l \leq J$  and
     $S_{dl}$  is "unassigned" by increasing order
    Let  $(S_{e1}, \dots, S_{eg})$  be the resulting sequence;
    previous_k := k;
    for y := 1 to g
    {
        Union( $Wk_j, S_{ey}$ );
        if  $T(Wk_j) \leq \delta$ 
        {
            mark  $S_{ey}$  as "assigned";
            find k while  $dk=ey$ ;
            break for-loop;
        } // end if
    }
    else  $Wk_j := Wk_j - \{S_{ey}\}$ ;
    } // end inner for-loop
    if previous_k = k
        break while-loop;
    } // end while-loop
} // end outer for-loop

```

Step 3: Run a simple TSP algorithm to optimize each $Wk_k \forall 1 \leq k \leq m$.

The algorithm above disperses all nodes into a set of ICA agents in order to explore the MN totally. This step fixes the needed number m of ICA agents and organizes their initial WorkPlans. In the next section, a sub-set S' of MN will be identified, thanks to an evolutionary method, in order to optimize the computing flow

management. In this case, the number of ICA agents is reduced. The goal is to optimize the selection of the nodes from the set of information providers proposing the required customers services. Final Workplans will be deduced from initial ones in order to collect needed data satisfying users' requests as soon as possible with reasonable costs. Let m' be the new number of ICA agents. We have also $J' = |S'|$ the new number of nodes so $m' \leq m$, $J' \leq J$ and $S' \leq S$.

4.2.3.2. The Data Flow Optimisation

In order to deduces final Workplans of ICA agents from initial ones, by using evolutionary algorithms (EA) which inspired from genetic algorithms by adding a new aspect to the field of artificial intelligence. We have designed an efficient coding for a chromosome (the solution) respecting the problem constraints [Zgaya et al., 05b]. A possible solution is an instance of a flexible representation of the chromosome, called Flexible Tasks Assignment Representation (FeTAR). The chromosome is a matrix CH ($I' \times J'$) where rows represent independent identified tasks (services), composing globally simultaneous requests and columns represent recognized distributed nodes (providers). Each element of the matrix specifies the assignment of a node S_j to the task T_i as follows

$$CH[i, j] = \begin{cases} 1: & \text{if } S_j \text{ is assigned to } T_i ; 1 \leq i \leq I' \text{ and } 1 \leq j \leq J' \\ * : & \text{if } S_j \text{ may be assigned to } T_i \\ X: & \text{if } S_j \text{ cannot be assigned to } T_i \end{cases}$$

We notice that each task must be assigned, so we assume that each task must be performed at least by a node, selected from a set of nodes proposing the service which corresponds to a response to the concerned task where this is the first selection step. After that, we apply the second selection step which is one of the most important aspects of all EA. It determines which individuals in the population will have all or some of its genetic material passed on the next generations. We have used random technique, to give chance to weak individuals to survey: parents are selected randomly from current population to crossover with some probability p_c ($0 < p_c < 1$).

In our case, we use the fitness function where a chromosome is firstly evaluated according to the number of responses which respect due dates, namely responses minimizing correspondent ending dates and respecting correspondent due dates. Then a solution is evaluated according to its cost. Therefore, a chromosome has to express ending responses date and the information cost. As we mentioned, a request

reqw is decomposed into It,w tasks. Therefore, the total processing time $EndReqw$ for each reqw is computed by the means of the algorithm *fitness_1* below. This time includes only the effective processing time on the MN. We assumed that, the ending date Dw corresponding to the total execution time of a request reqw, includes also the average navigation time of ICA agents. This is expressed by:

$$\gamma = \delta - \frac{\sum_{j=1}^J CT_j}{J} \quad (1)$$

$$\Rightarrow \forall 1 \leq w \leq R, D_w = EndReqw + \gamma \quad (2)$$

Fitness_1 algorithm

Step 1:

m' is the ICA agents number so

$\forall k$ with $1 \leq k \leq m'$, initialize :

- The set of tasks U_{ck} to \emptyset
- Total time $EndU_{ck}$ to perform U_{ck} to 0

Step 2:

Look for the set of tasks U_{ck} performed by each ICA_{ck} and their processing time $EndUk$ as follows:

for $k := 1$ to m'

for $j := 1$ to J'

for $i := 1$ to I'

if S_{cj} belongs to the Workplan of ICA_{ck} and S_{cj} is assigned to T_{ci}

{

$U_{ck} := U_{ck} \cup \{T_{ci}\};$

$EndU[ck] := EndU[ck] + P_{cicj};$

}

Step 3:

Compute processing time of each request require the identification of ICA agents which perform tasks composing the request. Total processing time of a request is the maximum processing times of all ICA agents which perform tasks composing this request. This is calculated as follow:

for $w := 1$ to R

{

for $k := 1$ to m'

$treatedAC[ck] := false;$

$EndReq[w] := 0;$

```

        i := 1;
        while i ≤ I' and ∃ k1/1 ≤ k1 ≤ m' and
            treatedAC[ck1] = false
        {
            if Tci ∈ reqw
            {
                ck := 1;
                while k ≤ m' and Ti ∉ Uk
                    ck := ck+1; //end while
                if ¬ TreatedAC[ck]
                {
                    EndReq[w] := max(EndReq[w], EndU[ck]);
                    TreatedAC[ck] := true;
                } //end if
            } //end if
        } //end while
    } //end outer for-loop

```

Form the other side, total cost of a request req_w is $CostReq[w]$ expressed by C_w , is given by the mean of the algorithm below:

Fitness_2 algorithm

Repeat steps 1 and 2 for each request req_w ($1 ≤ w ≤ R$)

Step 1:

$CostReq[w] := 0$

Step 2:

for $i := 1$ to I'

{

if $T_{ci} ∈ req_w$ {

find the node S_{cj} ($1 ≤ j ≤ J'$) assigned to T_{ci}
in FeTAR instance

$CostRe[w] := CostRe[w] + Co_{cicj}$

} //end if

} // end for

Knowing that by using expression (1), we can deduce ending date from *fitness_1* algorithm, the new FeTAR representation of the chromosome express for each request req_w $1 ≤ w ≤ R$, its ending date and its cost.

An example of a generated FeTAR instance with $I'=8$ and $J'=10$, where the evaluation of this chromosome is illustrated by a evaluation vector which

explicit: for each req_w , its total cost (C_w) and the total time required for his response (D_w). The average cost of all requests and the response time can be deducted from generated vector, can be illustrated as follows (Table 4.5):

w	d_w	C_w	D_w
1	10	5	6
2	5	1	1
3	10	4	2
4	5	3	2
4	3	2	1
6	5	3	2

CH	S ₁	S ₁₃	S ₂₄	S ₅₅	S ₆₈	S ₇₀	S ₇₁	S ₇₈	S ₇₉	S ₉₃
T ₈	*	*	*	*	1	*	*	*	*	*
T ₁₂	*	*	*	*	x	*	*	1	*	*
T ₁₈	*	1	*	*	x	*	*	*	*	*
T ₂₂	*	*	*	*	x	*	1	*	*	*
T ₃₅	X	*	1	x	x	x	*	*	x	x
T ₅₁	X	X	*	*	x	x	x	1	*	*
T ₅₂	*	*	*	1	x	x	x	x	*	*
T ₅₈	*	*	*	1	x	x	*	*	*	*

Table 4.5. Example of a FeTAR Instance

In this thesis, we are interested in the interaction between SA agents and ICA agents, especially in case of some network disturbances. In that case, these two kinds of agents have to negotiate the reassignment of tasks which still need providers. We will illustrate that in the rest.

4.3. Ontology Mapping Negotiation Model

4.3.1. Perturbation Situation

Some perturbations can occur during the mobile agents moving through the distant network nodes (bottleneck, failure, crash...). In this case, the ICA agents have to avoid the unavailable nodes in their remained FWps. In addition, they have to change their itineraries in order to take into account the cancelled tasks which still need assignment because of the conflicts. Therefore, a new assignment process has to occur to find suitable new available providers [Saad et al., 08c]. To do this, we have to benefit of active ICA agents who are still travelling through the network and to exploit new ones otherwise. So ICA agents have to interact with SA agents in order to find suitable solution to the current situation. Thus, in [Zgaya et al., 07c], we propose a negotiation process inspired from the well-known contract net protocol (CNP) between the ICA agents who represent the participants of the negotiation and SA agents who are the initiators.

At the beginning, our protocol has studied only the cases of the simple messages and it took into account neither the ontology, nor the problem which take place when the participants don't understand the communication messages, or when the new agent wants to participate in a negotiation process. Thus agent must understand the protocol and the communication language messages. In this case the agents need an interoperable language between themselves for understanding each other. But as we know in open and dynamic environments (such as the Web and its extension the Semantic Web) are by nature distributed and heterogeneous. In these environments ontologies are expected to complement agreed communication protocols in order to facilitate mutual understanding and interactive behaviour between such agents. Thus, agents may differ in their view of the world, creation, representation and exploration of domain ontologies they commit to. Because, for each common domain ontology; people may store their data in different structures (structural heterogeneity) [Malucelli and Oliveira, 04]. And they use different terms to represent the same concept (semantic heterogeneity). Moreover there is no formal mapping between ontologies.

4.3.2. A Negotiation Ontologies based on Knowledge Management Systems (NOKMS)

To avoid the perturbation situation, we propose a general architecture for negotiation process which uses ontology-based knowledge management system, [Saad et al.,08a] and [Saad et al.,08b]; Figure 4.1

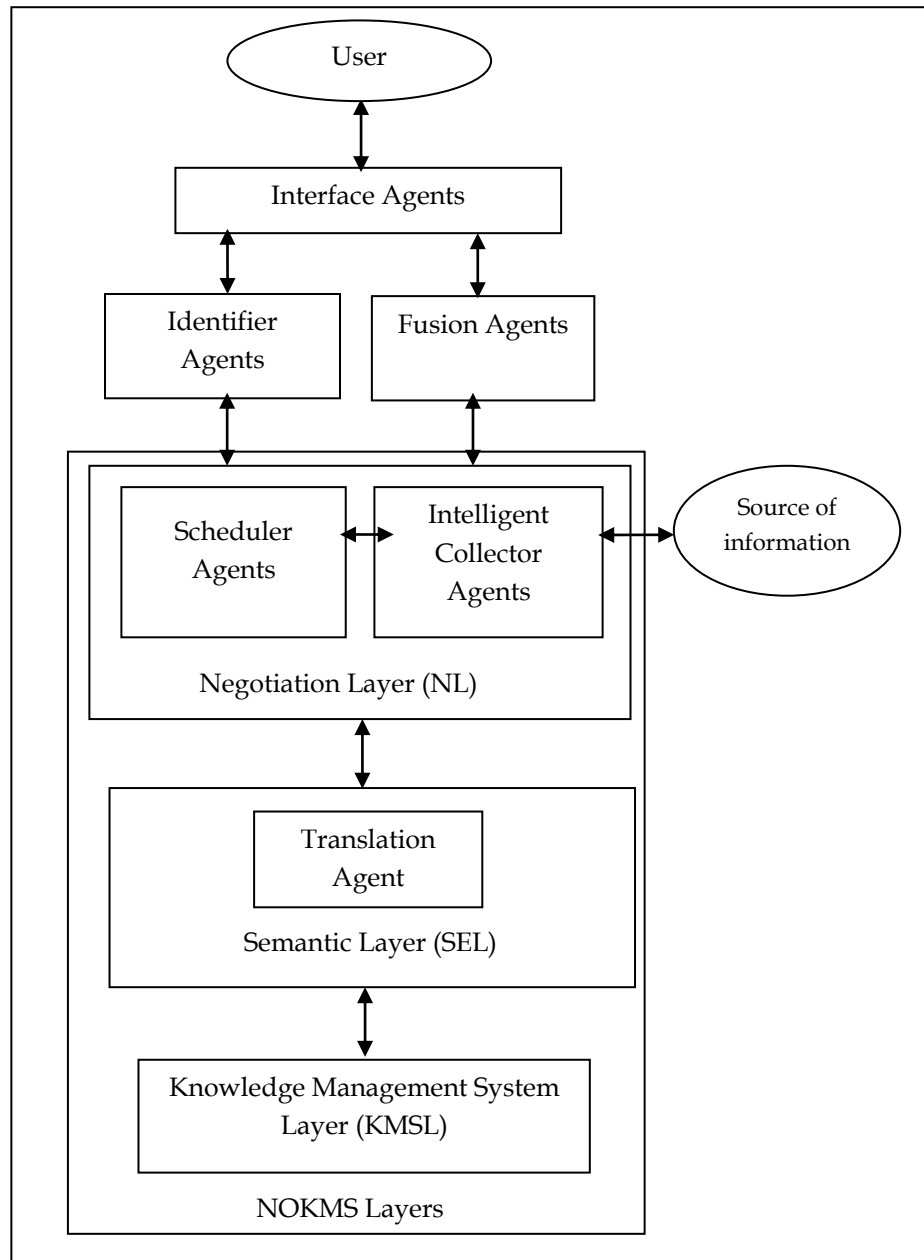


Figure 4.12. Multi-Agents Structure

We organize our architecture as follow: the first layer contains the *Negotiation Layer (NL)* where the initiator static agents (which are SA agents in our system) send the first message to the participant mobile agents (which are the ICAs agent) to start the negotiation process. The second layer represents the *Semantic Layer (SEL)*, in the case of not understanding the negotiation messages; The SEL uses a translator semantic (which is a TA agent in our system) in order to help it to translate automatically the various types of exchanges messages between the different agents. When, the agents don't have suitable ontologies which contain the suitable vocabulary for their communications and their negotiations. The third layer is the Knowledge

Management Systems Layer (KMSL) uses ontology in purpose of automatic classifying and using of the news ontologies and meta-ontologies.

4.3.2.1. *Negotiation Layer (NL)*

The first Layer of our architecture is the negotiation layer where the negotiation is defined as a process, whose transitions and states are described by the negotiation protocol, that agents have to follow for interaction, we will illustrate this layer in details in (section 4.4).

The agents participate in the negotiation by using their languages for formulating negotiation messages in order to interact and to take the decision. The language used by the agent to interact and execute the exchange of the messages and knowledge is called Agent Communication Language (ACL) like (section 2.3.2). In our proposal, (Figure 4.13), NL contains the initiators (SAs), agent communication language (ACL) and the participants (ICAs) in the negotiation process.

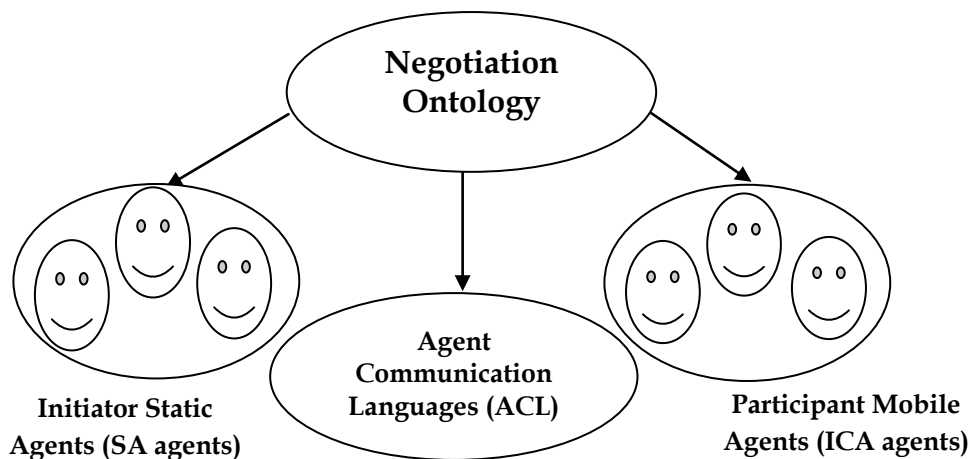


Figure 4.13. Negotiation Layer (NL)

In this layer, the initiators start the negotiation process by sending the ACL messages to the participants. The later have to understand the negotiation protocol and the communication language messages. Thus, the problem will tack place when the participants don't understand the communication messages, or when the new agent wants to participate in a negotiation process. In this case, the agents need an interoperable language between themselves for understanding each other. We find that the best solution is to use ontology as we illustrated in previous section. Indeed, our negotiation layer represent the negotiation process, where we present the negotiation protocol (section 4.4) which uses a flexible ontology and this protocol

allows a partial agreement from each ICA agent, to be confirmed partially or totally by the initiator of the negotiation (SA agent). It allows too the renegotiation process, if necessary for the rest of tasks which need to be reassigned (section 4.5).

4.3.2.2. *Semantic Layer (SEL)*

The second layer in our architecture is SEL. in fact, negotiation process will be easier handled when we use the our travel ontology (chapter 3). Ontology can be regarded as a vocabulary of terms and relationships between those terms in a given domain. Ontologies have been studied in different researches domains because they facilitate the communication among the negotiation agents where ontology is used as an Interlingua. Our purpose is to find a solution especially in the case of misunderstanding of the negotiation messages among the agents.

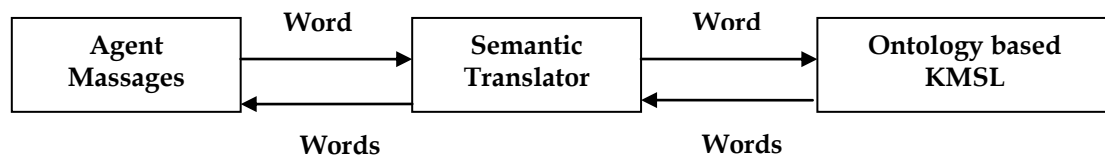


Figure 4.14. Semantic Layer (SEL)

For this reason, (Figure 4.14), SEL helps the system in its research to find the best optimize solution. SEL uses the semantic translator (TA agent) which, in turn, translates the not understanding messages that sent from the initiators (Propose, Confirm, etc ...). And vice versa for the messages of the participants (Agree (total, Partial), refuse, etc...). The formula which must be used to solve this problem is:

$$\langle \text{Sender}, \text{Receiver}, \text{Ontology1}, \text{Ontology2}, P \rangle$$

P: a predicate which used to determine the relationship among the ontologies (languages) and decide the level of transibility between the initiator ontology (language) and the participant ontology (language).

$$P = (\text{onto-relationship } ?\text{Ontology1 } ?\text{Ontology2 } ?\text{Level})$$

Where

$$\text{Level} : \{ \text{Weakly-Translatable}, \text{Strongly-Translatable}, \text{Approx-Translatable} \}.$$

In SEL, TA agent examines the level of transibility among the ontologies by sending a word (concept and in our system each concept represent a service) to the ontology-based KMSL which resend the set of semantically equivalences words (concepts). In fact, the ontology-based KSML connect with KMSL to answer the query of the

semantic translator which determines the level of transibility to facilitate the translation process (section 4.4).

4.3.2.3. Knowledge Management Systems Layer (KMSL)

Basically; the role of Ontology in the Knowledge Systems is to facilitate the construction of domain model. A meta-ontology and knowledge model, which is necessary for this construction and usage, describe the primitives used by a knowledge representation language, like (concepts, individuals, relations, etc).

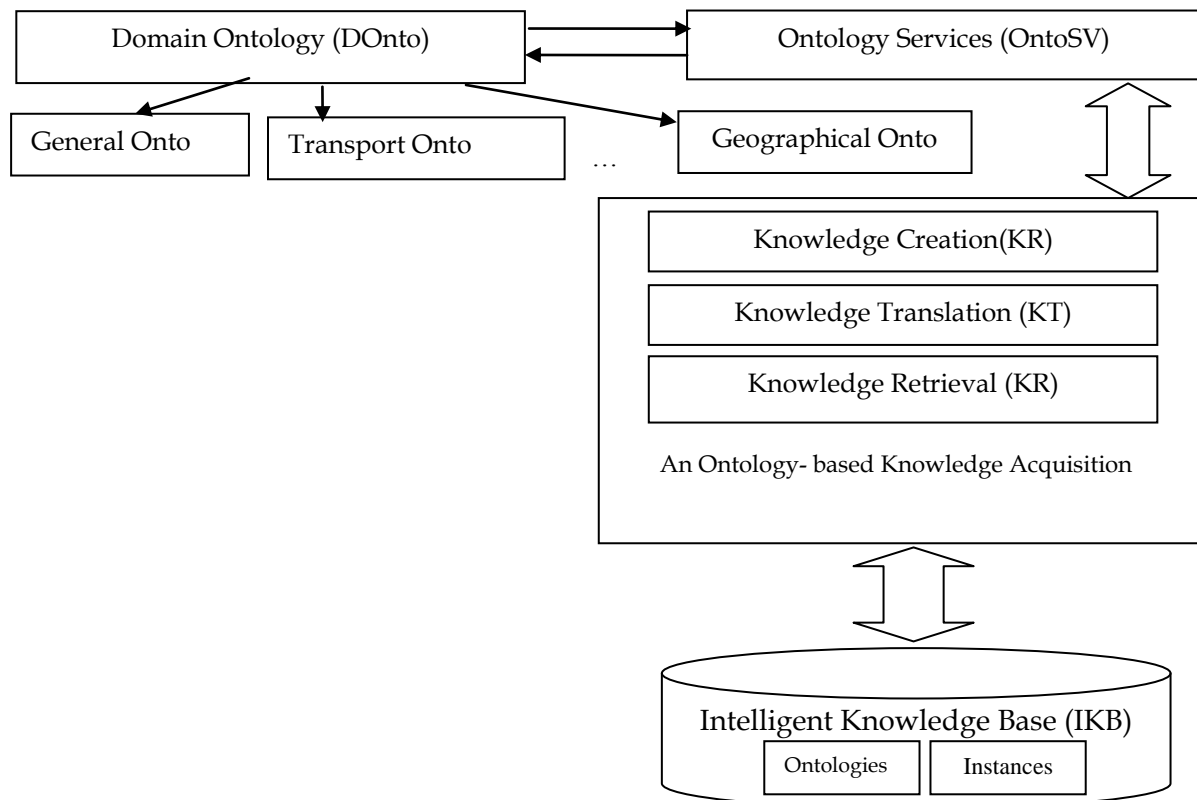


Figure 4.15. Knowledge Management System Layer (KMSL)

In this section, we introduce the Knowledge Management Systems Layer (KMSL). The architecture of this layer, as it represented in Figure 4.15, consists of:

- **Domain Ontology (DOnto):** DOnto contains the list of application domains in which the ontology is applicable. By using this domain, the agents communicate with each other through common domain knowledge. A general ontology (our travel ontology) defines the vocabulary with which queries and assertions are exchanged between agents. DOnto gives the flexibility to the negotiation ontologies which can capture the valid

knowledge for different domains (e.g. Transport domain, Geographic domain, etc).

- **Ontology Services (OntoSV):** The task of OntoSV is to define the semantics of ontologies (actions, predicates used in the content of the conversation with the Ontology Agents (AOs), (see chapter 3)) which the agents use to interact with each other, and support the knowledge acquisition operations (Creation, Translation, and Retrieval). OntoSV can adopt Open Knowledge Base Connectivity (OKBC) [Geiger, 95], as fipa-meta-ontology (an ontology used to access the OAs). The syntax of translation process used in our OntoSV, and applied by the TA agent, is defined as follow:

<Sender, Receiver, Ontology1, Ontology2, F>

F: the translation process service. This service is applied to translate expressions (terms, sentences) among translatable ontologies (i.e. before using this translation action, the SEL must check whether the ontologies are translatable or not by using the predicate P which illustrated in the previous section).

F = (translate <expression> <translate-description>)

<translate-description> = (translate <From A> <To B> <Level>)

(A, B) could be an ontology or a language. Level is the levels of transibility among the ontologies which can be calculate by the predicate P.

- **Knowledge Acquisitions:** is a very important part in the ontology process because they are used to create a new DOnTo or languages, to perform the translation among ontologies and to retrieve the knowledge from the Intelligent Knowledge Base; it consists of the following parts:
 - *Knowledge Creation (KC):* this operation is used to create a new ontology with a new Donto when a new agent wants to participate in the negotiation and he has not the appropriate ontology, or when the ICAs don't understand the ontology because they have not this ontology, in these cases the KMSL executes KC process.
 - *Knowledge Translation (KT):* translates the terms and sentences among ontologies (Languages). But before that, it uses the results of the SEL to verify whether the ontologies are translatable or not.
 - *Knowledge Retrieval (KR):* agents can access to the meta-ontology (our travel ontology) through a query interface (as we explained in the

previous chapter). In fact, they use their ontology to view the ontology individuals in the intelligent knowledge base.

- *Intelligent Knowledge Base (IKB)*: Ontology is defined as terminological component of a Knowledge Base (KB) (section 2.2). In addition, an agent uses its knowledge base which contains theorems to reason about the application domain. Combining these two approaches, each agent of (MAS) holds a KB which based on the domain ontology (application domain). In our KMSL; ontology and a set of instances of classes (the ontology individuals) together constitute our Intelligent KB (IKB). IKB uses the OKBC, which in turn, connects to a wide variety of IKBs servers where these IKBs are applied the Knowledge Acquisitions.

Finally, after have seeing the general view of our architecture, we will explain the details of different layers in the next sections.

4.4. Ontology Negotiation Protocols

As we have presented in last section, the negotiation layer contains the negotiation process whose transitions and states are described by the negotiation protocol, that agents have to follow for interactions. In this section, firstly, we describe the negotiation initiators (4.4.1). Then, we present the different participants in negotiation process (4.4.2). The illustration of our negotiation protocols will be in section (4.4.3). Finally, the different forms of agent's messages will present in section (4.4.4).[Saad et al.,10].

4.4.1. Negotiation Initiators

An initiator of a negotiation is a SA agent who never knows the exact position of each travelling ICA agent. However, he knows all initial Workplans (IWPs) schemes and the final assignments of the servers (final effective Workplans). SA agent does not need to wait for all answers to make a decision, since he can accept a subset of responses to make pressing sub-decisions; urgent actions must be made according to the current positions of ICA agents. Consequently, SA agent can make decisions every short period of time. In that case, he must update the set of services which need to be reassigned by providers through the confirmation step. After that, he has to propose a new contract according to the updated services set and to the different capabilities of the participants of the negotiation. We suppose that errors on the network are identified before that an ICA agent leaves one functioning node towards a crashed one.

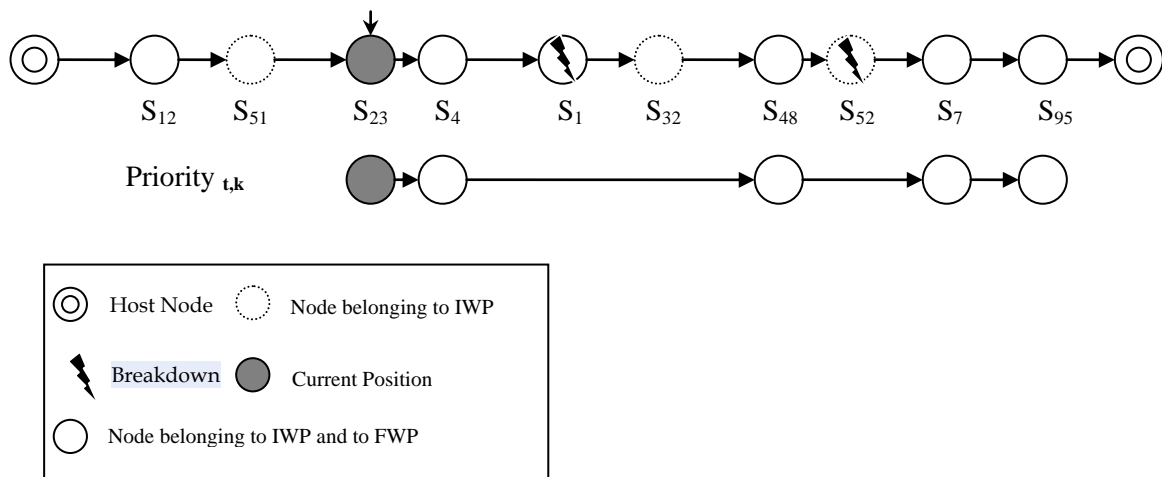
4.4.2. Negotiation Participants

For a given task, the participants may respond with a proposal or a refusal to negotiate. In our protocol we have two types of participants in negotiation process according to the SA agent propose.

4.4.2.1. Intelligent Collector Agents (ICAs)

A participant of a negotiation is a ICAs agent who never knows anything about the other participants of the same negotiation process. Obviously, he knows his own IWp scheme and his final assignments of servers (final effective Workplan). In addition, each ICA agent has his own priorities, preferences and constraints which are dynamic, depending on the network state and on his current position in the already defined FWp. He has own ontology too.

- **Priorities** express servers where the ICA agent prefers visit because they are already programmed in his remained final Workplan (FWp).Figure 4.16



Figuer 4.16. Example of The Agent Priority

- **Preferences** express servers which are already programmed in the remained initial Workplan (IWP) but not in the final one. Figure 4.17

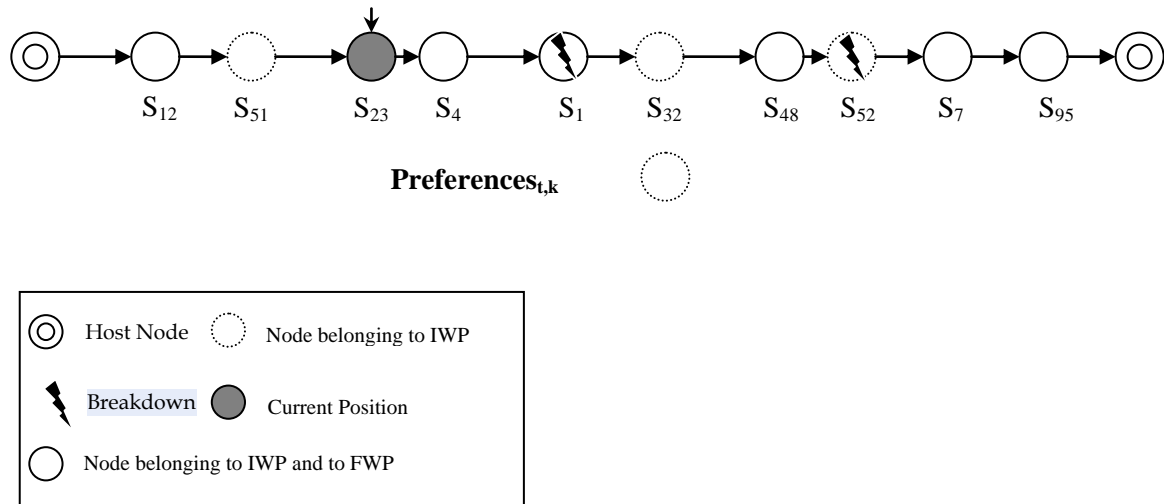


Figure 4.17. Example of The Agent Preference

- **Constraints** of an ICA agent express the tasks which he can't perform or the servers which he can't visit because they cause problems (overloading, time consuming, high latency...).Figure 4.18

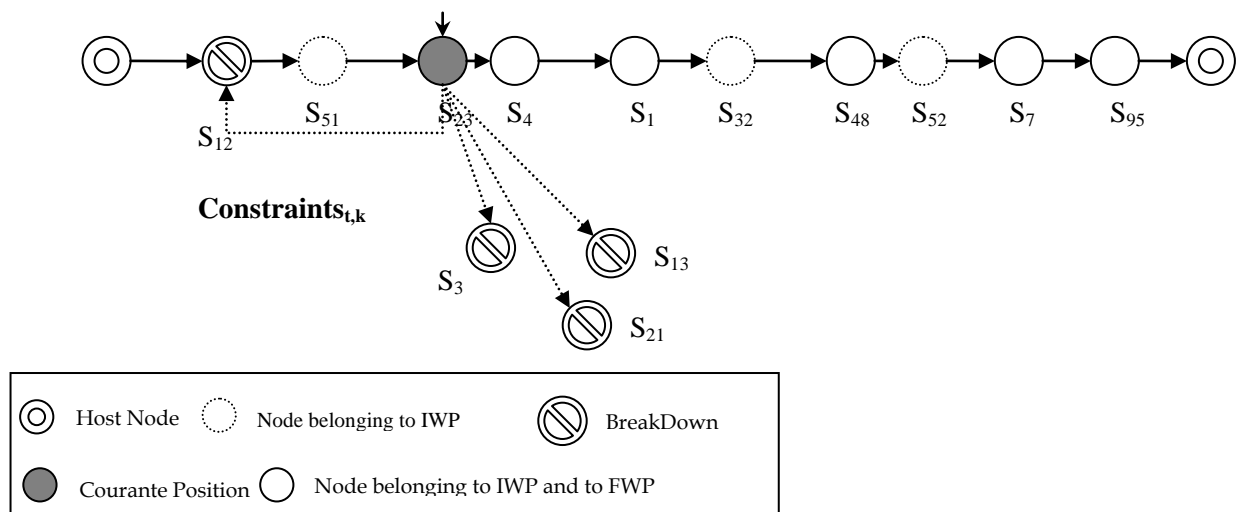


Figure 4.18- Example of The Agent Constraint

- **Ontology** if we expect that all agents share same ontology which is General Ontology. The later defines the Communication Vocabularies (Cv) with which queries and assertions are exchange between agents, then, the agents can understand each other easily. But one of the big problems to communication-based agents is that each one uses different terms with the same meaning or the same term for different meanings. Once we took this problem as a challenge, representing these differences in a common ontology

becomes essential. The ontology includes the entire domain's knowledge, which is made available to all the components active in an information system. The use of a common ontology guarantees the consistency (an expression has the same meaning for all the agents) and the compatibility (a concept is designed, for the same expression, for any agent) of the information present in the system (chapter 3). However, it is not sure that all the agents will use a common ontology. Usually, each agent has its heterogeneous private ontology and it cannot fully understand other agent's ontology. In our system, each time an ICA agent receives a new contract; it analyzes it to make a decision (refusal or total/partial acceptance) according to its ontology.

4.4.2.2. *Translation Agents (TAs)*

Another participant of a negotiation is a Translation Agents TAs. TA responsible for providing the translation services that support the negotiation agents (i.e. SA agents and ICT agents), in the case of not understanding the negotiation messages. Thus, it helps solving the interoperability problems. TA uses a dictionary (or a lexical database) to obtain the set of synonyms terms of each term from the source of general ontology. The task of TA consists of applying methodology to detect semantic similarities between two concepts in the conversion between different ontologies. Once the TA has established the similarity between a pair of terms from different ontologies, this knowledge is stored in Knowledge Management System Layer (KMSL) [Saad et al., 2008b], in order to be available for future negotiation rounds. The intelligent of this system is improved occurs with time, because the matched terms is memorized, when the number of negotiations rounds increases. We aim that our system by using TA provides the following services:

- **Mapping Terms Service (MTS):** in common domain ontology, people may store their data in different terms to represent the same concept (semantic heterogeneity). We take the following negotiation example: regarding travel query. A customer needs to reserve a travel by train from Lille to Paris. We define the "Concept" (e.g., "Means-Of-Transport") and an information provider offers the seam reservation but it is uses the "Concept" (e.g. "Transport-Mode"). In this case, both terms belong to the transport domain, but they are syntactically different and semantically equivalent. The two agents, unaware of the misunderstanding, are likely to engage in negotiation. Then, when the negotiation could be fruitful, they fail to understand each other.

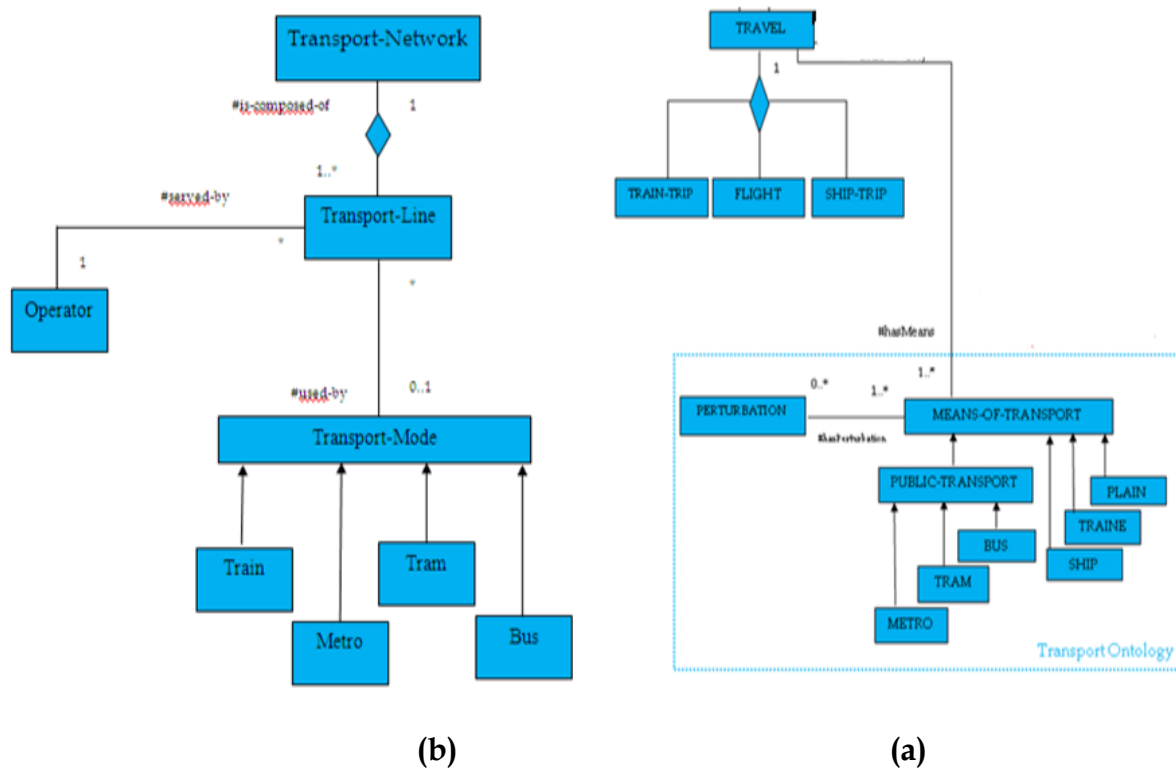


Figure 4.19. Partial View of the Transport Ontologies of both the SA and ICA Agents

Our system domain describes the scenario of a travel domain where a SA engages in a negotiation process to reserve a travel from different information providers, those providers registered in our system (section 3.4). There are the possibility when we do the negotiation process the receiver of the message don't understand the concept because it is not listed in its ontology.

Figure 4.19, (a) and (b) shows two UML diagrams represent partial views of the ontologies of a SA agent (our general travel ontology) and ICA agent (information provider ontology). Figure 4.19 (a) illustrates a partial view of the SA agent ontology (transport ontology partition of our travel ontology as we explained in previous chapter), while Figure 4.19 (b) represents a view of the ICA agent ontology (Transport Ontology). Both views are composed of a set of concepts. Each concept has a description in natural language, relationships with other concepts.

In these examples we may observe some differences that will cause interoperability problems during the negotiation process as we motioned for the difference between the "Means-Of-Transport" concept in our Travel Ontology and "Transport-Mode" in the local ontology of information provide. In our scenario, the SA in our system and ICA agent which visit the server of

the provider to collect the travel information, have the same objective: they want to provide a travel from Lille to Paris by train in the same application domain (transport domain). And each agent want still use its own private ontologies. Due to this common objective, we provide our travel ontology (chapter 3) which contains transport vocabulary, i.e., for using our ontology (as a global ontology) in our negotiation protocol. This vocabulary contains terms which are used during the negotiation process.

- ***Translation Services (TS)***: here we discuss the translating ontologies in the context of Multilingual Ontology Mapping. We exemplified the negotiation between two transport systems that use two different ontologies (English and French) languages, respectively. We represent as the terms “Destination” in the source ontology is mapped to the term “Arrivée” in the target ontology. These terms represent the destination areas related to client travel. This services dose not covered in this thesis.

4.4.3. Ontology Negotiation Protocols

During the negotiation process, we need the initiators and the participants (as we have explained) and the negotiation protocol. Now, we can illustrate our negotiations protocols. The implementation of our negotiation process combines the Ontology Negotiation Protocol (ONP) which will interact with an additional protocol called Ontology Mapping Protocol (OMP). We will explain the two protocols later. We adopt the formula of the Agent Communication Language **ACL**²⁰ messages is as follow :

<Sender, Receiver, Services, Performative, Contents, Language, Ontology, Protocol>

- ***Sender***: the identity of the sender of the message.
- ***Receiver***: the identity of the intended recipients of the message.
- ***Services***: the "yellow pages" proposed by the recipient of the message
- ***Performative***: the type of the communicative act of the ACL message. The performative parameter is a required parameter of all ACL messages. *Performative* = {Propose, Agree (total, Partial), Confirm, Cancel, Not Understood...}. We explain the use of each performative of communicative acts according to FIPA²¹, in the Table 4.6.

²⁰<http://www.fipa.org/specs/fipa00061/SC00061G.html>

²¹<http://www.fipa.org/specs/fipa00037/SC00037J.html>

- **Content:** the content of the message. The meaning of the content of any ACL message is intended to be interpreted by the receiver of the message.
- **Language:** the language in which the content parameter is expressed.
- **Ontology:** the ontology(s) is used to give a meaning to the symbols in the content expression (terms, relations, etc...).
- **Protocol:** the protocol which is used to described by the negotiation mechanism

Performative	Description
Propose	The action of submitting a proposal to perform a certain action, given certain preconditions.
Accept	The action of accepting a previously submitted proposal to perform an action.
Confirm	The sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition.
Not Understood	The sender of the act (<i>i</i>) informs the receiver (for example, <i>j</i>) that it perceived that <i>j</i> performed some action, but that <i>i</i> did not understand what <i>j</i> just did. A particular common case is that <i>i</i> tells <i>j</i> that <i>i</i> did not understand the message that <i>j</i> has just sent to <i>i</i> .
Refuse	The action of refusing to perform a given action, and explaining the reason for the refusal.
Inform	The sender informs the receiver that a given proposition is true.
Query-Ref	The action of asking another agent for the object referred to by a referential expression.
Cancel	The action of one agent informing another agent that the first agent no longer has the intention that the second agent performs some action.

Table 4.6. FIPA Performative of Communicative Acts

The usage of this formula is very easy when the agents interact by exchanging the messages which contain the same ontology. But the semantic interoperability problems take place when the sharing information and knowledge use different ontologies, or when there are multiple ontologies which resemble a universal ontology. How can we use the message formula in our system? We will illustrate that in the section 4.4.4.

4.4.3.1. *Ontology Negotiation Protocol (ONP)*

The first protocol is Ontology Negotiation Protocol (ONP) which represents the general scenario of negotiation process. The SAs agents start the negotiation process by sending the messages to the ICAs agents. As we illustrated previously, we search to find the solution when there are some network errors and the agents search to find suitable new available providers for new assignment process. Here, the ICA agents participate in the negotiation by using their languages for formulating negotiation messages in order to interact and to take the decision.

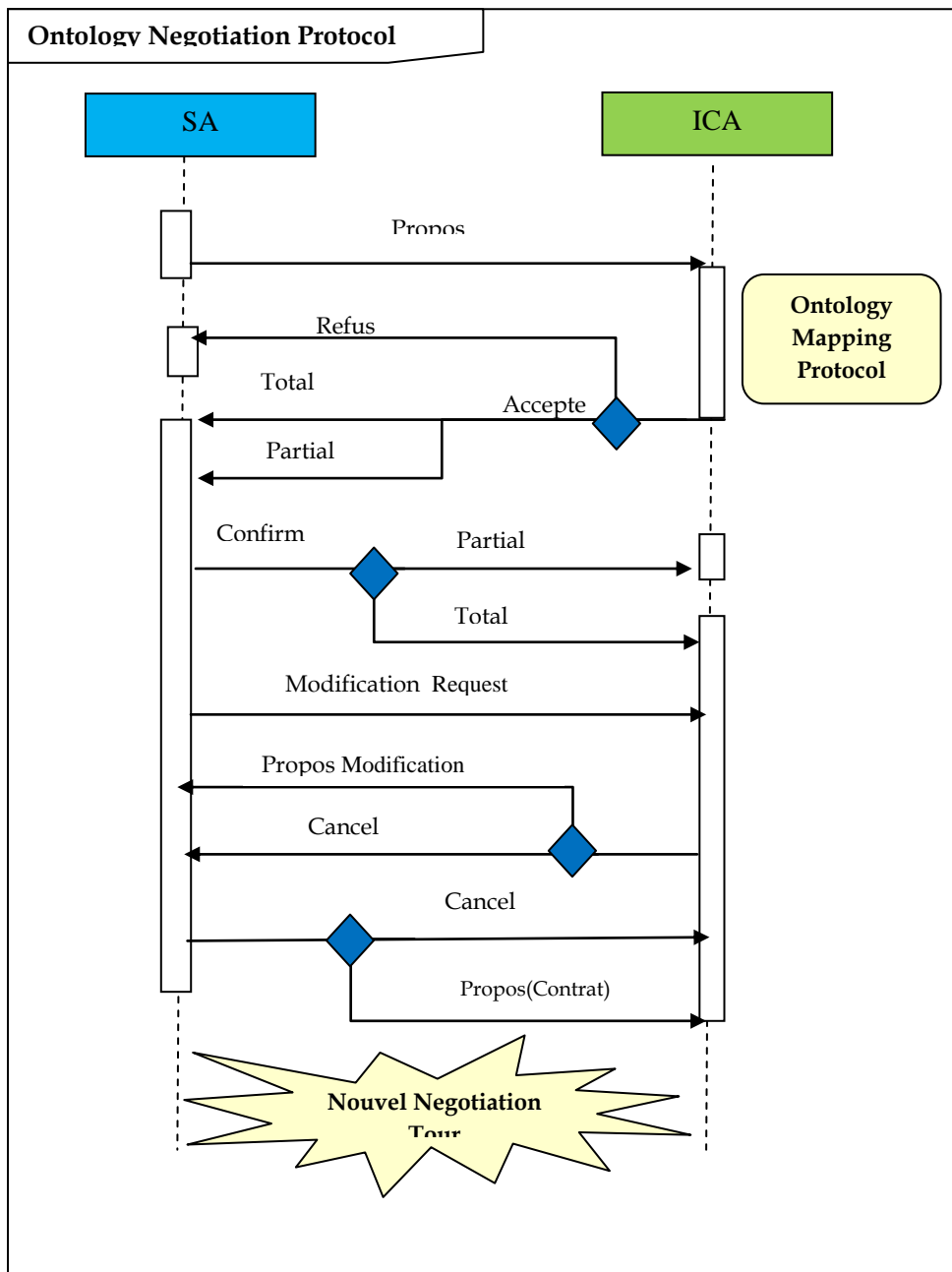


Figure 4.20. Ontology Negotiation Protocol (ONP)

Our Ontology Negotiation protocol (ONP), (Figure 4.20), is characterized by successive messages exchanges between SA agents and ICA agents. We designed our protocol so that a negotiation process can occur between several initiators and participants, it can be, for example, the case of simultaneous requests overlapping, but it is not the purpose of this thesis. Here, we describe the ONP between a unique initiator and several participants. In our ONP, we allowed a partial agreement of the proposed contract from each ICA agent, to be confirmed partially or totally by the initiator of the negotiation (SA agent).

A renegotiation process is necessary while there are still tasks which need to be reassigned. The purpose of this solution is to allow the ICA agents to cooperate and coordinate their actions in order to find globally near-optimal robust schedules according to their priorities, preference, constraints and ontologies. This solution depends on their current positions in their correspondent Workplans. Through the negotiation process tours, SA agents must assure reasonable total cost and time. We will detail the different exchanged messages between initiators and participants in next paragraph.

4.4.3.2. *Ontology Mapping Protocol (OMP)*

As we mentioned previously that another problem may take place when the participants don't understand the communication messages, or when the new agent wants to participate in a negotiation process then he has to understand the protocol and the communication language messages.

For implements the message flow which is necessary for solving the problems of interoperability, including the interaction of SA and ICA agents when requesting/receiving a service. We designed the Ontology Mapping Protocol (OMP) with the purpose of facilitating the interaction between the agents and services (Figure 4.21).

After having received an ONP and not being able to interpret the requested service, the ICA sends a message with the performative NOT_UNDERSTOOD to the TA. He asks him who sent the ONP and the name of the unknown service. The TA sends the name of the service which it has just received to the SA in order to get further information about it. The SA will analyze that request and send back attributes of the concept, i.e. all the information about this service.

Upon receiving the answer from SA, the TA knows the description, of the demanded service under negotiation and sends it to the ICA. The later selects among all service

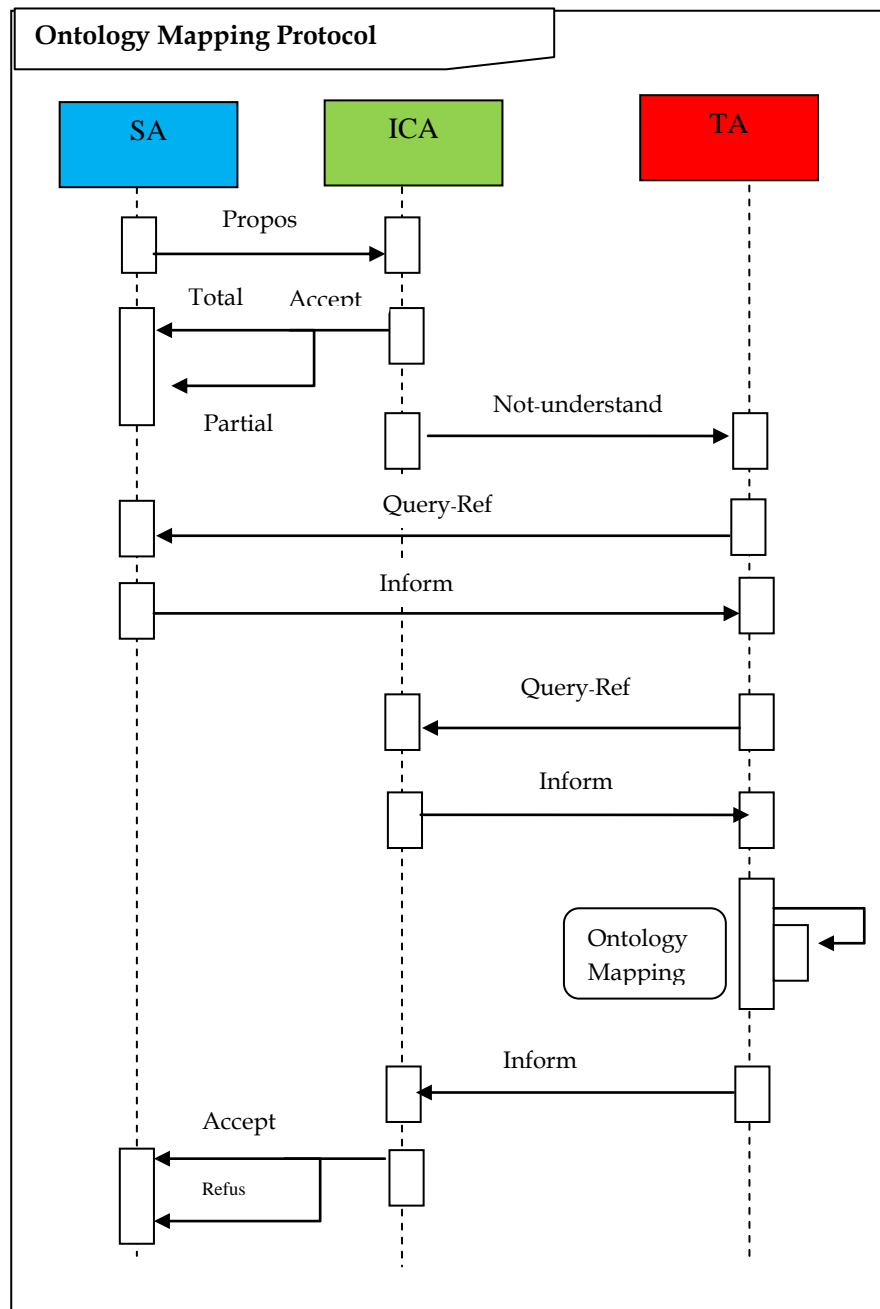


Figure 4.21. Ontology Mapping Protocol (OMP)

the ones whose time value is near of the received value. After the selection, the ICA answers with a list containing names of potential correspondent concepts.

After receiving all the information about the service under negotiation and a list of possible corresponding services, the TA is able to apply methods in order to match the services. So we have applied the Quick Ontology Mapping (QOM) method (section 4.5), where this method aims to detecting semantic similarity of terms, [Saad et al., 08a]. Every term of the proposed, potential correspondent service is compared to the requested term. By using QOM method, we apply the first task of our OMP which is the Mapping Terms Service (MTS). For the second service which is

Translation Services (TS), we test the level of transibility between the two ontologies by applying the predicate P (section 4.3.2). Then, The TA tries to translate the different expressions (terms, sentences) among the ontologies through the translation process which used in OntoSV.

In final step, the TA informs the ICA about the result of the comparisons delivered from the ontology mapping methods. The ICA is then able to respond to the SA, either with a ACCEPT or with a REFUS that is part of our ONP.

4.4.4. Agents Messages

In the last section, we proposed a structure for our ONP and OMP protocols. In what follows, we detail the different exchanged messages between initiator and participants in negotiation process.

4.4.4.1. *Proposition of the contract:*

The contract message is a proposition of a new organization (the first contract) or reorganization of final Workplans to achieve tasks. If the execution of some services was cancelled because of some network perturbations, it is indeed the case of reorganization. This will be done by reassigning one more time servers to these tasks which represent the set of the Dynamic Reassigned Tasks (DRT) [Saad et al., 08c]. The initiator sends an individual contract to each active ICA_k agent who proposes the contract-reception service:

$$\langle SA_i, ICA_k, contract-reception, propose, \partial, fipa-sl, Ontology, protocol \rangle$$

With $\partial = \partial_1$ if it acts of the first contract and $\partial = \partial_2$ otherwise:

$$\begin{aligned} \partial_1 &\equiv \text{Workplan (Owner : ICA}_k & \partial_2 &\equiv \text{FinalWk (Owner : ICA}_k \\ &\text{Initial : } i_1, \dots, i_{k_i} & &\text{Final : } f_1, \dots, f_{k_f}) \\ &\text{Final : } f_1, \dots, f_{k_f}) \end{aligned}$$

With i_1, \dots, i_{k_i} represent references of nodes which belong to the initial Workplan of the ICA agent k (ICA_k) and f_1, \dots, f_{k_f} represent references of nodes which belong to the final Workplan of the same agent. Thus we have $k_i \leq k_f$.

4.4.4.2. *Response to the contract:*

When a participant receives the proposed contract, he studies it and answers by:

- **Total Acceptance** : if he agrees to coordinate all tasks chosen by the initiator, included in his remaining trip (remained final Workplan), according to his current position,

$$\langle ICA_k, SA_i, \emptyset, \text{accept-proposal}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

- **Partial Acceptance**: if he agrees to coordinate a subset of the tasks selected by the initiator, included in his remaining trip (remained final Workplan) or if he doesn't understand the received message sending by the initiator. Then, according to his current position, the partial-accept-proposal message content expresses the references of cancelled tasks and those of unavailable servers (the reason of the non total-acceptance):

$$\langle ICA_k, SA_i, \emptyset, \text{partial-accept-proposal}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

$$\text{With } \partial \equiv (\text{tasks: } t_1, \dots, t_n \text{ nodes : } s_1, \dots, s_m)$$

- **Refusal**: if he does not agree with any task in the proposed contract (i.e. he uses the ONP for check the services only) or if he doesn't understand the received message sending by the initiator (i.e. he didn't understand the message, here he uses OMP to analyze the message). Then, the refusal message content expresses the references of unavailable servers (the reason of the refusal):

$$\langle ICA_k, SA_i, \emptyset, \text{refuse}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

$$\text{With } \partial \equiv (r_1, \dots, r_m)$$

The initiator does not wait for all answers because he must act rapidly, so he just waits for some answers for a very short period of time to make a decision.

4.4.4.3. Confirmation:

An initiator has to confirm independently the agreed part of each contract k proposed to an agent ICA_k who represents an autonomous participant of the negotiation, the confirmation can be:

- **Total**: if the initiator agrees with the total response to the previous proposed contract ,

$$\langle ICA_k, SA_i, \emptyset, \text{confirm}, \emptyset, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

- **Partial:** if the initiator agrees with a partial response to the previous proposed contract, the partial-confirm-proposal message content expresses the references of agreed tasks:

$$\langle ICA_k, SA_i, \emptyset, \text{partial-confirm-proposal}, \partial, \text{owl-dl}, \text{ontology}, \text{protocol} \rangle$$

$$\text{With } \partial \equiv (g_1, \dots, g_p)$$

4.4.4.4. *Modification request:*

If the DRT table is not yet empty (Saad et al., 2008a); the initiator asks the participants to propose a new distribution of services assignments which are canceled, the request-modification message content expresses the DRT table:

$$\langle SA_i, ICA_k, \emptyset, \text{request-modification}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

$$\text{With } \partial \equiv (\text{DRT})$$

4.4.4.5. *Modification proposition:*

According to our DRT algorithm, where we design a reassignment procedure strategy of servers to tasks, , taking into account not only the dynamic positions of ICA agents in their Workplans, but also their constraints, priorities, preferences and ontologies, according to their respective current positions. The proposition message content expresses for each participant k the new proposition of his remained Workplan according to his current state:

$$\langle ICA_k, SA_i, \emptyset, \text{propose}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

$$\text{With } \partial \equiv \text{FinalWk (Owner: } ICA_k, \text{ Final: } f_1, \dots, f_{k_f})$$

Where f_1, \dots, f_{k_f} represent references of nodes which belong to the final Workplan of the agent ICAk.

4.4.4.6. *Quit:*

After have sending the conformation. The participants (or the initiator) don't want to continue the negotiation process. Then, he decides to stop the process. In this case, if the DRT table is not empty, the initiator can resend another contract to the participants. the desist message content is as follow:

$$\langle SA_i, ICA_k, \emptyset, \text{desist}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

With $\partial \equiv (\text{DRT})$

4.4.4.7. *Not Understand:*

In our system the problem of heterogeneity may arise; when one of ICA_k agents receives the message and it don't understand the concepts. Then ICA Agent will send a message to the TA, setting the performative of the ACL message to NOT UNDERSTOOD.

The TA Agent will examine the level of transibility between the ontologies correspondent. Then, he accesses to the services provided by the KMSL (OntoSV) which helps, in this case helping, to solve the existing heterogeneity problem. He tries to facilitate the negotiation process (i.e, reduce the number of negotiation rules), the not understood message will to be, as follow:

4.4.4.8. $\langle \text{ICA}_k, \text{SA}_i, \emptyset, \text{not understood}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

With $\partial = (c_1, \dots, c_n)$

4.4.4.9. *Cancel:*

To avoid indefinite waiting for answers or for modifications, the initiator agent must make a decision at the end of a fixed period of time, illustrated by the last field of an agent message. Therefore he cancels the contract if there is no more solution (lack of resources, no available provider...) or he creates new ICA agents to execute the current contract:

$\langle \text{SA}_i, \text{ICA}_k, \emptyset, \text{cancel}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

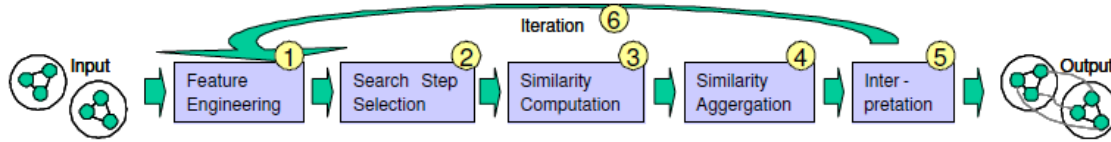
4.5. Dynamic Reassigned Tasks (DRT) algorithm with Ontology Mapping Approach

4.5.1. Ontology Mapping Approach

We consider that each provider of the services when it registers its services in the system, it registers its ontology too. But ontologies have many of combination problems where in open multi-agent systems, communication problems that arise from heterogeneous ontologies should be solved, rather than avoided.

In this section, we apply the Quick Ontology Mapping (QOM) methodology for identifying mappings between two ontologies based on the intelligent combination

of ontology features and similarity measures. QOM is one of tools which are used to solve the ontology mapping problem. As described in section 2.7.3, QOM is defined by the steps of the process model as follows:



Figuer 4.22. Mapping Process

1. Feature Engineering: transforms the initial representation of ontologies into a format digestible for the similarity calculations.

2. Search Step Selection: All entities of the first ontology are compared with all entities of the second ontology for to find the best mappings.

3. Similarity Computation: The similarity computation between an entity of the first ontology O1, and an entity of the second ontology O2 is done by using a wide range of similarity functions. Each similarity function is based on a feature and the respective similarity measure. For QOM, they are shown in the following Table 4.7.

4. Similarity Aggregation: QOM does not aggregate the individual similarity results by a linear combination, but before aggregation it employs a function that emphasizes high individual similarities and de-emphasizes low individual similarities.

5. Interpretation: uses the individual or aggregated similarity values to derive mappings between entities from O1 and O2.

6. Iteration: Several algorithms perform iteration over the whole process in order to bootstrap the amount of structural knowledge. Iteration may stop when no new mappings are proposed.

	No.	Feature	Measure
Concept Similarity	1	Label	string similarity
	2	URI	string equality
	3	sameAs relation	explicit equality
	4	direct prosperities	SimSet
	5	all inherited prosperities	SimSet
	6	all super-concepts	SimSet
	7	all sub-concepts	SimSet
	8	concept siblings	SimSet

	9	direct instances	SimSet
	10	instances	SimSet
Relation Similarity	1	Label	string similarity
	2	URI	string equality
	3	sameAs relation	explicit equality
	4	domain and range	object equality
	5	all super-properties	SimSet
	6	all sub- properties	SimSet
	7	properties siblings	SimSet
	8	property instances	SimSet
Instance Similarity	1	Label	string similarity
	2	URI	string equality
	3	sameAs relation	explicit equality
	4	all parent-concepts	SimSet
	5	property instances	SimSet
Property-instance Similarity	1	domain and range	Object equality
	2	parent property	SimSet

Table 4.7. QOM: Features and Measures for Similarity

4.5.2. The Dynamic Reassigned Tasks (DRT) algorithm

The Dynamic Reassigned Tasks (DRT) algorithm can be formally defined by:

- The realization of each task requires a renewable resource (server), or node, selected from a set of J available nodes, noted $S = \{S_1, \dots, S_J\}$,
- The set of J' nodes ($J' \leq J$) selected from S , assigned to tasks to perform them, is noted $S' = \{S'_1, \dots, S'_{J'}\}$, ($S' \subseteq S$),
- The set of Ω nodes ($\Omega \leq J$) which are not available, is noted $S_c = \{S_{c1}, \dots, S_{c\Omega}\}$, ($S_c \subseteq S$),
- The initial Workplans are built to explore the total network. They contain α_i nodes for each ICA_i and it is noted as follow: $IW_{ki} = (S_{v1}, \dots, S_{v\alpha_i})$ for each ICA_i
- The final Workplans built to perform tasks. They contain β_i nodes ($\beta_i \leq \alpha_i$) for each ICA_i and it is noted $FiW_{ki} = (S_{w1}, \dots, S_{w\beta_i})$ for each ICA_i ,
- The remained final Workplans depends on the current position of an ICA_i at a moment t . They contain Δ_i nodes for each ICA_i and it is noted as follow: $ReFiW_{ki}(t) = (S_{x1}, \dots, S_{x\Delta_i})$ for each ICA_i ,

- The remained final Workplans $ReFiWk_i(t)$ does not contain the crashed nodes; if a node in $ReFiWk_i(t)$ is no more available, it is directly removed from it and added to the set of crashed nodes S_c ,
- $S_a \prec_{Wk_i} S_b$ means that S_a precedes S_b in the Workplan Wk_i correspondent to the agent ICA_i
- If a Workplan Wk_i of an agent ICA_i where $Wk_i = (s_{s_i}, \dots, s_{e_i})$ then $s_{e_i} = H$ (the home node)
- $L_s(S_a, S_b)$ corresponds to shortest latency between servers S_a and S_b .

ICA agents have to ignore crashed nodes in their remained routes, so they have to avoid visiting them. However, they have to find substitute nodes to perform tasks initially assigned by these crashed nodes. Consequently, according to the actual position in their Workplans, each ICA agent has to update its remaining route, if it is possible. This will be done by reassigning substitute servers tasks which need to be reassigned; we call this set the Dynamic Reassigned Tasks (DRT). This reassignment depends on the actual positions of ICA agents in their final Workplans. It depends also on their ontologies, constraints, priorities and preferences. The new assignment constitutes a contract between ICA agents and SA agents. Let R be a variable which corresponds to the number of ICA agents not yet finished their travelling through the network, so the DRT process algorithm is described as follows:

Algorithm DRTOnto
Step: =1; While ($R > 0$ and $DRT \neq \emptyset$) Case of Step: Case 1 : For $i := 1$ to R do Build the contract according to the priorities of the agent ICA_i and update DRT table; /* see paragraph A.*/ Step++; Case 2 : For $i := 1$ to R do When the response is not understand then build the contract according to the priorities of the

agent ICAi by applying the QOM algorithm

and update DRT table;

/* see paragraph A.*/

Step++;

Case 3 :

For i := 1 to R do

Build the contract according to the preferences

of the agent ICAi and update DRT table;

/*see paragraph B.*/

Step++;

Case 4 :

For i := 1 to R do

When the response is not understand then build

the contract according to the preferences of the

agent ICAi by applying the QOM algorithm

and update DRT table;

/*see paragraph B.*/

Step++;

Case 5 :

For i := 1 to R do

Build the contract according to the constraints of the agent ICAi

and update DRT table;

/* see paragraph C.*/

Step++;

Case 6 :

For i := 1 to R do

When the response is not understand then build the contract

according to the constraints of the agent ICAi

by applying the QOM algorithm

and update DRT table;

/* see paragraph C.*/

Step++;

Step := Step-5;

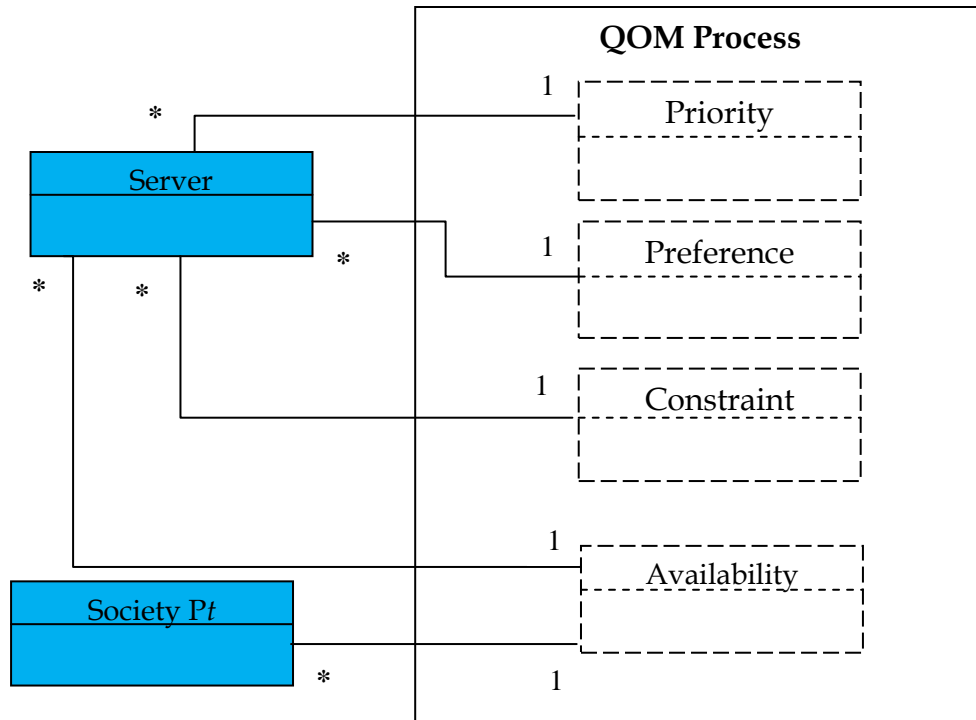
Dispatch the contract over the R active ICA agents;

Update R;

If (DRT $\neq \emptyset$)

Create new ICA agents according to our Workplan algorithm [12].

The UML diagram which represent the DRT process algorithm with the QOM , as follow:



Figuer 4.22. DRTOnto Process

A. Build the contract according to priorities:

We remind here that Δ_i corresponds to the number of nodes constituting the remained final Workplan for each ICA_i agent at a moment t . Where in this part there are two algorithm deals with agent priorities: the first when the agent understands the received message and the second when he doesn't understand it. We illustrate the two algorithms in the following:

For $j := \Delta_i$ down to 1 do

Look for the set of tasks $\mathcal{T}_j \subseteq DRT$ and S_{xj}

Propose services for \mathcal{T}_j where $S_{xj} \notin S_c$ and S_{xj} in $ReFiWk_i(t)$;

with respect to $Min(t)$ and $Min(cost)$.

```

If ( $\mathcal{T}_j \neq \emptyset$ ) {
    Update ReFiWki(t);
    /*ICAi will perform  $\mathcal{T}_j$  in Sxj */
    DRT := DRT -  $\mathcal{T}_j$ ; //update DRT }

```

The idea is to explore nodes where the agent ICA_i will visit soon, in order to perform tasks from DRT, on nodes which belong to his remained Workplan. With this intention, we propose to begin this exploration from the last node on the remained Workplan to execute the tasks which belong to \mathcal{T}_j .

The second algorithm applies the QOM algorithm for doing the mapping between the agent ontologies according to the agent priorities.

```

For j:=  $\Delta_i$  down to 1 do
    Look for the set of tasks  $\mathcal{T}_j \subseteq$  DRT and Sxj
    Propose services for  $\mathcal{T}_j$  where Sxj  $\notin$  Sc and Sxj in ReFiWki(t);
    Apply the QOM mapping algorithm,
    with respect to Min(t) and Min (cost).
    If ( $\mathcal{T}_j \neq \emptyset$ ) {
        Update ReFiWki(t);
        /*ICAi will perform  $\mathcal{T}_j$  in Sxj */
        DRT:= DRT -  $\mathcal{T}_j$ ; //update DRT }

```

B. Build the contract according to preferences:

We remind here that α_i corresponds to the number of nodes constituting the initial Workplan for each ICA_i agent. Especially for the following algorithm, we use a Travelling Salesman Problem algorithm (TSP) [Carey et al., 79] to optimize remained itinerary:

```

For j := 1 to  $\alpha_i$  do

    Look for the set of tasks  $\mathcal{T}_j \subseteq$  DRT and Svj
    Propose services for  $\mathcal{T}_j$  where Svj  $\notin$  Sc and Svj in IWKi - ReFiWki;
    with respect to Min (t) and Min (cost).
    If ( $\mathcal{T}_j \neq \emptyset$ )
        If (Sx1  $\prec_{mk}$  Svj) {
            Insert Svj into ReFiWki(t) and apply a TSP algorithm;
            /*ICAi will visit Svj to perform  $\mathcal{T}_j$  */
            DRT:= DRT -  $\mathcal{T}_j$ ; //update DRT }

```

The idea is to look for nodes which are not in the remained final Workplan of the agent ICA_i , belong to its initial Workplan and are still available ($\notin S_c$). If a node respect all these constraints and it succeeds the current position of the agent (S_{x1}) in the initial Workplan, then it is added to the remained Workplan.

Another problem takes place when the agent doesn't understand the message thus the agent may be fined this problem when he look for nodes in the initial Workplan. So we apply the preferences algorithm which uses the QOM algorithm to solve the ontology mapping problem:

```

For j := 1 to  $\alpha_i$  do
  Look for the set of tasks  $\mathcal{T}_j \subseteq DRT$  and  $S_{vj}$ 
  Propose services for  $\mathcal{T}_j$  where  $S_{vj} \notin S_c$  and  $S_{vj}$  in  $IWK_i - ReFiWk_i$ ;
  Apply QOM mapping algorithm;
  with respect to Min (t) and Min (cost).
  If ( $\mathcal{T}_j \neq \emptyset$ )
    If ( $S_{x1} \prec_{mk} S_{vj}$ ) {
      insert  $S_{vj}$  into  $ReFiWk_i(t)$  and apply
      a TSP algorithm;
      /* $ICA_i$  will visit  $S_{vj}$  to perform  $\mathcal{T}_j$  */
       $DRT := DRT - \mathcal{T}_j$ ; //update DRT }

```

C. Build the contract according to constraints

```

For j := 1 to J do
  Look for the set of tasks  $\mathcal{T}_j \subseteq DRT$  and  $S_j$ 
  Propose services for  $\mathcal{T}_j$  and  $S_j \notin S_c$  and  $S_j$  not in  $ReFiWk_i$ 
  and  $\exists k, 1 \leq k \leq \Delta_i$  where  $S_{xk}$  in  $ReFiWk_i(t)$  and  $Ls(S_{xk}, S_j) \approx 0$  and  $Ls(S_j, S_{x(k+1)}) \approx 0$ ;
  with respect to Min(t) and Min (cost).
  If ( $\mathcal{T}_j \neq \emptyset$ ) {
    Insert  $S_j$  into  $ReFiWk_i(t)$  just after  $S_{xk}$ ;
    /* $ICA_i$  will visit  $S_{xj}$  to perform  $\mathcal{T}_j$  */
     $DRT := DRT - \mathcal{T}_j$ ; //update DRT }

```

The idea is to look for any node which is not in the remained final Workplan and is still available ($\notin S_c$). If a node respects these constraints, we insert it in the remained final Workplan in an appropriate position.

```

For j := 1 to J do
  Look for the set of tasks  $\mathcal{T}_j \subseteq \text{DRT}$  and  $S_j$ 
  Propose services for  $\mathcal{T}_j$  and  $S_j \notin S_c$  and  $S_j$  not in  $\text{ReFiWk}_i$ 
  and  $\exists k, 1 \leq k \leq \Delta_i$  where  $S_{xk}$  in  $\text{ReFiWk}_i(t)$  and  $Ls(S_{xk}, S_j) \approx 0$  and  $Ls(S_j, S_{x(k+1)}) \approx 0$ ;
  Apply QOM mapping algorithm;
  with respect to  $\text{Min}(t)$  and  $\text{Min}(\text{cost})$ .
  If ( $\mathcal{T}_j \neq \emptyset$ ) {
    Insert  $S_j$  into  $\text{ReFiWk}_i(t)$  just after  $S_{xk}$ ;
    /*ICAi will visit  $S_{xj}$  to perform  $\mathcal{T}_j$  */
    DRT:= DRT-  $\mathcal{T}_j$ ; //update DRT }

```

The idea is to look for any node in the network which is not in the remained final Workplan and is still available and there are the possibility to do the ontology mapping because this node don't have the same entity correspondences to that sends by initiator according to its ontology.

D. Dispatch the contract over all existent active participants

SA agent never knows the exact position of a travelling ICA agent. However, he knows all initial Workplans schemes and the assignments of final servers (final effective Workplans). In addition, he has the crashed servers list, so he knows all services which need to be reassigned by providers. In that case, it must update the set of services which need to be reassigned by providers, proposing a new contract according to the updated services set. When SA agent creates a contract, it cannot send it directly to ICA agents because he never knows their exact position through the network. That's why; we choose to post the contract to all functioning servers. In that case, when an ICA agent visits a functioning server, it finds a waiting contract as an asynchronous message. We suppose that errors on the network are identified before an ICA agent leaves one functioning node toward a crashed one. Finally, to avoid a wrong message receiving, we delete all non received messages on all functioning servers before posting new ones.

E. Perform the contract and Update DRT table

When ICA agents receive the contract, they perform it according to their actual priorities, preferences, constraints and ontologies which can evolve depending on the time. When an ICA agent agree for his new Workplan, he executes. Otherwise, he confirms the accepted part of the contract, update the DRT table and change his remained Workplan. ICA agent must also send to SA agent what he was not agreed on, giving him a new proposition which must be compatible with his current new situation (see paragraphs A, B, C and D).

4.6. Summary

In this chapter, we proposed an optimizing approach of the data flow management, in order to satisfy, in a better manner, customers' requests. The adopted approach decreases considerably computing time because Workplans are just deduced; they are computed when network traffic varies considerably. We have presented a new solution for the problem of language interoperability between negotiation agents, by incorporating architecture for Negotiation process with that uses an Ontology-based Knowledge Management System (NOKMS). The proposed solution prevents the misunderstanding during the negotiation process through the agents' communications. The architecture consists of three layers: (NL, SEL and KMSL). In this work, we described, in details, the negotiation process as well as we illustrated the different messages types by using the different ontologies. Our proposed NOKMS improves the communications between heterogeneous negotiation mobile agents and the QoS in order to satisfy the transport customers. Indeed, the ICA agents can to ignore crashed nodes in their remained routes, so they have to avoid visiting them. This will be done by (DRT) algorithm for reassigning substitute servers tasks which need to be reassigned. This reassignment depends on the actual positions of ICA agents in their final Workplans. It depends also on their ontologies, constraints, priorities and preferences. The new assignment constitutes a contract between ICA agents and SA agents.

Chapter 5. Implementation and Results

5.1. Introduction

As we mentioned in the last chapter, our aim is the development of multi-agent system used a flexible ontology to support the transport passengers in planning their travels by modelling their requests. Some perturbations can occur during the mobile agents moving through the distant network nodes (bottleneck, failure, crash...) to search the services. So, our system supports the automatic negotiation during the perturbation case. The most important issue that must be addressed is the lack of understanding that may occur between agents due semantic differences in the representation of concepts.

In this Chapter, we present firstly the technical programming tools used in justifying our choices in section (5.2). We illustrate the implementation of our ISSAUM by using those tools (section 5.3). Then, we describe the results of implementation of our system in details through the algorithms presented in chapter 4. We present also the development of our travel ontology (section 5.4). Finally, the summary of this chapter will be in section 5.5.

5.2. Implementation Programming Tools

5.2.1. The Multi-Agent Platform: Java Agent Development Framework(JADE)

The need to implement systems with multiple autonomous components requires a software infrastructure, which used as environment for the deployment, and the implementation a set of agents. This infrastructure is called **platform development of multi-agent systems**. But, the implementation of such systems is often difficult at the level of handling the complex data structures, distribution, and the communication level.

In addition, artificial intelligence is a research-rich field and this richness leading to complexity and multiplicity of approaches proposed which in turn leads to numerous models of agents, environment, interactions and organizations. These models are often combined within one multi-agent system. Thus, it must to choose a multi-agent system adapted to the implementation requirements. Indeed, there are

several multi-agents platforms such as JADE²², MadKit²³, ZEUS²⁴, AgentBuilder²⁵, JACK²⁶, etc.

For the selection of the platform, we have leaved the unimportant criteria such as learning difficulties or the unavailability of sources. However, we have taken some important criteria in consideration:

- The possibility of implementing complex systems;
- The flexibility: we tried to avoid platforms that support a particular methodology;
- The acceleration of development through the existence of sufficient software components in order to produce accomplished applications;
- The distributed processing and in particular the existence of support for the paradigm of mobile agent MA;
- Finally, the possibility of integration of web services.

The two platforms that do not specify any methodology and can be regarded as "frameworks" are JADE and JACK, but JADE has several other interesting features such as the possibility of integrating web services and the existence of good support for content languages and ontologies

Thus, we chose the platform JADE, for the development of our system, the simulation results of our distributed optimization approaches, and the negotiation process (as we will see in next section). JADE is software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications (section 2.3.2). The communication architecture offers a flexible and an efficient messaging, where JADE creates and manages a queue of incoming ACL messages, private to each agent.

In fact, JADE is completely implemented in Java language, so, it supports the mobility. And, it is now one of the few multi-agent platforms that offer the possibility of integrating web services. On the other hand, JADE tries to facilitate the development of agent applications by optimizing the performance of a distributed agent system.

²² <http://jade.tilab.com>

²³ <http://www.madkit.net/>

²⁴ <http://www.cs.iastate.edu/~baojie/acad/current/zeus/zeus.htm>

²⁵ <http://www.agentbuilder.com/>

²⁶ <http://aosgrp.com/products/jack/index.html>

The agent platform can be distributed across machines that do not need to share the same operating system. Therefore, only one Java Virtual Machine (JVM), is executed on each host. Each JVM is basically a container of agents that provides a complete runtime environment for agent execution and allows several agents to run concurrently on the same host (Figure 5.1).

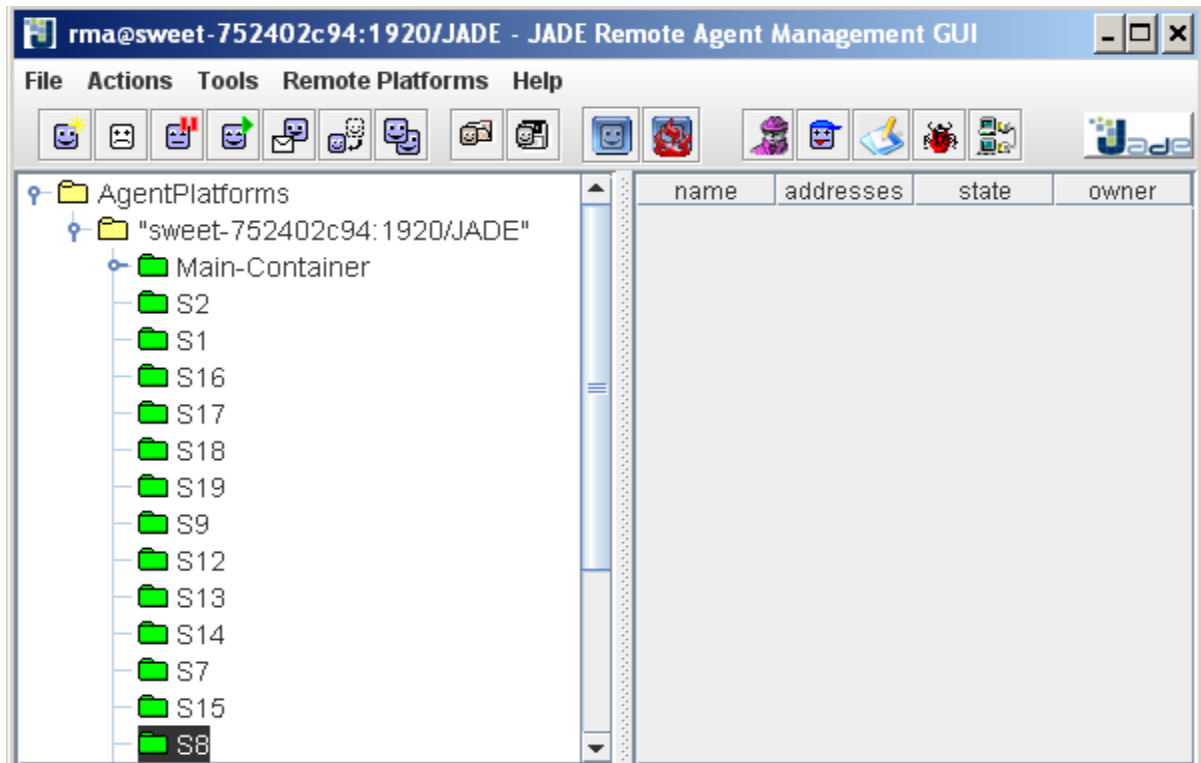


Figure 5.1. Snapshot of the RMA GUI

Finally, the concurrent tasks that a JADE agent performs are typically carried out as “behaviours”. Behaviours are created by extending the class *jade.core.behaviours.Behaviour*. To make an agent execute a certain task, an instance of the corresponding behaviour subclass has to be created and the *addBehaviour()* method of the *jade.core.Agent* class has to be called. One agent can implement and coordinate numerous behaviours in order to fulfil its goal.

5.2.2. Protégé and BeanGenerator

In this section, we present the protégé editor and his BeanGenerator plug-in which are the tools for the ontology development.

5.2.2.1. *Protégé*

Protégé²⁷ is a free, open source ontology editor and a knowledge acquisition system. It is a tool supporting the construction of ontologies and it also provides an application platform for knowledge based systems and libraries for application building. Protégé is being developed at Stanford University in collaboration with the University of Manchester (see section 2.2.5).

It is the best-known ontology editor with plug-ins that supports OWL and enables:

- Loading and saving OWL and RDF ontologies,
- Editing and visualizing OWL classes and their properties,
- Defining logical class characteristics as OWL expressions,
- Executing reasoner such as description logic classifiers,
- Editing OWL individuals for Semantic Web markup.

Protégé has flexible architecture and is easy to configure and extend. Protégé has an open-source Java API for the development of custom-tailored user interface components or arbitrary semantic web services. Protégé recently has over 100,000 registered users.

There are several other ontology editors such as OilEd, OntoEdit, etc (section 2.2.5). From our experience Protégé with its plug in architecture gives much wider possibilities. Protégé has many plug-ins and several are important for work done within the dissertation. For example, *OntoViz* plug-in was used for visualization of ontology to graphs. All ontology figures in the thesis are created using this plug-in. the result graphs are similar to UML diagrams. We use also, *Bean Generator* plug-in which we will explain it in the next section.

²⁷ <http://protege.stanford.edu>

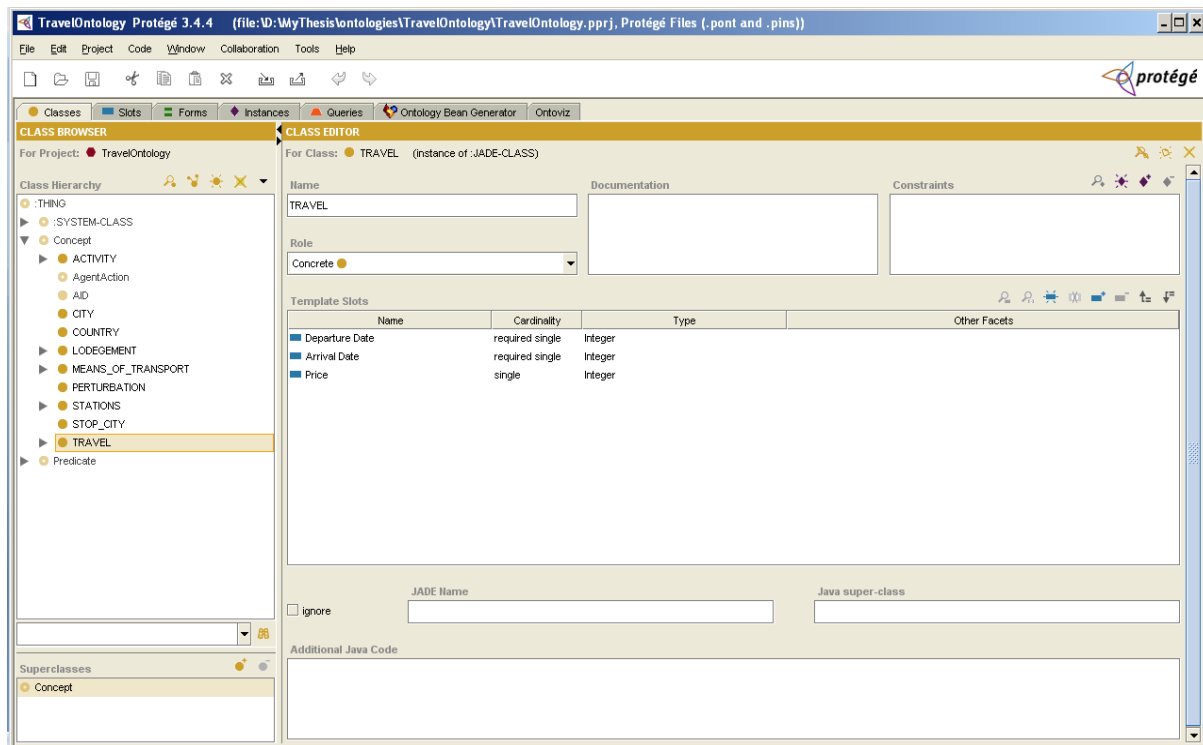


Figure 5.2. Protégé Interface

Figure 5.2 shows how the design of an ontology looks in Protégé-2000. In order to create Java files from this design, it is necessary to install the beanGenerator plug-in for Protégé-2000. The beanGenerator makes it possible to create a set of Java source files containing the ontology, which can directly be used in JADE.

5.2.2.2. *BeanGenerator*

The beanGenerator²⁸ is a plug-in for Protégé-2000 which generates a set of Java source files, containing the ontology which was designed in Protégé-2000.

²⁸ <http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator>

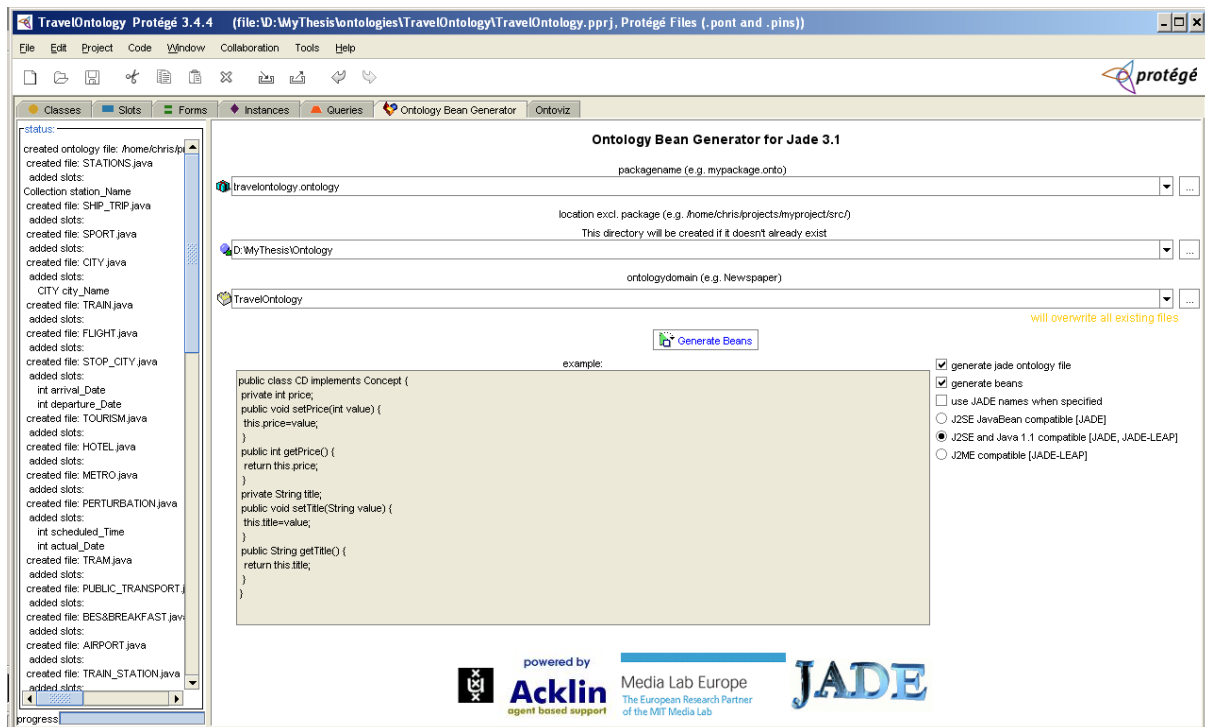


Figure 5.3. Bean Generator Interface

Figure 5.3 shows how by supplying a base directory, package name and ontology name, all source files can be easily generated by pressing the “Bean Generator” button. The generated beans can be either, JavaBean, J2SE or J2ME compatible.

5.3. The Implementation of ISSAMU System

5.3.1. Ontology Validation And Generate Java Classes

As we have described in the chapter 3. For developing an ontology, there are 4 steps: the definition of the ontology purpose, conceptualization and formalization. We have illustrated those steps in details. Final step is the validation which we explain here.

To validate the ontology, we create several instances based on real examples of travel. During ontology instantiation, we verified that all concepts were used and all the need information required to support the travel planning were represented, (Figure 5.4).

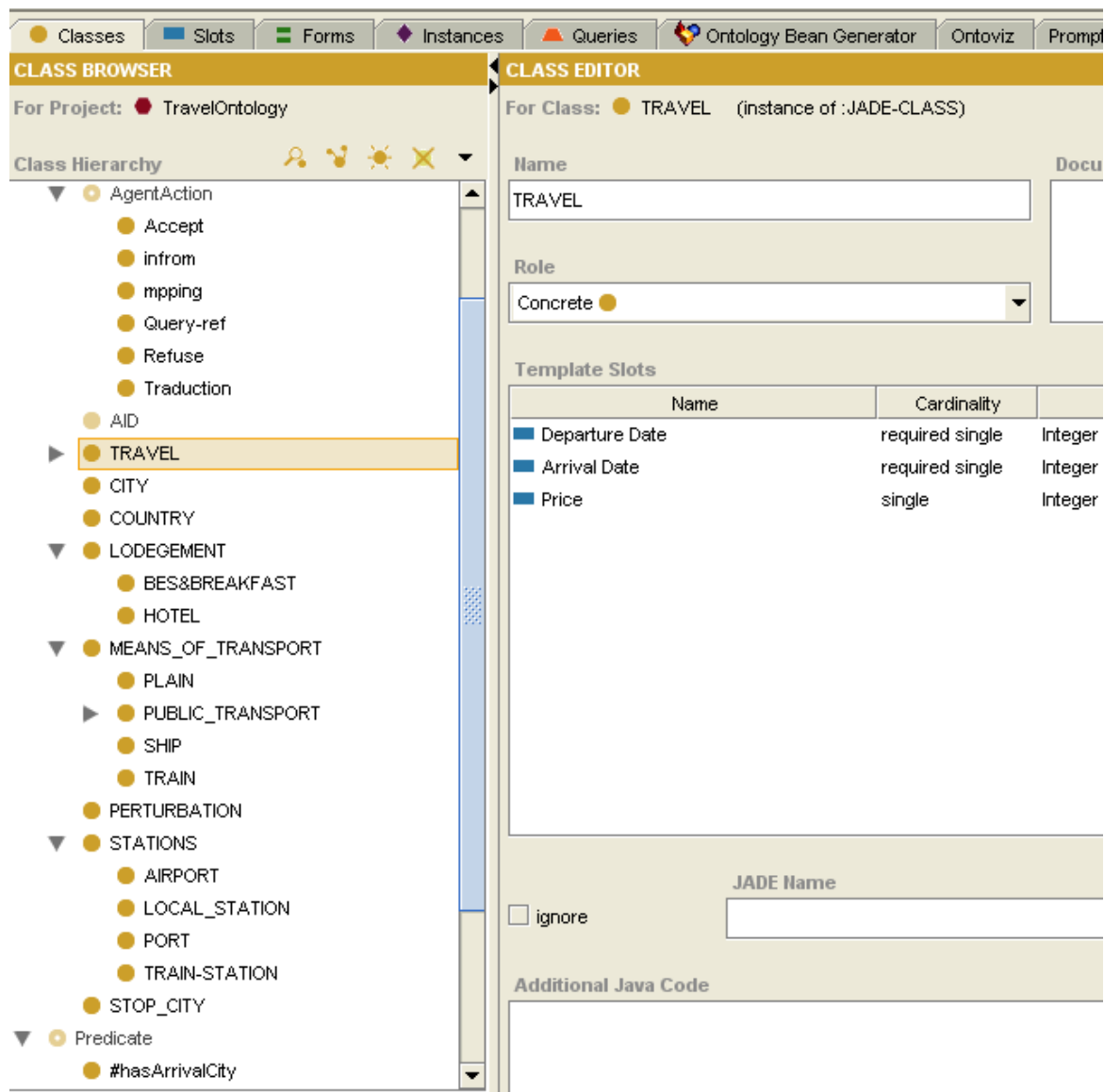


Figure 5.4. Ontology Hierarchy in Protégé

To Generate the Java classes for our travel ontology, we use the BeanGenerator which associates each schema in our ontology with a Java class or interface. The sum of all created objects that implement the interfaces `jade.content.Concept`, `jade.content.Predicate` or `jade.content.AgentAction` represent the ontology. Furthermore, the common class which defines the vocabulary of all classes, registering (Predicates, Concepts and AgentActions), and storing name mappings, etc., is created.

The following code is generated by the BeanGenerator tool. It is the correspondent to Java class for the Concept “Tavel”, as follow (figure 5.5):


```
import jade.content.*;
import jade.content.Concept;
import jade.util.leap.*;
import jade.core.*;
/**
 * Protege name: TRAVEL
 * @author ontology bean generator
 * @version 2010/05/30, 16:42:06
 */
public class TRAVEL implements Concept {
    /**
     * Protege name: arrivale-date
     */
    private String arrivale_date;
    public void setArrivale_date(String value) { this.arrivale_date=value;}
    public String getArrivale_date() { return this.arrivale_date; }

    /**
     * Protege name: depature-date
     */
    private String depature_date;
    public void setDepature_date(String value) { this.depature_date=value;}
    public String getDepature_date() {return this.depature_date;}
}
```

Figure 5.5. Example of BeanGerator Class

5.3.2. The Implementation of The our Agents by JADE

To see the communications and behaviours of agents, JADE provides graphical tools, which are agent too, such as «RMA» (**R**emote **M**anagement **A**gent). RMA represents the main interface of monitor and control the platform and all its remote containers. It remotes management of the life-cycle of agents (creating, suspending, resuming, killing, etc...), as in (figure 5.1). We can use RMA to launch the other graphical tools, like Sniffer. Sniffer displays the flow of interactions between selected agents and it displays the content of each exchanged message. Figure (5.6), represents the communications between our ISSAUM agents.

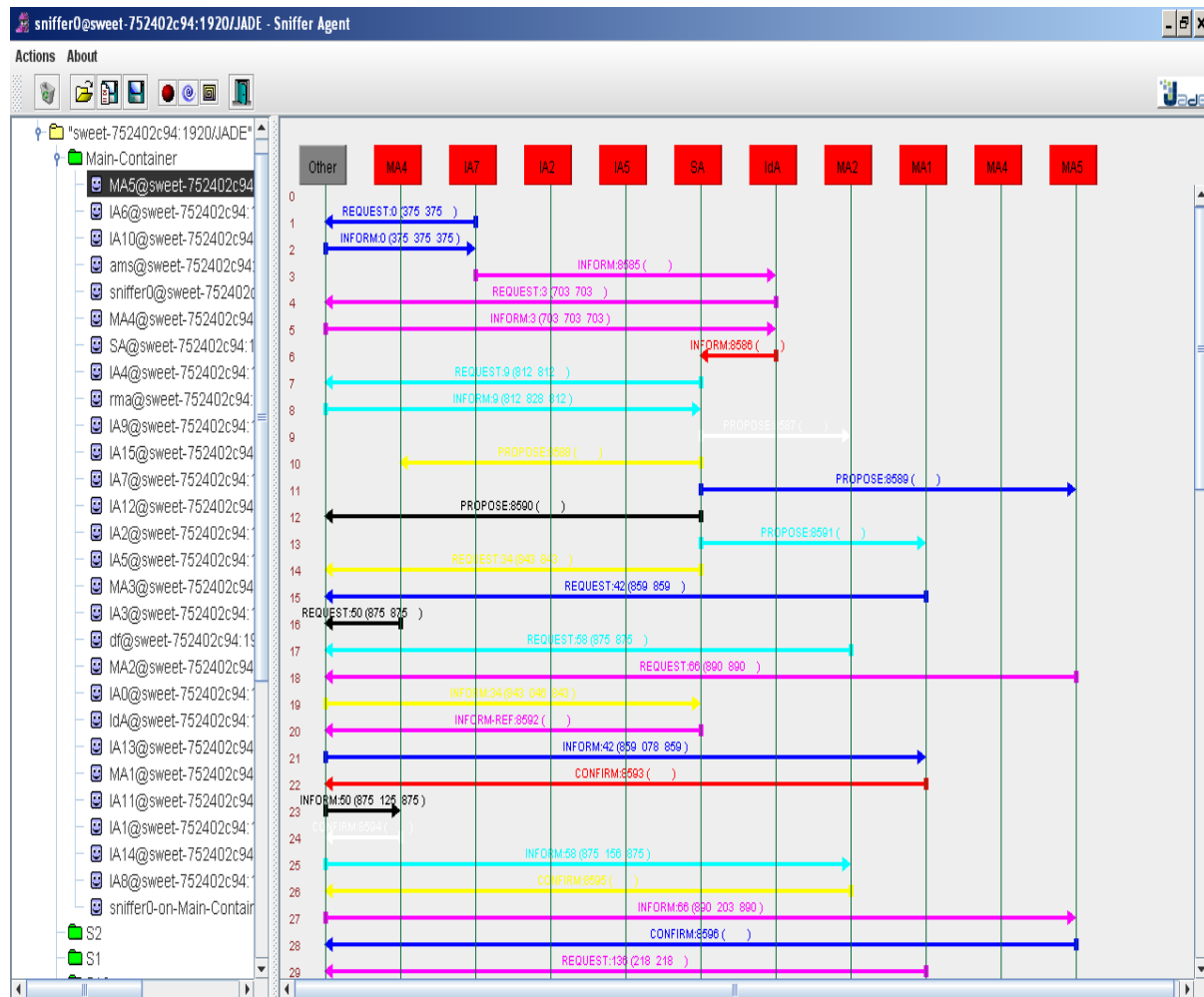


Figure 5.6. The agents Communication in ISSAUM

We have implemented a graphical java interface to show our general simulation (Figure 5.8). This interface allows us to retrieve the results for a selected chromosome. The figure (5.7) illustrates an example of result which response to set of 2 simultaneous user's requests. The requests decomposed into 2 services according to our travel ontology. Each request (req_i) was captured by an IA_i agent ($1 \leq i \leq 5$). And each identified task was served by an agent ICA agent who chosen by the SA agent. For example: the ICA_2 agent, (MA_2) in the interface, goes to search the service T_{14} in the node S_{10} for collect the information $S_{10}T_{14}$. The results here show that the selected chromosome respects the maximum response time of each request "due date".

The data, which we used in our system, are available in the appendix A. The simulation results presented in this chapter is based on the ETMN. The later consists of 20 nodes which propose 100 services. The ETMN can fully explore by 5 ICA agents, as we will see in the rest of this chapter.

ICA	Service	Data	Source
MA2	T 14	S10T14	S10
MA5	T 5	S11T5	S11

Request	TotalCost	TotalTim	Duedate
IA3	29	1.7	2
IA6	32	1.1	2

Number of ICA : 2

Number of Servers : 2

New (nbRequests, nbTasks) = (2 , 2)

Figure 5.7. The Results Interface

5.4. The Results of Simulation

5.4.1. Services Modeling by The Flexible Ontology and Decomposition of Requests

In this section, we illustrate a real case study applied on our ISSAUM project. Transport users require relevant, interactive and instantaneous information during their travels. Hence, ISSAUM offers a support tool to response to their demands.

Let us suppose that at the instant $t=11.00$ and during 2 seconds ($\Delta\epsilon = 2s$), we assume the existence of a number of users connected to our system who formulate a number of simultaneous queries. Where, (A, B, C and D) are 4 cities in different countries (for example: A= Lille, B=Paris, C= London, D= Berlin), as follow, (Figure 5.8.):

- Query1: travel at the instant t from B to C by train without stop;
- Query2: travel at the instant t from A to C by (train , airplane) with /without stop;
- Query3: Ask about the perturbations of transport circulate (the public transport too) between B and D (today/this week);
- Query4: Look for the best activities in corresponds to the travel by airplane X, from B (today at 12.00/ the next week) to go to C ;
- Query5: travel at the instant t from A to D with best price;
- Query6: Looking for a hotel of a good (Quality /Cost) in D during the next weekend and make a reservation;
- Query7: find the different activities in C for winter 2011;
- Etc.

Each IA sends the simultaneous requests $\Delta\epsilon$ to the IdA of the same agent society (section 4.2.2). The later decomposes those requests according to our travel ontology into $I'=64$ tasks different, and it should remark that there is no direct way between A and C, or between A and D.

Thus, we can assume that the IdA agent decomposes the queries into a set of independents tasks, as follows, (Figure 5.9):

- T_1 : "Look for a hotel of a good (Quality / Cost) in D during next weekend, and make a reservation";
- T_3 : "Perturbations of traffic between B and D (today / this week)";
- T_6 : "Find the shortest way to go from A to C at the instant $t=11.00$ ";
- T_9 : "Looking for the best travel time from B to C according to the forecast of road traffic for the next week ";
- T_{13} : "Ask about cultural events in C (next week)";
- T_{16} : "travel from B to C today (at the instant $t=11.00$ or starting from 12.00)";
- T_{19} : " Looking for a travel by airplane X from B to C (today at the instant $t=11.00$ or the next week-end) with the best choice of the activities (tourism, cultural, etc...) which are related to this travel";
- Etc.

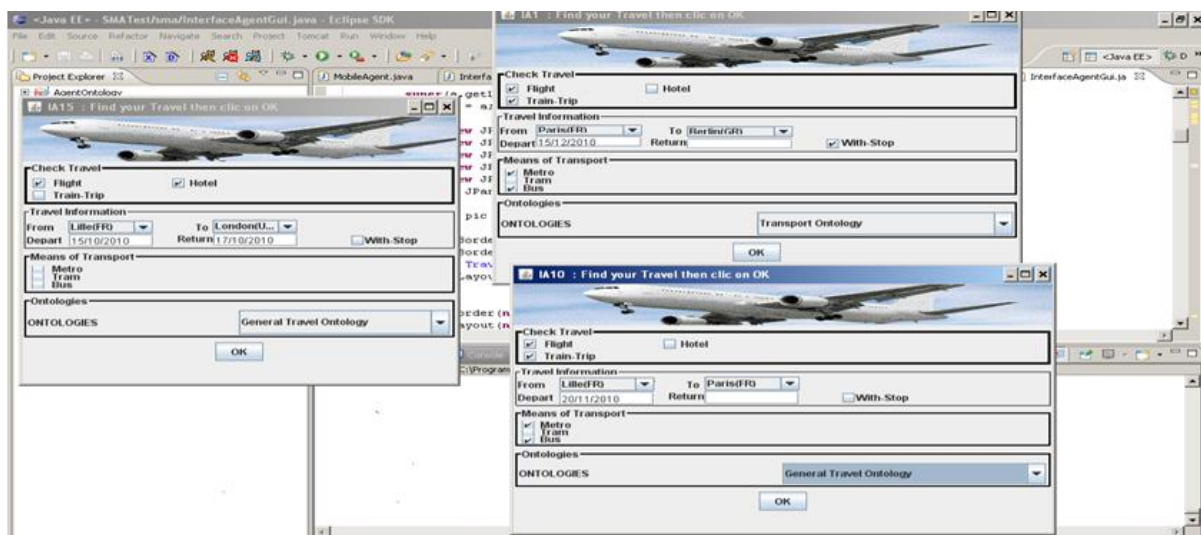


Figure 5.8. The User's Interfaces

According to our Travel Ontology, we find that those tasks correspond to the following concepts, Table 5.1:

Tasks	Associated Concepts	Services Labels
T_1	CITY, COUNTRY, LOGEMENT, HOTEL, Min(Price), Max(Stars), DepatureDate= next week-end	Travel Service (Lodgement)
	TRAVEL, CITY, COUNTRY, LOGEMENT, Min(Price), Max(Stars), DepatureDate= next week-end	Travel Service
T_3	PERTURBATION, MEANS-OF-TRANSPORT, TRAIN, CITY, COUNTRY, Actual Time = today at instant $t=11.00$	Transport Service
	PERTURBATION, MEANS-OF-TRANSPORT, AIRPLANE,CITY, COUNTRY, Actual Time = today at instant $t=11.00$	Transport Services
	PERTURBATION, MEANS-OF-TRANSPORT, PUBLIC-TRANSPOR, BUS, TRAM, METRO, CITY, COUNTRY, Actual Time = today at instant $t=11.00$	Transport Services
	PERTURBATION, MEANS-OF-TRANSPORT, TRAIN, CITY, COUNTRY, Scheduled Time= this week	Transport Service
	PERTURBATION, MEANS-OF-TRANSPORT, AIRPLANE, CITY, COUNTRY, Scheduled Time= this week	Transport Service
	PERTURBATION, MEANS-OF-TRANSPORT, PUBLIC-TRANSPOR, BUS, TRAM, METRO, CITY, COUNTRY, Scheduled Time= this week	Transport Service
T_6	TRAVEL, CITY, COUNTRY, TRAIN, DepatureDate= today at instant $t=11.00$	Travel Service
	TRAVEL, CITY, COUNTRY, TRAIN, STOP- CITY, DepatureDate= today at instant $t=11.00$	Travel Service
	TRAVEL, CITY, COUNTRY, AIRPLAN, DepatureDate = today at instant $t=11.00$	Travel Service
	TRAVEL, CITY, COUNTRY, AIRPLANE, STOP- CITY , DepatureDate = today at instant $t=11.00$	Travel Service
	TRAVEL, CITY, COUNTRY, AIRPLANE, STOP- CITY, DepatureDate= today at instant $t=11.00$	Travel Service
	TRAIN, CITY, COUNTRY, TRAIN-STATION, DepatureDate= today at instant $t=11.00$	Transport Service
	AIRPLANE, CITY, COUNTRY, AIRPORT ,	Transport Service

	DepatureDate= today at instant $t = 11.00$	
	TRAIN, CITY, COUNTRY, TRAIN-STATION	Geographic Service
	AIRPLANE, CITY, COUNTRY, AIRPORT	Geographic Service
T_9	TRAVEL, CITY, COUNTRY, TRAIN, PERTURBATION, MEANS-OF-TRANSPORT, Scheduled Time= next week	Travel Service
	TRAVEL, CITY, COUNTRY, AIRPLANE, PERTURBATION, MEANS-OF-TRANSPORT, Scheduled Time = next week	Travel Service
	PERTURBATION, MEANS-OF-TRANSPORT, TRAIN, CITY, COUNTRY, Scheduled Time = next week	Transport Service
	PERTURBATION, MEANS-OF-TRANSPORT, AIRPLANE, CITY, COUNTRY, Scheduled Time = next week	Transport Service
	PERTURBATION, MEANS-OF-TRANSPORT, PUBLIC-TRANSPOR, BUS, TRAM, METRO, CITY, COUNTRY, Scheduled Time = next week	Transport Service
	CITY, COUNTRY, AIROPORT, PERTURBATION, Scheduled Time = next week	Geographic Service
	CITY, COUNTRY, TRAIN-STATION, PERTURBATION, Scheduled Time = next week	Geographic Service
	CITY, COUNTRY, LOCAL-STATION, PERTURBATION, Scheduled Time = next week	Geographic Service
T_{13}	TRAVEL, CITY, COUNTRY, ACTIVITY, CULTURE , Activity Date=next week.	Travel Service (Activity)
	CITY, COUNTRY, ACTIVITY, CULTURE, Activity Date=next week.	Geographic Service
T_{16}	TRAVEL, TRAIN, CITY, COUNTRY, DepatureDate= today at $t=11.00$	Travel Service
	TRAVEL, TRAIN, CITY, COUNTRY, DepatureDate= today at $t>12.00$	Travel Service
	TRAIN, CITY, COUNTRY, DepatureDate= today at $t=11.00$	Transport Service
	TRAIN, CITY, COUNTRY, DepatureDate= today at $t>12.00$	Transport Service
	TRAVEL, AIRPLANE, CITY, COUNTRY, DepatureDate= today at $t=11.00$	Travel Service
	TRAVEL, AIRPLANE, CITY, COUNTRY,	Travel Service

	DepatureDate= today at $t > 12.00$	
	AIRPLANE, CITY, COUNTRY, DepatureDate= today at $t = 11.00$	Transport Service
	AIRPLANE, CITY, COUNTRY, DepatureDate= today at $t > 12.00$	Transport Service
T ₁₉	TRAVEL, AIRPLANE, CITY, COUNTRY, DepatureDate= today at $t = 12.00$, Min(Price)	Travel Service
	TRAVEL, AIRPLANE, CITY, COUNTRY, DepatureDate= next week, Min(Price)	Travel Service
	TRAVEL, AIRPLANE, CITY, COUNTRY, ACTIVITY, SOPRT, ActivityDate=today.	Travel Service (Activity)
	TRAVEL, AIRPLANE, CITY, COUNTRY, ACTIVITY, TOURISEM, ActivityDate= today	Travel Service (Activity)
	TRAVEL, AIRPLANE, CITY, COUNTRY, ACTIVITY, CULTURE, ActivityDate= today	Travel Service (Activity)
	TRAVEL, AIRPLANE, CITY, COUNTRY, ACTIVITY, CULTURE, ActivityDate=next-week	Transport Service (Activity)
	TRAVEL, AIRPLANE, CITY, COUNTRY, ACTIVITY, CULTURE, ActivityDate=next-week.	Transport Service (Activity)
	TRAVEL, AIRPLANE, CITY, COUNTRY, ACTIVITY, TOURISEM, ActivityDate= next-week.	Transport Service (Activity)
.....

Table 5.1. Table of Associated Concepts

It should be observer that a task can be represented by several services with different constraints; for example: the task T₁₉ can be representing by 8 different services which correspond to the same task. "Travel from A to C by using airplane" where this travel has different constraint, either: "today with best price", or "the next week-end with best price". Other constraints which related to this travel: "looking for different related activities services (culture, tourism, etc) for today or for the next week-end".

Finally, the responses will be recomposed by FA agent who must merge the services according to the user's constraints, taking into account the relevance information. The table 5.2 represent the decomposition of requests by IdA agent into different tasks (services).

	req ₁	req ₂	req ₃	req ₄	req ₅	req ₆	req ₇	req ₈	req ₉	req ₁₀
T ₁	x									
T ₂										x
T ₃	x		x					x		
T ₆	x		x					x	x	
T ₉	x								x	
T ₁₃	x		x				x		x	
T ₁₆	x		x						x	x
T ₁₉	x		x						x	
T ₂₀	x				x		x		x	
T ₂₁	x				x		x		x	
T ₂₂	x				x		x		x	
T ₂₅	x		x		x		x			x
T ₂₆	x		x				x	x		x
T ₂₈	x		x				x	x		x
T ₂₉			x		x			x		x
T ₃₀	x				x			x		
T ₃₁	x		x		x					x
T ₃₂	x		x		x					
T ₃₃	x				x					
T ₃₄	x				x					x
T ₃₅	x		x		x		x		x	
T ₃₆	x		x		x		x			
T ₃₇	x				x		x			
T ₃₈	x						x		x	
T ₃₉	x						x			
T ₄₀				x	x	x				x
T ₄₁					x					
T ₄₂					x		x			
T ₄₄							x		x	
T ₅₂							x		x	
T ₅₃					x		x			
T ₅₆			x		x		x		x	
T ₅₇			x		x		x	x		
T ₅₈	x				x		x	x		
T ₅₉	x				x					
T ₆₀	x		x		x					
T ₆₁		x			x		x			
T ₆₃		x					x			
T ₆₄		x					x			x
T ₆₅	x				x		x			x
T ₆₆					x				x	x
T ₆₇	x				x	x	x		x	
T ₆₈			x		x	x			x	
T ₆₉	x		x		x		x	x		

T ₇₁	x				x		x	x		x
T ₇₃	x						x			
T ₇₄	x		x				x			
T ₇₅	x	x	x		x		x			x
T ₇₆	x	x			x		x			
T ₇₇	x				x		x	x		
T ₇₈	x		x		x		x	x	x	
T ₇₉	x	x	x		x					
T ₈₀										x
T ₈₁	x	x	x						x	
T ₈₂										
T ₈₃										
T ₈₄					x					
T ₈₅					x					
T ₈₆							x			x
T ₈₈					x					x
T ₉₀										x
T ₉₅	x						x			x
T ₉₆	x									
T ₉₉	x	x	x						x	

Table5.2. Decomposition of Requests by IdA agents

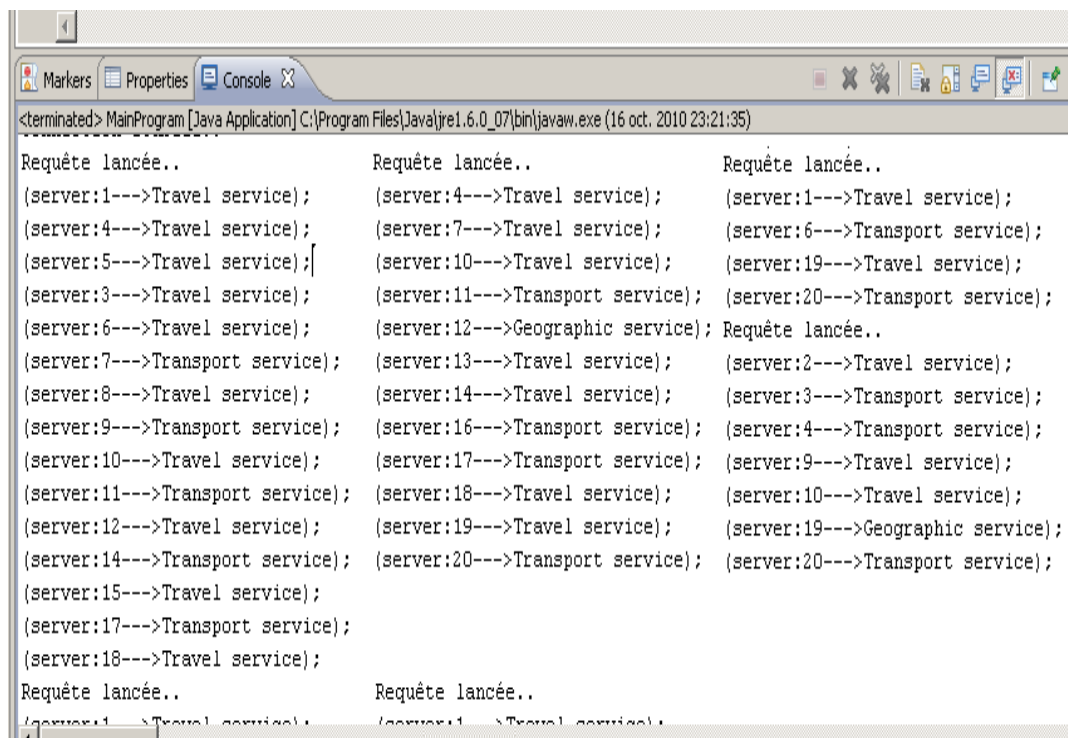


Figure 5.9. The Result of Services Decompositions

5.4.2. The Optimize Solution by Scheduler Agent (SA) to Find the Work Plans

As we motioned in section (4.2.3), according to the optimize process which applies by the SA agents. The first level of the optimization helps to find the IWps for the ICAs to explore the entire EMTN. The results presented in this chapter is based on EMTN which consists of 20 nodes ($J=20$). Those servers propose 100 different services ($I=100$). The EMTN can be fully explored by 5 ICA agents.

Then, for $m=5$, S_i : transport information system provider where $1 \leq i \leq n$. Then, IWps for ICAs, (Table 5.3), are:

- $IWp_1 = (S_{20}, S_{15}, S_1, S_3) ;$
- $IWp_2 = (S_{18}, S_7, S_{10}, S_{17}) ;$
- $IWp_3 = (S_2, S_{13}, S_{19}, S_6) ;$
- $IWp_4 = (S_{16}, S_{14}, S_5, S_{12}, S_4) ;$
- $IWp_5 = (S_{11}, S_8, S_9) ;$

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	S_{20}
ICA_1	x		x												x					x
ICA_2							x			x							x	x		
ICA_3		x				x							x						x	
ICA_4				x	x							x		x		x				
ICA_5								x	x		x									

Table 5.3. The IWps of ICAs

The SA agent generates the corresponding FeTAR solution which affects the servers to the various required tasks. So, for a set of the requests demanded by users at the instant t , the chromosome CH (table 5.4) is selected to deduce the FWps of the ICAs from the IWps, as follow where $m'=5$, (Figur 5.10):

- $FWp_{t,1} = \{S_{20}\{T_9, T_{37}, T_{39}\}, S_{15}\{T_{28}, T_{58}\}, S_1\{T_{19}, T_{29}, T_{66}, T_{88}\}, S_3\{T_3, T_{26}, T_{32}, T_{33}, T_{38}, T_{42}, T_{61}, T_{85}\} \} ;$
- $FWp_{t,2} = \{ S_{18}\{T_1, T_{13}, T_{30}, T_{36}, T_{41}, T_{65}, T_{76}, T_{77}\}, S_7\{T_{34}\}, S_{17}\{T_{25}, T_{44}, T_{60}, T_{80}\} \} ;$
- $FWp_{t,3} = \{ S_2\{T_{59}, T_{78}, T_{79}, T_{84}\}, S_{13}\{T_{53}\}, S_{19}\{T_6, T_{16}, T_{22}, T_{52}, T_{57}, T_{67}, T_{96}\}, S_6\{T_{68}\} \} ;$
- $FWp_{t,4} = \{ S_{16}\{T_{63}, T_{73}, T_{74}, T_{90}\}, S_{14}\{T_{71}\}, S_5\{T_{20}, T_{86}, T_{95}, T_{99}\}, S_{12}\{T_{75}, T_{83}\}, S_4\{T_{21}, T_{64}, T_{81}\} ;$
- $FWp_{t,5} = \{ S_{11}\{T_{40}, T_{56}, T_{69}\}, S_8\{T_{31}, T_{35}\}, S_9\{T_2, T_{82}\} \} .$

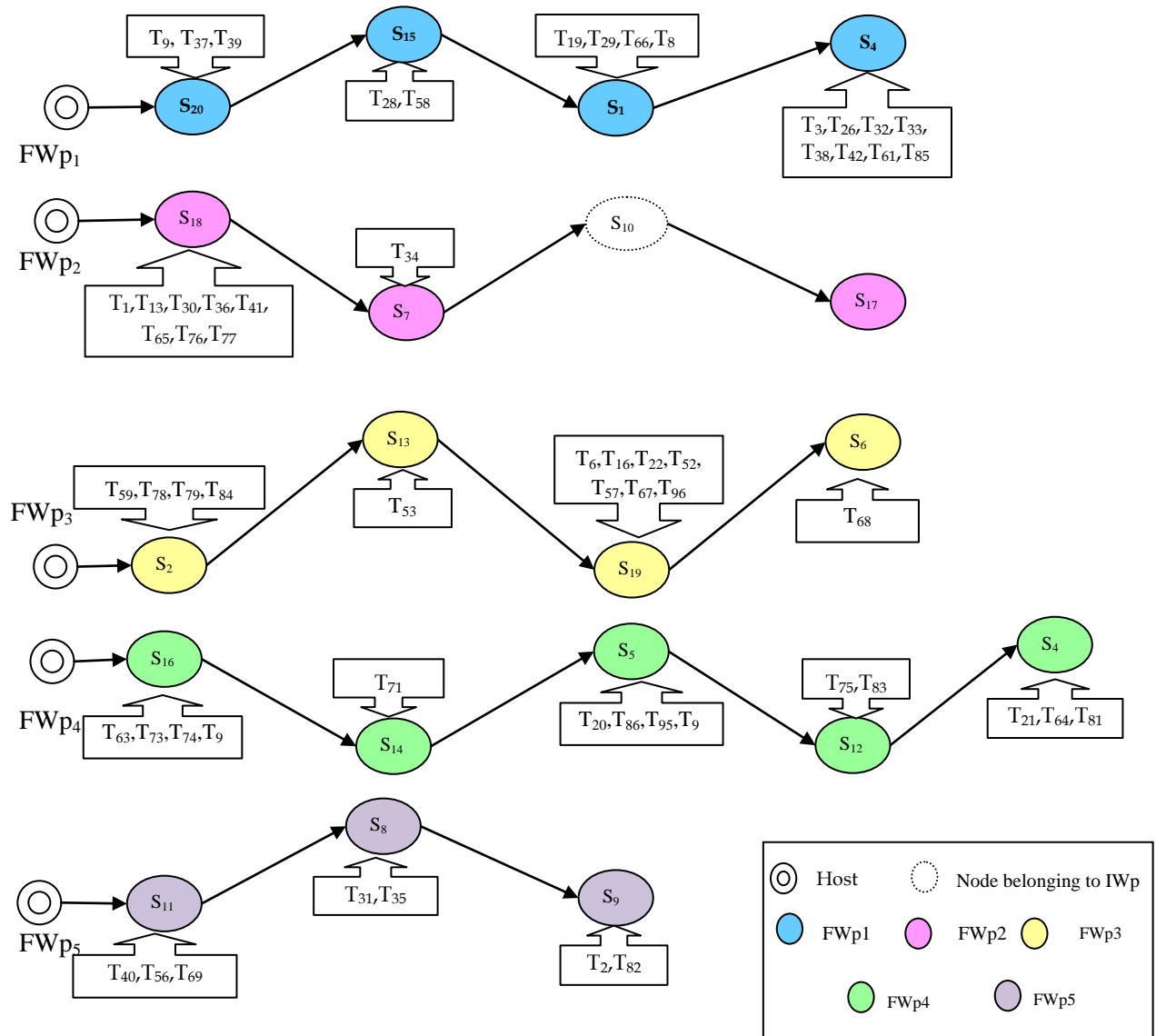


Figure 5.10. The FWps of the ICA agents with the Assigned Tasks

CH	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
T ₁	*	X	*	*	*	*	*	*	*	*	*	*	X	*	*	X	*	1	X	X
T ₂	*	X	*	*	X	X	*	*	1	*	X	*	*	*	*	X	*	*	*	*
T ₃	*	X	1	*	*	X	*	*	*	*	*	*	X	*	*	*	X	*	*	*
T ₆	*	X	*	*	*	X	*	*	*	*	*	*	X	X	*	*	*	*	1	*
T ₉	X	X	X	*	X	X	X	*	*	*	*	*	*	X	*	X	*	X	X	1
T ₁₃	*	X	*	*	*	X	X	*	*	*	*	*	X	X	*	*	X	1	*	X
T ₁₆	X	X	X	X	*	X	*	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₁₉	1	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₂₀	X	X	*	X	1	X	X	X	X	*	X	X	X	X	*	X	X	X	X	*
T ₂₁	X	X	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₂₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1	*
T ₂₅	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X	*	1	X	X	*
T ₂₆	X	*	1	X	X	X	X	X	X	X	*	*	X	X	X	X	X	*	X	X
T ₂₈	X	*	*	X	*	X	X	X	X	*	X	*	X	X	1	X	X	X	*	*

T ₂₉	1	*	X	X	X	X	X	X	X	X	*	X	*	X	X	X	X	X	X	X
T ₃₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	*	*
T ₃₁	*	X	*	*	X	*	X	1	X	X	X	*	X	*	X	X	X	*	X	X
T ₃₂	*	X	1	*	X	X	*	*	*	X	X	*	X	*	X	X	X	*	X	X
T ₃₃	*	X	1	*	X	X	*	X	X	X	X	*	X	X	X	*	X	X	X	X
T ₃₄	*	*	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₃₅	*	X	*	X	X	X	X	1	X	X	X	X	X	X	X	X	X	*	X	X
T ₃₆	*	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₃₇	X	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1
T ₃₈	*	X	1	X	X	X	X	X	X	*	X	X	X	*	X	X	*	X	X	X
T ₃₉	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	X	1
T ₄₀	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	*	*	X	X	X
T ₄₁	X	X	*	X	*	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₄₂	*	X	1	X	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	X
T ₄₄	X	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X	1	X	X	X
T ₅₂	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	1	X
T ₅₃	*	X	X	X	X	X	X	X	X	X	X	X	1	X	X	*	X	X	X	X
T ₅₆	X	*	*	X	X	X	X	X	X	X	1	X	X	X	X	X	X	*	X	X
T ₅₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	X
T ₅₈	X	*	*	X	X	X	X	X	X	*	X	X	X	X	1	X	X	X	*	X
T ₅₉	*	1	X	X	X	X	X	X	X	X	*	X	*	X	X	X	X	X	X	X
T ₆₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	*	*	X
T ₆₁	X	X	1	X	X	X	X	X	X	X	X	X	X	X	*	X	*	X	X	X
T ₆₃	*	X	*	*	*	X	*	*	*	X	X	X	X	X	X	1	X	*	X	X
T ₆₄	*	*	*	1	*	X	X	X	*	X	X	X	X	X	X	X	X	X	X	X
T ₆₅	*	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X	X	1	X	X
T ₆₆	1	X	*	X	X	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X
T ₆₇	X	X	*	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X	1	X
T ₆₈	*	X	*	X	X	1	X	*	X	X	X	X	X	X	X	X	*	X	X	X
T ₆₉	X	X	X	X	X	X	X	X	*	X	1	X	X	X	X	X	X	X	X	X
T ₇₁	X	X	*	X	X	X	*	X	X	X	X	X	X	1	X	X	X	X	X	X
T ₇₃	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	1	X	X	*	X
T ₇₄	X	X	X	X	X	X	*	X	X	X	X	X	X	*	X	1	X	X	X	X
T ₇₅	*	X	*	X	X	X	X	X	X	*	X	1	X	X	X	X	X	X	X	X
T ₇₆	X	*	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X
T ₇₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	*
T ₇₈	X	1	X	X	X	X	X	X	X	X	X	*	X	X	X	X	X	X	X	X
T ₇₉	*	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	*
T ₈₁	*	X	*	1	X	X	X	X	X	X	X	*	X	X	X	X	X	*	X	X
T ₈₂	*	X	*	X	X	X	*	*	1	X	X	X	X	*	X	X	X	X	X	X
T ₈₃	X	X	X	X	*	X	*	X	X	X	X	1	X	*	X	X	X	X	X	X
T ₈₄	*	1	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₈₅	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₆	*	X	*	X	1	X	X	X	X	X	X	X	X	X	X	X	X	*	X	X
T ₈₈	1	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
T ₉₀	X	X	X	X	X	X	X	X	X	*	X	X	X	X	*	X	1	*	X	X
T ₉₅	X	X	X	X	1	X	X	X	*	X	*	X	*	X	X	X	X	X	X	X
T ₉₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X
T ₉₉	*	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 5.4. Chromosome CH (generated FeTAR instance)

The FWps of ICA agents, which are deduced from IWps by, represent in the table 5.4 (read x), where we have 5 mobile agents (ICAs). Those agents will visit 19 servers (nodes) to collect 54 services.(Table 5.5)

	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅	S ₁₆	S ₁₇	S ₁₈	S ₁₉	S ₂₀
ACI ₁	x		x												x					x
ACI ₂							x			x							x	x		
ACI ₃		x				x							x						x	
ACI ₄				x	x							x				x				
ACI ₅								x	x		x									

Table 5.5. The FWps of ICAs

The figure (5.11) represents the simulation result of our example.

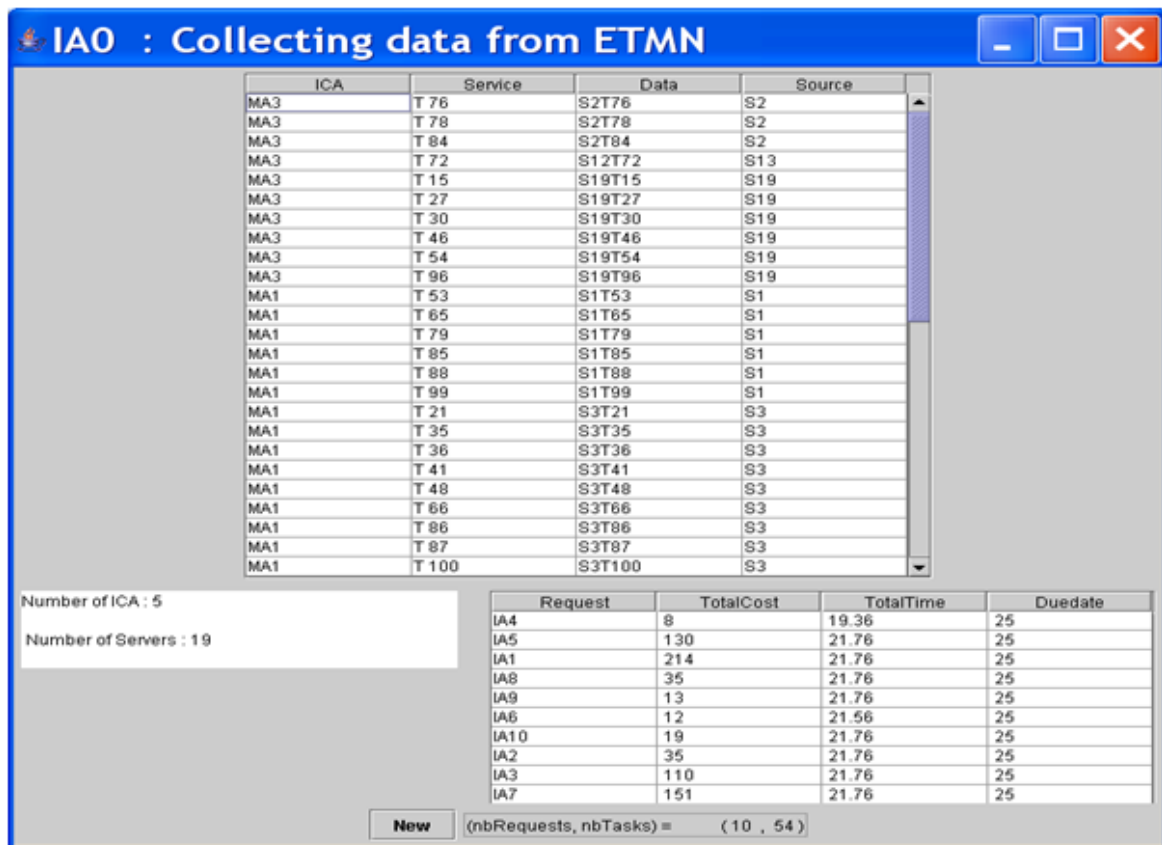


Figure.5.11. The Simulation Result of Our Example

5.4.3. The Contribution of the Dynamic Data Archiving Model (DDAM)in ISSAUM

As we mentioned in the section (3.4), each information provider offers a service, it must include both the value of its time indicator (IndT) and the value of updating indicator (IndU). So, (Table 5.6) shows the indicators of services, which relate to the

services of our example. Indeed, by using those indicators, may be, we can reduce the number of ICA agents to 3 agents visit 10 nodes to collect 37 tasks (services). The FWps of ICAs agents reduced as follow:

- $FWp_{t,1} = \{ S_{20}\{T_9, T_{37}, T_{39}\}, S_{15}\{T_{28}, T_{58}\}, S_1\{T_{19}, T_{29}, T_{66}, T_{88}\}, S_3\{T_3, T_{26}, T_{32}, T_{33}, T_{38}, T_{42}, T_{61}, T_{85}\} \}$
- $FWp_{t,2} = \{ S_{18}\{T_1, T_{13}, T_{30}, T_{36}, T_{41}, T_{65}, T_{76}, T_{77}\}, S_7\{T_{34}\}, S_{17}\{T_{25}, T_{44}, T_{60}, T_{80}\} \};$
- $FWp_{t,3} = \emptyset;$
- $FWp_{t,4} = \emptyset;$
- $FWp_{t,5} = \{ S_{11}\{T_{40}, T_{56}, T_{69}\}, S_8\{T_{31}, T_{35}\}, S_9\{T_2, T_{82}\} \}.$

Tasks	IndT	IndU
T_1	1	$12h \leq x = 24h$
T_2	1	$12h \leq x = 24h$
T_3	3	-
T_6	0	-
T_9	2	$60day > x = 24h$
T_{13}	0	-
T_{16}	0	-
T_{19}	3	-
T_{20}	0	-
T_{21}	0	-
T_{22}	0	-
T_{25}	3	-
T_{26}	2	$30day > x = 24h$
T_{28}	1	$3h \leq x = 24h$
T_{29}	3	-
T_{30}	1	$5h \leq x = 24h$
T_{31}	2	$10day > x = 24h$
T_{32}	2	$20day > x = 24h$
T_{33}	3	-
T_{34}	1	$10h \leq x = 24h$
T_{35}	1	$8h \leq x = 24h$
T_{36}	2	$30day > x = 24h$
T_{37}	3	-
T_{38}	1	$8h \leq x = 24h$
T_{39}	2	$7day > x = 24h$
T_{40}	2	$60day > x = 24h$
T_{41}	3	-
T_{42}	2	$10day > x = 24h$
T_{44}	1	$6h \leq x = 24h$
T_{52}	0	-
T_{53}	0	-

T_{56}	3	-
T_{57}	0	-
T_{58}	1	$6h \leq x = 24h$
T_{59}	0	-
T_{60}	1	$8h \leq x = 24h$
T_{61}	1	$10h \leq x = 24h$
T_{62}	2	$20day > x = 24h$
T_{63}	0	-
T_{64}	0	-
T_{65}	3	-
T_{66}	2	$7day > x = 24h$
T_{67}	0	-
T_{68}	0	-
T_{69}	3	-
T_{71}	0	-
T_{73}	0	-
T_{74}	0	-
T_{75}	0	-
T_{76}	3	-
T_{77}	1	$15h \leq x = 24h$
T_{78}	0	-
T_{79}	0	-
T_{80}	1	$3h \leq x = 24h$
T_{81}	0	-
T_{82}	3	-
T_{83}	0	-
T_{84}	0	-
T_{85}	3	-
T_{86}	0	-
T_{88}	3	-
T_{90}	1	$12h \leq x = 24h$
T_{95}	0	-
T_{96}	0	-
T_{99}	0	-

Table 5.6. Time and Updating Indicators

Thanks to ADDM, the data flow can be greatly reduced, thereby limiting the use of the resources of ISSAUM: navigation of ICAs, server access and data transfer. (Figuer 5.12).

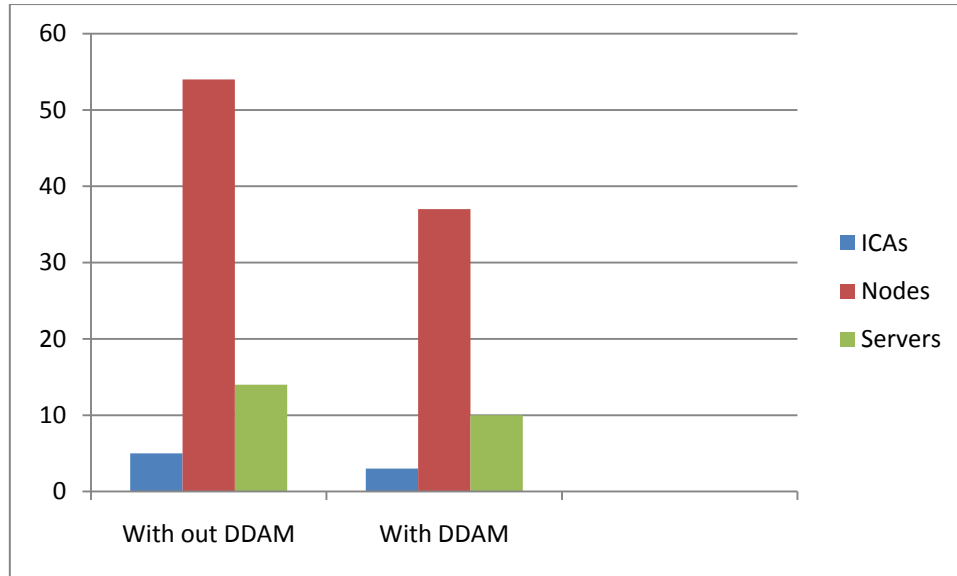


Figure 5.12. Example of the improvement of ISSAUM performance by using ADDM

5.4.4. The Ontology Mapping Negotiation Process

It supposed that the ICAs agents have to visit their first node, with respect to their FWps, without any problem before the announcement of all the nodes which are not available. For example, we suppose that there is a set of the nodes which are not available, as follow, (Figure 5.14):

$Ind_i = \{S_1, S_3, S_7, S_{17}, S_{13}, S_{19}, S_{14}, S_5, S_{12}, S_9\}$; we deduce the tasks to reallocate them:

$\Phi_i = \{T_{19}, T_{29}, T_{66}, T_{88}, T_3, T_{26}, T_{32}, T_{33}, T_{38}, T_{42}, T_{61}, T_{85}, T_{34}, T_{25}, T_{44}, T_{60}, T_{80}, T_{53}, T_6, T_{16}, T_{22}, T_{52}, T_{57}, T_{67}, T_{96}, T_{71}, T_{20}, T_{86}, T_{95}, T_{99}, T_{75}, T_{83}, T_2, T_{82}\}$;

We have the 34 tasks to reallocate.

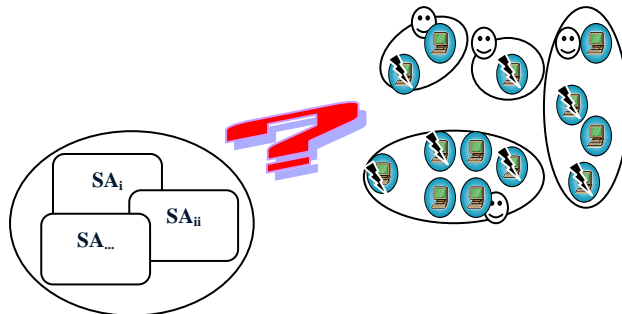


Figure .5.13. The Perturbation Case

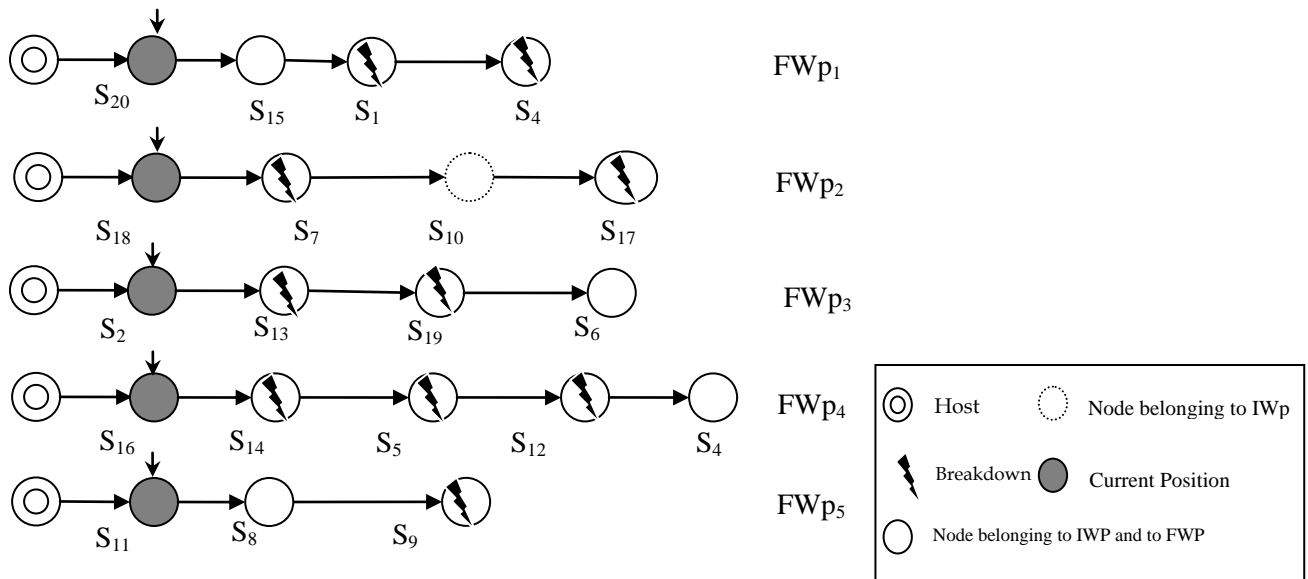


Figure .5.14. The Modification of FWps after The Perturbation Case

5.4.4.1. *First Scenario:*

This scenario is applied on one transport system (for example: the French transport operator) where it uses French Travel Ontology. In this study, we try to illustrate the case when ICAs agents don't understand the messages sent from the SA agent.

Indeed, there are the possibilities of the occurrence of the misunderstanding in our system because our agents use different ontologies for different domains (Transport, geographic, etc...). After sending all FWps to the ICA agents, as we explain in our example, the ICAs agents are supposed to visit their first nodes by the order as in their FWps without problems before the declaration of all unavailable nodes. Thus, the misunderstanding will take place after sending the first message (Propose (contract)) by the SA agent to the ICAs agents, where some of these later discover that they don't understand the contents of the message. In this case, before sending any message to the SA, ICAs ask the TA agent from the SEL to determine the level of transibility between the SA ontology and ICA ontology. According to the result, The TA can apply the Ontology Mapping Protocol (OMP) and he sends the result to the ICA agent. Then, the later decide what it must send to SA (i.e. they will send either: accept (partial) or refuse). The ICAs agents send their agreements if they would like to participate in the negotiation process.

The proposed negotiation process allows us to reassign the set Φ_i of tasks which are not available. The reassignment of tasks based on the (priorities, preference, constraint and ontology) of each ICA agent. Thus, upon the reception of the proposed contract, each ACI can be respond by a partial agreement because, in this case, the perturbation has affected a subset of each FWps.

- The ICA₁ agent will not to visit the nodes S₁ and S₃. So, he informs the SA that he no longer executes the tasks: (T₁₉, T₂₉, T₆₆, T₈₈, T₃, T₂₆, T₃₂, T₃₃, T₃₈, T₄₂, T₆₁, T₈₅);
- The ICA₂ agent will not to visit the nodes S₇ and S₁₇. So, he informs the SA that he no longer executes the tasks: (T₃₄, T₂₅, T₄₄, T₆₀, T₈₀);
- The ICA₃ agent will not to visit the nodes S₁₃ and S₁₉. So, he informs the SA that he no longer executes the tasks: (T₅₃, T₆, T₁₆, T₂₂, T₅₂, T₅₇, T₆₇, T₉₆);
- The ICA₄ agent will not to visit the nodes S₁₄, S₅ and S₁₂. So, he informs the SA that he no longer executes the tasks: (T₇₁, T₂₀, T₈₆, T₉₅, T₉₉, T₇₅, T₈₃);
- The ICA₅ agent will not to visit the node S₉. So, he informs the SA that he no longer executes the tasks: (T₂, T₈₂);

In this case, the SA agent built the Φ_i , and confirms the rested routes of the ICAs agents. Then, he asks each ICA agent to propose a new set of assignment tasks Φ_i , according to his priorities, as follow:

- There is no agent accepts to reassign T₁₉ the task because the proper servers are not available (S₁, S₅).
- The ICA₅ agent, who has not yet left the node S₁₁, proposes to reassign the task T₂₉.
- The ICA₁ agent, who is preparing to leave the node S₂₀, proposes to reassign the task T₆₆. Indeed, this task belongs to his next destination (node S₁₅).
- No agent accepts to reassign T₈₈ the task because the proper servers are not available (S₁, S₃).
- For the task T₃ proposed by :
 - ICA₂ in the node S₁₈, and in the node S₁₀ (here, this node belong to preference of ICA₂ agent);
 - ICA₄ in the nodes (S₁₆ and S₄);
 - ICA₅ in the nodes (S₁₁ and S₁₈).
 - **Ontology Mapping Protocol tour :**

After having received a *Propose* through (Ontology Negotiation Protocol (ONP)) from the SA agent and not being able to interpret the requested service:

T₃ = "Perturbations of traffic between B and C (today / this week)".

The ICA₁ sends a message with the performative NOT_UNDERSTOOD to the TA; because he uses different ontology (Transport Ontology), as we have explained in the section (4.4.2.2). ICA₁ agent asks TA about the name of the unknown service, as follow:

(NOT_UNDERSTOOD

```

:sender (agent-identifier
        : name ICA1@home:1099/JADE
        : addresses (sequence http://home:7778/acc))
:receiver (set (agent-identifier
               : name TA@home:1099/JADE
               : addresses(sequence http://home:7778/acc)))
: content (Propose ( : services T3
                    : owner ICA1@home:1099/JADE
                    : duration 120 ))
: language SPRQL
:protocol OMP
: ontology Travel Ontology)

```

Figure 5.15. The form of NOT_UNDERSTOOD Message

The TA sends the name of the service (T₃) which it has just received to the SA in order to get further information about it. The SA will analyze that request and send all the information about this service, as follow, (table 5.7):

Concepts	Attributes	Relations
Means-Of-Transport	Means-Name	Is-a
	Price	
Train	T-name	Is-a
Airplane	A-name	Is-a
Metro	M-name	Is-a
Tram	T-name	Is-a
Bus	B-name	Is-a
Perturbation	Actual Time	hasPerturbation
	Scheduled Time	
Country	Country name	hasCity
City	City name	hasCountry
		hasStation

Table 5.7. The SA agent Results for Mapping Service

Upon receiving the answer from SA, the TA knows the description of the service (T_3). He sends to ICA_1 all the information about the service under negotiation. Then, the ICA_1 sends a list containing names of potential correspondent concepts.

Concepts	Attributes	Relations
Operator	Op-name	
Transport-Line	Line name	Is-composed-of
		Served-by
Transport-Mode	Mode-Name	Is-a
	Mode-price	Used-by
Train	Train-name	Is-a
Metro	Metro-name	Is-a
Tram	Tram-name	Is-a
Bus	Bus-name	Is-a

Table 5.8. The ICA Agent Result for Mapping Service

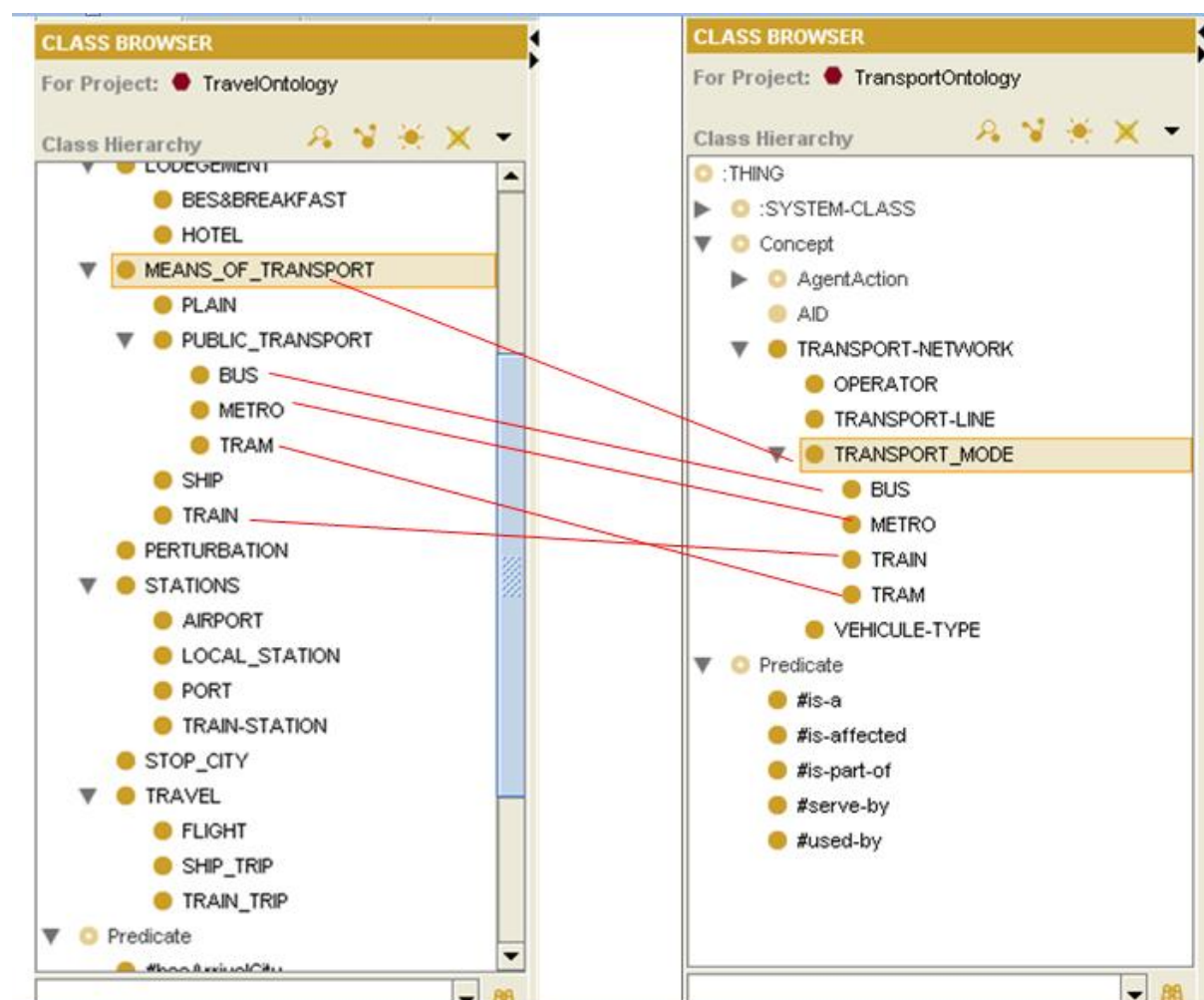


Figure 5.16. The Mapping Process in Our Example

After receiving all the information about the service under negotiation and a list of possible corresponding services, the TA is now able to apply the similarity methods in order to match the services, (section 5.4).

Travel Ontology	Transport Ontology	Confidence		
MEANS-OF-TRANSPORT	Transport- Mode	0.5	0.9	0.83
TRAIN	Train	1.0		
METRO	Metro	1.0		
TRAM	Tram	1.0		
BUS	Bus	1.0		
Means-Name	Mode-Name	0.30	0.61	
price	Mode-price	0.60		
T-name	Train-name	0.70		
M-name	Metro-name	0.70		
T-name	Tram-name	0.70		
B-name	Bus-name	0.70		
Is-a	Is-a	1.0	1.0	

Table 5.9. The Result of Similarity Measures

The global score for concepts (classes) matching is the average of their matching confidence:

$$\text{Class-score} = (0.5 + 0.1 + 0.1 + 0.1 + 0.1) / 5 = 0.90.$$

The global score for attribute matching is the average of each individual attribute matching.

$$\text{Attribute-score} = (0.30 + 0.60 + 0.70 + 0.70 + 0.70 + 0.70) / 6 = 0.61.$$

The final score is then the average of (concepts, attributes, and relations) will be:

$$\text{Final-Score} = (0.90 + 0.61 + 1.0) / 3 = 0.83.$$

All the mapping results are stored on the server side for future negotiation rounds. As a result, if the same pair of concepts needs to be compared again, there is no need to repeat the whole matching process. As already explained in the previous Chapter, the performance improvement of the system occurs with time and depends on the quantity of the negotiations and on the number of matching's performed.

Finally, ICA₁ agent proposes to assign the task T₃ in the nodes (S₂₀ and S₁₅);

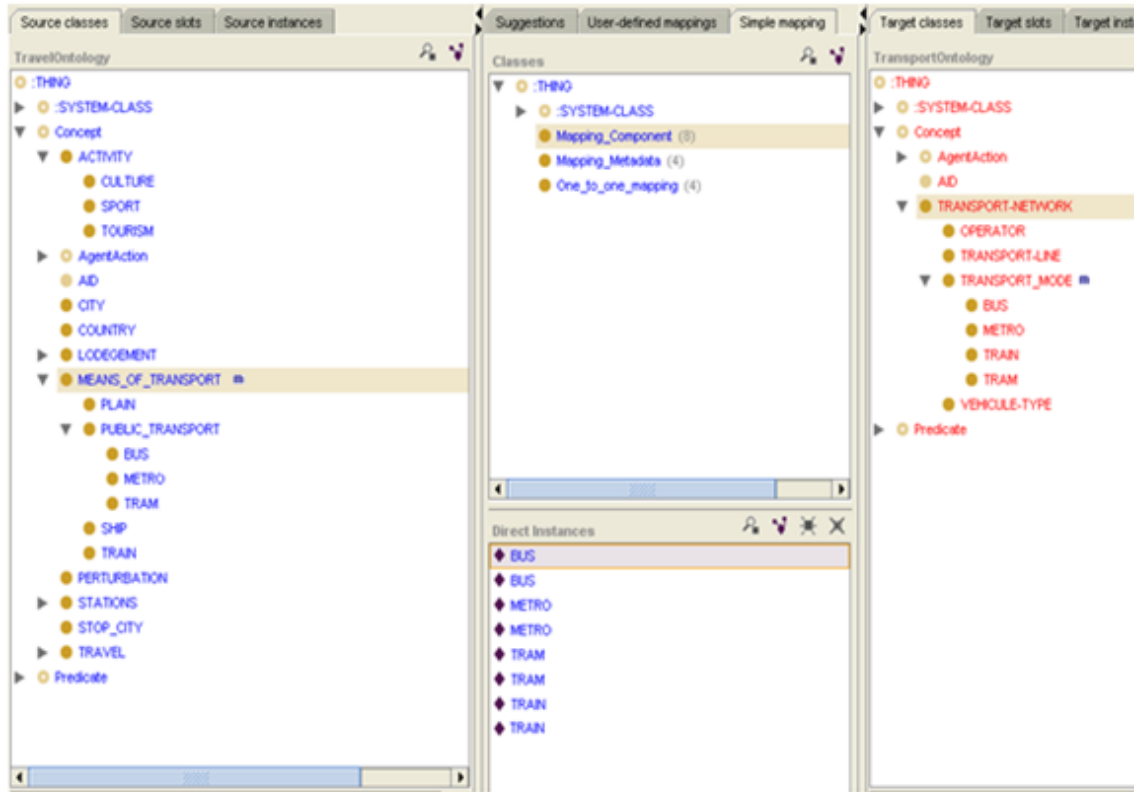


Figure 5.17. Mapping Results

- The task T₂₆ is proposed by :
 - ICA₂ in the node S₁₈;
 - ICA₃ in the node S₂;
 - ICA₅ in the node S₁₁;
- The task T₃₂ is proposed by :
 - ICA₂ in the node S₁₈;
 - ICA₄ in the node S₄;
 - ICA₅ in the node S₈;
- The task T₃₃ is proposed by ICA₄ agent in the nodes(S₁₈ and S₂);
- There is no agent accepts to reassign the task T₃₈ because the possible providers of this task, which belong to the priorities of ICA₁, ICA₂, ICA₄, respectively, are not available (S₁₄, S₁₇). The node S₁₀ proposes also the task T₃₈ may S₁₀ belongs to the preference of the ICA₂.

- The task T_{42} is proposed by :
 - ICA_2 in the node S_{18} ;
 - ICA_5 in the node S_{11} ;
- The task T_{61} is proposed by the agent ICA_1 in the node S_{15} ;
- The task T_{34} is proposed by the agent ICA_3 in the node S_2 ;
- The task T_{85} is proposed by the agent ICA_2 in the node S_{18} ;
- There is no agent accepts to reassign the task T_{71} because the possible providers of this task are not available (S_3 , S_7 and S_{14}).
- The task T_{20} is proposed by :
 - ICA_1 in the nodes (S_{20} and S_{15});
 - ICA_2 in the node S_{10} which belong to his preference;
- The task T_{86} is proposed by ICA_2 in the nodes S_{18} ;
- The task T_{95} is proposed by ICA_5 in the nodes S_{11} ;
- There is no agent accepts to reassign the task T_{99} because the possible providers of this task are not available (S_1 and S_5).
- The task T_{25} is proposed by :
 - ICA_1 in the nodes S_{20} ;
 - ICA_4 in the nodes (S_{16} and S_4);
- The task T_{44} is proposed by ICA_5 in the node S_{11} ;
- The task T_{60} is proposed by ICA_2 in the node S_{18} ;
- The task T_{80} is proposed by ICA_1 in the nodes S_{20} ;
- The task T_{75} is proposed by ICA_2 in the nodes S_{10} which belong to his preference;
- There is no agent accepts to reassign the task T_{83} because the possible providers of this task are not available (S_5 , S_7 , S_{12} and S_{14}).
- The task T_2 is proposed by :
 - ICA_1 in the nodes (S_{20} and S_{15});

- ICA₂ in the node S₁₈ and in the node S₁₀ which belong to his preference;
- ICA₄ in the node S₄
- ICA₅ in the node S₈
- The task T₈₂ is proposed by ICA₅ in the nodes S₈;
- The task T₅₃ is proposed by ICA₄ in the nodes S₁₆;
- The task T₆ is proposed by :
 - ICA₁ in the nodes (S₂₀ and S₁₅);
 - ICA₂ in the node S₁₈ and in the node S₁₀ which belong to his preference;
 - ICA₄ in the nodes (S₄ and S₁₆);
 - ICA₅ in the nodes (S₈ and S₁₁);
- There is no agent accepts to reassign the task T₁₆ because the possible providers of this task are not available (S₅, S₇ and S₁₉).

In this scenario, the French transport system remark that the T₁₆ is a travel from B=Paris to C= London today (at the instant t=11.00 or starting from 12.00) (i.e. it can demand the reassignment of T₁₆ from the English transport operator, as we will see in the second scenario).

- The task T₂₂ is proposed by ICA₁ in the nodes S₂₀;
- There is no agent accepts to reassign the task T₅₂ because the possible providers of this task are not available (S₁₇ and S₁₉).
- The task T₅₇ is proposed by ICA₂ in the nodes S₁₈;
- There is no agent accepts to reassign the task T₇₆ because the possible providers of this task are not available (S₅, S₁₃ and S₁₉).
- Finally, there is no agent accepts to reassign the task T₉₆ because the possible provider of this task is not available (S₁₉).

As we mentioned, each ICA agent proposed all the possible assigned tasks according to his priorities. When the SA agent receives those propositions, he will decide the new contract, as follow:

- Direct reassigned tasks , as follow, (unique choice) :

T₂₉, T₆₆, T₆₁, T₈₅, T₃₄, T₈₆, T₉₅, T₄₄, T₆₀, T₈₀, T₈₂, T₅₃, T₂₂, T₅₇ ;

- direct reassigned tasks according to the optimization process, as follows, (multi choice):

$$T_3(S_{20}), T_{26}(S_{11}), T_{32}(S_8), T_{33}(S_{16}), T_{42}(S_{18}), T_{20}(S_{20}), T_{25}(S_{16}), T_2(S_8), T_6(S_{18}) ;$$

The answers of the ICAs on the SA agent propose are:

- The ICA₁ agent rejects the choice of SA agent for the tasks T₃ and T₂₀ in the node S₂₀ because he has already left the node. But he accepts the rest of the proposition, (partial accept).
- The ICA₄ agent rejects the choice of SA agent for the tasks T₂₅ and T₃₃ in the node S₁₆ because he has already left the node. But he accepts the rest of the proposition, (partial accept).
- The agents ICA₂, ICA₃, ICA₅, accept totally the contract, (accept total).

Thus, SA agent updates Φ_i , as follows:

$$\Phi_i = \{ T_3, T_{20}, T_{25}, T_{33}, T_{75}, T_{19}, T_{88}, T_{38}, T_{71}, T_{99}, T_{83}, T_{16}, T_{52}, T_{67}, T_{96} \}$$

The SA agent confirms the remaining routes of the ICA agents and he asks each ICA agent to propose a new set of assignment tasks Φ_i according to his priorities:

- The ICA₁ agent propos to assigned the tasks T₃ and T₂₀ in the node S₁₅;
- The ICA₄ agent propos to assigned the tasks T₃₃ and T₂₅ in the node S₄;

Upon the receiving of those propositions, the SA agent sends a new contract, which contains all the proposed assignments, to the ICAs agents. Then, the agents (ICA₁ and ICA₄) send the accept total which they will confirm by the SA agent. Then, the Φ_i will be as follow:

$$\Phi_i = \{ T_{75}, T_{19}, T_{88}, T_{38}, T_{71}, T_{99}, T_{83}, T_{16}, T_{52}, T_{67}, T_{96} \}.$$

In this case, the SA agent asks each ICA agent to propose a new set of assignment tasks Φ_i according to his preference. Therefore, the ICA₂ agent proposes to assign the task T₇₅ in the node S₁₀.

Thus, the SA agent sends a contract, which contains this assignment, to the ICA₂ agent to approve it. The ICA₂ accept and the SA agent updates the Φ_i as follow:

$$\Phi_i = \{ T_{19}, T_{88}, T_{38}, T_{71}, T_{99}, T_{83}, T_{16}, T_{52}, T_{67}, T_{96} \}.$$

For this example, the negotiation process stops at this level because all the possible servers of the rested set of tasks, which need to be reassigned, are not available (in the transport operator).

Finally, in this example, the SA agent decides to send the T_{16} to the English transport operator by using the Meta-System to continue the negotiation process. As we will see

5.4.4.2. *Second Scenario*

In the second scenario, we apply our NOKMS on multi-transport operators (for example: French transport operator, English transport operator and German transport operator), which are heterogeneous community of multi-agent systems.

The French's customers want to travel to other cities out of France. The French transport system (Sys1), in this case, firstly its SA agent sends the propos (contract) message to its ICAs participant, as we noted in the first scenario.

In some times, ICAs agent cannot reassign all the tasks as T_{16} where this task can be achieved by another system like English transport system (Sys2). The usage of another transport system comes from the flexibility of our NOKMS architecture.

In this state, the Sys1 sends their query to the Sys2 through the Meta-System which considers as the intermediate between the two systems, and which in turn, interprets the incoming ACL-Sys1 based on its NOKMS structure. The interpreted message is then converted into an 'interlingua' representation inside the Meta-System. Where, The Meta-System translates the Interlingua representation to the destination ACL-Sys2.

As an example: when the Sys1 have found that it cannot reassign the task T_{16} and this task can be assigned by another system, then it send this task to the Meta-System using its French Transport Ontology as follow:

$T_{16} = \text{"Voyager de l'endroit B à l'endroit C (aujourd'hui, à l'instant } t/\text{aujourd'hui, à partir de 12:00)"}"$

The Meta-System is then tries to translate this task, where firstly it verifies the level of transibility between the two ontologies in its SEL. The later evokes the KMSL which translate the proposed expression to the determined ontology (English Transport Ontology in our case), then the KSML return the following result:

$T_{16} = \text{"Travel from B to C today (at the instant } t=11.00 \text{ or starting from 12.00)"}"$

In some cases, when KMSL verifies the IKB, and find that it has not the suitable ontology to translate the coming ontology form Sys1to Sys2. The KMSL uses the KC to create the new ontology correspond to the ontology of Sys2 according to some policies. After translating the ACL-Sys1, Meta-System sends the new ACL which correspond to the ACL-Sys2 to the Sys2 to start the new tour of negotiation between the two systems. The Meta-System currently adopts FIPA semantic model which described in Semantic Language (SL), as the Interlingua of the agent communications.

5.4.4.3. *Negotiation Tours*

The proposed negotiation process allows the reassignment of the cancelled services. The two figures below represent different generated optimal solutions instances assignments for the same network error scenario. The first figure (5.18), as we mentioned previously, the results of mapping stored on the server side for future negotiation rounds. Thus, the negotiation process will improve in the time. In fact, we have applied our example in two cases:

- In the first case, we don't use our ONP without the applying the mapping between the concept by using OMP.
- In the second case, we have used our OMP to apply the mapping

The figure below illustrates a comparison between the two cases.

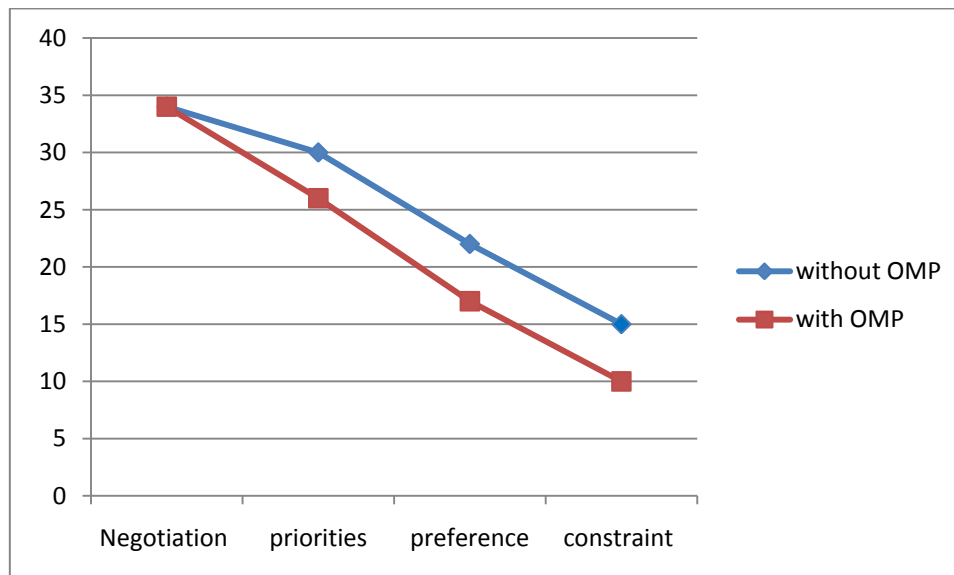


Figure 5.18. Negotiation Tours According to OMP

The list below presents the concepts memorized during a set of negotiations tours, Table 5.1.

<p>Travel: Trip, journey, jaunt, type-of-trip ;</p> <p>City: metropolis, urban centre, position, geographic-elements;</p> <p>Country: state, rural area;</p> <p>Means-of-transport: Transport-mode, transportation, Transport-network, transport-line, Vehicles;</p> <p>Stop-City: stop, point, changed-point, exchange-pole, connection-point, connection-link;</p> <p>Activity: interests, facilities, leisure-activities, museum, eating-out, nightlife, swimming, shopping;</p> <p>Activity Date: time-to-spend;</p> <p>Etc....</p>
--

Table 5.10. The Concepts Memorized During a Set of Negotiations Tours

The second figure (5.19) represents the negotiation torus without using our NOKMS and with our NOKMS. Remark that our proposed NOKMS give the flexibility to find new available providers out of its system where it doesn't find the suitable providers in it as in the case of the task T_{16} . Through an agreement between its SA agents and the new ICAs agents in the new system .the two systems connect with each other by Meta-System which consider as the intermediate between the two systems. So the correspondent transport users are satisfied in case of some network perturbations.

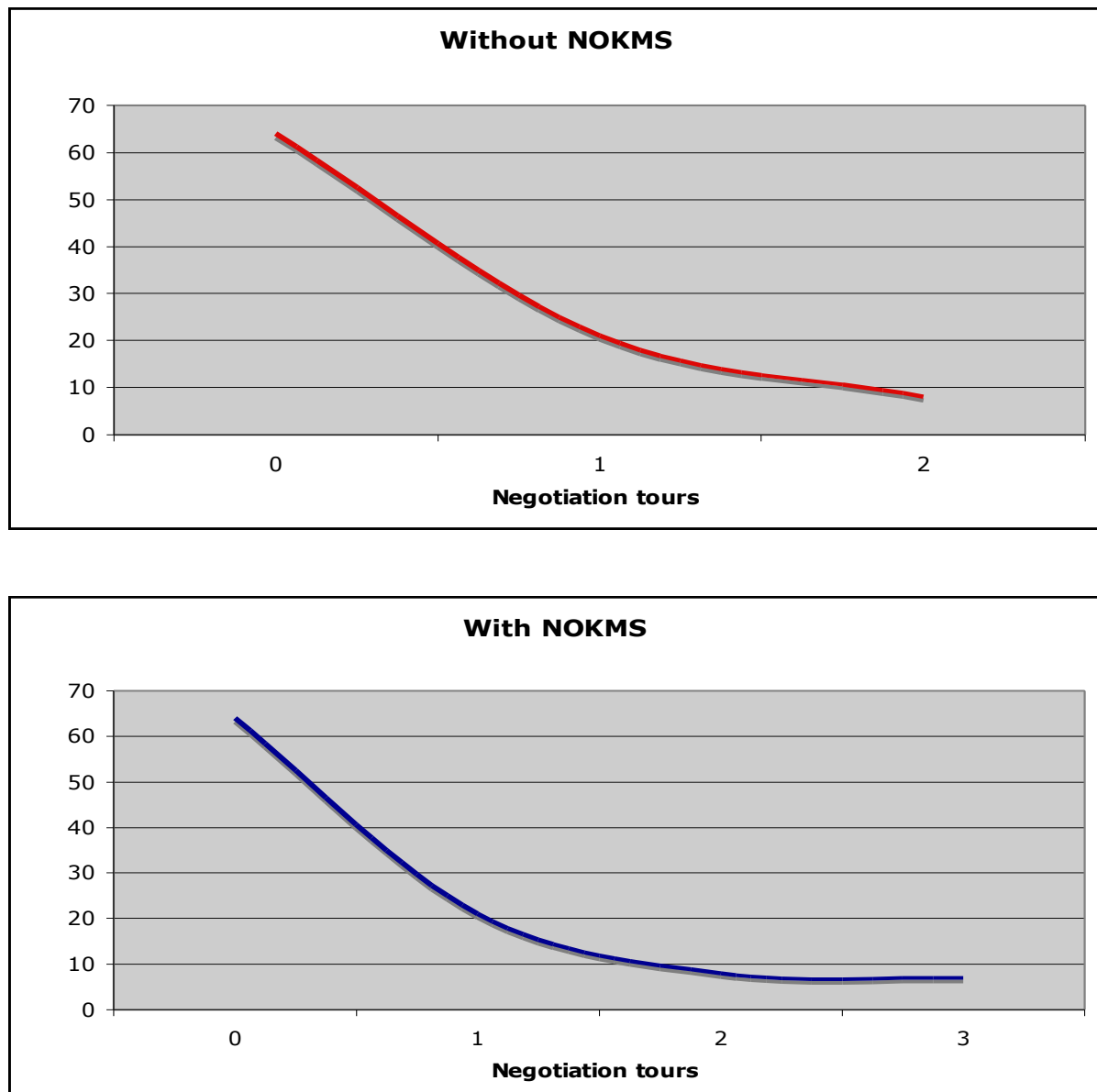


Figure 5.19. Negotiation Tours according to the NOKMS application

5.5. Summary

In this chapter, we have illustrated the work of our ISSAUM by using the flexible travel ontology to describe the meanings of the services which they represent. The mobile agent in our system commits to a top-level ontology, that defines specific vocabulary of transport domain for each information provider, made of terms used during the negotiation process.

The results of different simulation scenarios represent the robustness of our system in supporting the exponential growth of services available on large distributed

networks by using the flexible ontology. This ontology supports the services modelling to aid the decomposition of user's requests step.

In this chapter, we define a MA negotiation process to reassign required services to available network nodes. The reassignment process depends not only on the current positions of the MAs but also on their priorities, preferences constraints, and ontologies in their correspondent routes.

We have illustrated also the functionality of TA agent which aids the ICA to understand the messages sending from the SA agent by applying the ontology mapping process.

Conclusion and Future Works

In this thesis, we have proposed an information system of services to aid the urban mobility (ISSAUM) which is based on the multi-agent system approach with the contribution of the ontology approach and the optimization methods. The architecture of our system is open, dynamic and distributed where it consists of society of the agent whose life cycle based on the use of the system.

This work allows to optimize the management of the data flow of the users' requests, which can be simultaneous and numerous. The optimization take in the account the information cost, the response delay, the different size and format of the data, and the different provider's ontologies. The system is based on the MA diagram whose use was proved and tested.

The problem of handling heterogeneous information from different information providers that use different ontologies has to fully understand the exchanged services. So, agents cooperating in our ISSAUM must share a common ontology. Since our system is an open system composed of heterogeneous, autonomous agents. So, we have developed our travel ontology which is used to represent the common travel knowledge domain and it is also support the services modeling.

Our ISSAUM takes into account possible disturbance through the EMTN (crash, bottlenecks, etc.) in order to satisfy user requests in all the cases. For that, we developed a negotiation protocol between optimiser agents and mobile agents of the system. The proposed ontology mapping negotiation model based on the knowledge management system for supporting the semantic heterogeneity and it organized as follow: the first layer contains the *Negotiation Layer (NL)*. The second layer represents the *Semantic Layer (SEL)*, and the third layer is the *Knowledge Management Systems Layer (KMSL)* which uses ontology in purpose of automatic classifying and using of the news ontologies and meta-ontologies. Our approach aims to make the agents able to understand each other when using these ontologies and by applying the mapping services to resolved the misunderstanding problem.

We detailed the reassignment process by using Dynamic Reassigned Tasks (DRT) algorithm supporting by ontology mapping approach. The DRT Mapping algorithm based on the current state of travelling mobile agents in their correspondent routes called Workplans. Our goal is to give users all needed information even if some information providers are no longer available. Thus we improve the Quality of Services (QoS) of the response time with the best cost.

Finally, we presented the simulation results of this thesis by justifying the usage of the ontology solution in our system. We illustrated the role of our flexible in the negotiation process by applying the ontology mapping process in the cases of misunderstanding. We explained the different experimental scenarios which show a pertinent management of any amount of Δ_ε – simultaneous requests. Indeed, a great number of user requests through a short period of time Δ_ε , does not affect the system functioning which decompose them by using its flexible ontology, identifies the required services and the possible information providers.

Future Works

In our future researches, we aim to apply our (ISSAUM) system on different transport operators in a real-world to observe the performance of our flexible ontology.

We have obtained the results of our system with ontologies based on real information in the transport domain, but we also would like to test our approach in other application domains, (ex: medical system).

We would like to improve the mapping between different data types and improve the functionality of our NOKMS by using several SA agents to negotiate with several ICA agents.

Finally, we aim to integrate the web services in our system by using different ontology format.

References

- [Baader et al., 03] Baader, F., McGuinness, D., Nardi, D., Patel-Scheider, P. The Description Logic Handbook: Theory, implementation and application. Cambridge University Press, Cambridge, United Kingdom. 2003
- [Bechhofer et al., 01a]. Bechhofer, S., Goble, C., and Horrocks, I. DAML+OIL is not enough. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, Stanford University, California, USA. 2001
- [Bechhofer et al., 01b] Bechhofer, S., Horrocks, I., Goble, C., Stevens, R. OilEd: a Reason-able Ontology Editor for the Semantic Web, In *Proceedings of 14th International Workshop on Description Logics*, Stanford, USA, August, 2001.
- [Bernaras et al. 96] A. Bernaras, I. Laresgoiti, and J. Corera. Building and reusing ontologies for electrical network applications". In W. Wahlster, editor, *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI)*, pages 298–302, John Wiley & Sons, Ltd, Chichester, England, 1996.
- [Bermejo-Alonso, 06] J. Bermejo-Alonso, R. Sanz and I. L'opez, A Survey on Ontologies for Agents From Theory to Practice, ASLab ASL-A-2006-XXX v 1.0 Draft, June 12, 2006
- [Brickley and Guha, 02] Brickley D., Guha R. V. RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft, 2002. <http://www.w3.org/TR/PR-rdf-schema>
- [Bruijn et al., 06]. Jos de Bruijn Marc Ehrig Cristina Feier, Ontology mediation, merging and aligning , Francisco Mart'ın-Recuerda Fran,cois Scharffe, Moritz Weiten, May 20, 2006
- [Borst, 79] Borst, W. N. Construction of Engineering Ontologies. University of Twente. Enschede, Netherlands - Centre for Telematica and Information Technology, 1997.
- [Bouquet et al., 04] Paolo Bouquet, Jérôme Euzenat, Enrico Franconi, Luciano Serafini, Giorgos Stamou, and Sergio Tessaris. D2.2.1 specification of a common framework for characterizing alignment. Technical report, University of Trento, 2004.
- [Bouquet et al., 04a] Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L. & Stuckenschmidt, H. (2004), 'Contextualizing ontologies', *Journal of Web Semantics* , 1(4), 325.
- [Canada, 03] Transport Canada, Intelligent Transportation Systems Research and Development Plan for Canada: Innovation through Partnership. (website: <http://www.its-sti.gc.ca/en/randd/menu.htm>), (2003)
- [Carey et al., 79]; M. Carey and D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman, 1979.
- [Chaib-Draa and Dignum, 02] B. Chaib-Draa, F. Dignum, Trends in Agent communication Language, Computational Intelligence, Volume 18, Number 2, 2002
- [Caire and Cabanillas, 04] Caire, G., Cabanillas, D. JADE Tutorial Application-defined content languages and ontologies, Technical Report, TILab, 2004.
- [Ciocoiu et al., 01] Ciocoiu, M., Gruninger, M., Nau, D.S. Ontologies for Integrating Engineering Applications. *Journal of Computing and Information Science in Engineering* 1(1): 12-22. March 2001.

- [Cranefield and Purvis, 99] Cranefield, S. and Purvis, M. UML as an ontology modeling language, In Proceedings of the 146 Bibliography Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99), 1999.
- [Colombetti, 98] M. Colombetti, Different ways to have something in common. Proceedings of the third international conference on flexible query answering systems (FQAS '98). Springer, Berlin, (1998)
- [Davics, 91] L.Davics, *Handbook of genetic algorithm*, New York: Van Nostrand Reinhold, 1991
- [Domingue, 1998] Domingue, J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In: Gaines, B.R, Musen, M.A. (eds.) 11th International Workshop on Knowledge Management Support, IEEE Intelligent Systems & their applications, Vol.15(3):26-32, 1998.
- [Doan et al.,04] Doan, A., Madhavan, J., Domingos, P. & Halevy, A. (2004), Ontology matching: A machine learning approach, in S. Staab & R. Studer, eds, 'Handbook on Ontologies in Information Systems', Springer-Verlag, pp. 397-416.
- [Ehrig and Sure, 04a] Ehrig, M., Sure, Y., Ontology mapping - an integrated approach, In Proceedings of the European Semantic Web Conference (ESWS), pp. 76-91, 2004.
- [Ehrig and Staab, 04 b] Ehrig, M., Staab, S. QOM - Quick Ontology Mapping, In: Proceedings of the 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan. LNCS, Springer, 2004.
- [Falasconi et al., 96] S. Falasconi, G. Lanzola, and M. Stefanelli. Using ontologies in multi-agent systems. In *Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW)*, Banff, Alberta, Canada, 1996. University of Calgary.
- [Farquhar et al. 97] A. Farquhar, R. Fikes, and J. Rice. "The ontolingua server: a tool for collaborative ontology construction". *International Journal of Human-Computer Studies*, volume 46, pages 707-728, 1997.
- [Ferber, 99] Gerber, J." Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence", Addison-Wesley Longman, New York, 1999.
- [Fensel et al., 00] Fensel, D., Horrocks, I., van Harmelen, F., Decker, S., Erdmann, M., Klein, M. OIL in a nutshell. In: Dieng, R., et al. (eds.) Knowledge Acquisition, Modeling, and Management, In Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), LNAI, Springer-Verlag, 2000.
- [Fikes et al., 03] Richard Fikes, P.H., and Ian Horrocks, *OWL-QL – A Language for Deductive Query Answering on the Semantic Web*. KSL 03-14, 2003.
- [FIPA, 99] <http://www.fipa.org/specs/fipa00086/XC00086C.html>
- [Florez, 99] R.A.Florez-Mendez." Towards a Standardization of Multi-Agent System Frameworks". *ACM Crossroads Student Magazine*. Canada, 1999.
- [Geiger, 95] Geiger, K. Inside ODBC. Microsoft Press. 1995.
- [Green et al., 97] S.Green, L.Hurst, B. Nangle, P.Cunningham, F. Somers and R. Evans, "Software agents: A review", *Technical report*, TCS-CS-1997-06, Trinity College Dublin, Ireland (1997).
- [Gomez-Perez et al., 02] Gomez-Perez, A., Angele, J., Fernandez-Lopez, M., Christophides, V., Stutt, A. et Sure, Y. A survey on ontology tools. OntoWeb - Ontology-based information exchange for

knowledge management and electronic commerce, IST-2000-29243. Deliverable 1.3, Universidad Politecnica de Madrid. 2002.

[Gruber, 93] Gruber, T. R. Toward Principles for Design of Ontologies Used for Knowledge Sharing, In: Guarino, N. and Poli, R. (eds.) *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, 1993.

[Gruber,93a] Gruber, T. R. A Translation approach to portable ontology, specification, Knowledge Acquisition, Vol.5(2):199-200, 1993.

[Gruninger and Fox, 95] Gruninger, M.; Fox, M. Methodology for the design and evaluation of ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, in IJCAI-95, Canada. 1995

[Guarino , 98] N. Guarino. "Formal ontologies and information systems". In N. Guarino, editor, *Proceedings of FOIS'98*, page , IOS Press, Amsterdam, 1998.

[Horrocks 1998] Ian Horrocks: "The FaCT system", Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98, number 1397 in Lecture Notes in Artificial Intelligence, page 307-312. Springer, May 1998.

[Horrocks, 03] Horrocks, I. An example OWL ontology. (Foils from the ISCW 2003 tutorial). 2003;

[Horrocks, 04]: Ian Horrocks, Reasoning with Expressive Description Logics: Logical Foundations for the Semantic Web, Keynote talk at ICIIP, Beijing, China, October 2004

[Jennings et al., 00] N. R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated haggling: Building artificial negotiators. In *Pacific Rim International Conference on Artificial Intelligence*, page 1, 2000.d

[Kone et al., 00] M.T.Kone, A. Shimazu, T.Nakajima, The State of The Art in agent communication Languages. Knowledge and Inforamtion Systems, 2000

[Kraus, 01] S. Kraus, Automated Negotiation and Decision Making in Multiagent Environments, M. Luck et al. (Eds.): ACAI 2001, LNAI 2086, pp. 150–172, 2001. Berlin Heidelberg 2001

[Laclavík, 05] Michal Laclavík, "Ontology and Agent based Approach for Knowledge Management", *PhD, Slovakia*, 2005

[Lassila and Swick, 1999] Lassila, O., Swick, R. Resource Description Frame-work (RDF) Model and Syntax Specification, W3C Recommendation, 1999. <http://www.w3.org/TR/REC-rdf-syntax>

[Lassila and McGuinness, 2001] Lassila, O., McGuinness, D. L. The Role of Frame-Based Representation on the Semantic Web, Knowledge Systems Laboratory, Report KSL-01-02, January, 2001.

[Lopez, 99] Fernandez-Lopez, M., A.Gomez-Perez and J. Pazos Sierra, Building a Chemical Ontology Using Methontology and the Ontology Design Environment, IEEE Intelligent Systems, 1999.

[Luck, 03] Michael Luck, Peter McBurney, Chris Preist, "Agent Technology: Enabling Next Generation Computing", *A Roadmap for Agent Based Computing*, 2003

[Maedche et al., 02] Maedche, A., Motik, B., no Silva, N. & Volz, R. (2002), Mafra a mapping framework for distributed ontologies, in 'Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW-2002', Madrid, Spain.

- [Malucelli et al., 2006] Malucelli, A., Palzer, D., Oliveira, E. Ontology-based Services to help solving the heterogeneity problem in e-commerce negotiations. To be published in Journal of Electronic Commerce Research and Applications Special Issue Electronic data engineering: the next frontier in e-commerce, Vol.5(3), Elsevier, 2006.
- [Marilleau, 05] Nicolas Marilleau, An Agent Based Meta-Model For Urban Mobility Modeling, Proceedings of the First International Conference on Distributed Frameworks for Multimedia Applications (DFMA'05)
- [McGuinness et al., 2000] McGuinness, D. L., Fikes, R., Rice, J. and Wilder, S. An environment for merging and testing large ontologies, In Proceedings of 7th International Conference on Principles of Knowledge Representation and Reasoning. Colorado, USA, April, 2000.
- [Mizoguchi, 04] Mizoguchi, R., Tutorial on ontological engineering - Part 3: Advanced course of ontological engineering, New Generation Computing, OhmSha&Springer, Vol.22, No.2, 2004
- [Noy et al., 00] Noy, N. F., Ferguson, R. W., Musen, M. A. The knowledge model of Protégé-2000: Combining interoperability and exibility. In: Dieng, R., Corby, O. (eds.), Knowledge Engineering and Knowledge Management. Methods, Models, and Tools: 12th International Conference, LNCS, pp.17, 2000.
- [Noy and McGuinness, 00a] Noy, N. F. & Musen, M. A. (2000b), PROMPT: Algorithm and tool for automated ontology merging and alignment, in 'Proc. 17th Natl. Conf. On Artificial Intelligence (AAAI2000)', Austin, Texas, USA.
- [Noy and McGuinness, 00b] Noy, N. F. & Musen, M. A. (2000a), Anchor-PROMPT: Using non-local context for semantic matching, in 'Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)', Seattle, WA, USA.
- [Noy and McGuinness, 01] Noy, N. F., McGuinness, D. L. Ontology Development 101: A Guide to Creating your First Ontology, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March, 2001.
- [Obrst, 03] Leo Obrst. Ontologies for semantically interoperable systems. In CIKM '03: Proceedings of the Twelfth international conference on Information and knowledge management, pages 366-369, New York, NY, USA, 2003. ACM Press.
- [Peter, 03] Peter F. Patel-Schneider, Pat Hayes and Ian Horrocks: *OWL Web Ontology Language Semantics and Abstract Syntax*, W3C Working Draft, 2003.
- [Prasenjit et al., 05] Prasenjit Mitra, Natasha F. Noy, Anju Jaiswals, "OMEN: A Probabilistic Ontology Mapping Tool", International Semantic Web Conference 2005: 537-547.
- [Rouvrais, 02] S. Rouvrais, *Utilisation d'agents mobiles pour la construction de services distribués*, PhD Thesis, University of Rennes, France (2002).
- [Saad et al., 08a]. S. Saad, H. Zgaya, S. Hammadi: "The Flexible Negotiation Ontology-based Knowledge Management System: The Transport Ontology Case Study", In proceedings of the IEEE, the International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA'08). April 7 - 11, 2008 Damas, Syrie.
- [Saad et al., 08b]. S. Saad, H. Zgaya, S. Hammadi. "Une Ontologie de négociation flexible basée sur un système de gestion de connaissances : Cas d'étude d'Une Ontologie dans le domaine de Transport". Le congré de la Recherche Opérationnelle et de l'Aide à la Décision sont éligibles pour la conférence (ROADEF'08). Februry, 25-27, 2008, à Clermont-Ferrand, France

- [Saad et al.,08c]. S.Saad, H.Zgaya, S.Hammadi. "Using Ontology to Solve the Negotiation Problems in Mobile Agent Information Systems", In proceedings of the IEEE, the International Conference on Systems, Man, and Cybernetics (SMC'08). October 12-15, 2008, Singapore.
- [Saad et al.,08d]. S.Saad, H.Zgaya, H.Hammadi: "Novel Ontology Model for Communicating Heterogeneous Negotiation Mobile-Agent in a Transport Environment", Studies in Informatics and Control Journal (SIC), National Institute for R&D in Informatics ICI Bucharest, Volume 17 – Num 4, page (333-352) , December 2008.
- [Saad et al.,10]. J. S.Saad, H.Zgaya, H.Hammadi: "Knowledge Management - integral part of information and communications technology". Chapter:"Towards the Optimization of Client Transport Services: Negotiating by Ontology Mapping Approach between Mobile Agents". Page (195-220). April 2010; isbn: 978-953-7619-94-7.
- [Seaborne, 02] Seaborne, A., Jena . *Tutorial A Programmer's Introduction to RDQL*. HP Labs, 2002.
- [Seaborne et al., 05], Seaborne,, Hewlett-Packard Laboratories, Bristol, SPARQL Query Language for RDF. 2005.
- [Smadi and Miner, 05] A.Smadi, K.Miner, North Dakota Department of Transportation Intelligent Transportation Systems Statewide Plan, Final Report October 2004
- [Studer et al.,98] Studer, R., Benjamins, V.R., Fensel, D. *Knowledge Engineering: Principles and Methods*, IEEE Transactions on Data and Knowledge Engineering, Vol. 25(1-2):161-197, 1998
- [Sure et al., 2002] Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D. OntoEdit: Collaborative Ontology Engineering for the Semantic Web, In: Horrocks I., Hendler J. (eds.), First International Semantic Web Conference (ISWC'02), Sardinia, Italy, Springer Verlag, LNCS 2342, Berlin, Germany, pp. 221-235, 2002.
- [Thalheim, 00] Thalheim, B. *Entity-relationship modelling, Foundation of Database Technology*, Springer-Verlag, Berlin, Germany, 2000.
- [Theilmann, 99] W.Theilmann and K.Rothermel, "Efficient Dissemination of Mobile Agents", in *Proc. of the 19th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'99)*, pp. 9-14, edited by W. Sun, S. Chanson, D. Tygar and P. Dasgupta, Austin, TX, USA, 31May-5Jun 1999.
- [United States,03] United States Department of Transportation, Intelligent Transportation Systems Benefits and Costs 2003 Update, May 2003, www.its.dot.gov
- [Uschold, and Gruninger,96] Uschold, M. and Gruninger, M. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*. 1996.
- [Uschold and King , 95] Uschold, M.; King, M. Towards a Methodology for Building Ontologies. In: Skuce, D. (Ed.) *IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal, Canada, pp 6.1–6.10. 1995
- [Uschold , 98] M. Uschold. "Knowledge level modelling: concepts and terminology". *Knowledge Engineering Review*, 1998.
- [van Diggelen et al., 07] van Diggelen. J, Beun.R.G, Dignum.F, Rogier M. van Eijk and Meyer. *Ontology negotiation in heterogeneous multi-agent systems: The ANEMONE system*, Applied Ontology 2 (2007) 267–303 267.IOS Press

- [van Diggelen et al., 06] van Diggelen, J., Beun, R.G., Dignum, F., Rogier, M., van Eijk, and Meyer. *ANEMONE: An Effective Minimal Ontology Negotiation Environment*. AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.
- [Van Heijst et al., 97] Van Heijst, G., Schreiber, A. Th., Wielinga, B. J. *Using explicit ontologies in kbs development*, International Journal of Human-Computer Studies, Vol.45:184-292, 1997.
- [Wiesman and Roos, 04] Wiesman, F., Roos, N. Domain independent learning of ontology mappings, AAMAS, ACM Press, New York, USA, pp.846-853, 2004.
- [Visser et al., 97] Visser, P. R. S., Jones, D.M., Bench-Capon, T. J. M., and Shave, M. J. R. (1997). An analysis of ontological mismatches: Heterogeneity versus interoperability. In AAAI 1997 Spring Symposium on Ontological Engineering, Stanford, USA.
- [Wache et al., 01] Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S. *Ontology-Based Integration of Information-A Survey of Existing Approaches*, In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Workshop Ontologies and Information Sharing, Seattle, USA, 2001.
- [Walton and Krabbe, 95] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany NY, USA, 1995.
- [Zgaya, 07] H. Zgaya : Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la composition des services liés au transport. PHD thesis, EC-Lille, France, 2007
- [Zgaya et al., 05a] H. Zgaya, S. Hammadi, K. Ghédira "Workplan Mobile Agent for the Transport Network Application", IMACS'2005, Paris, July 2005.
- [Zgaya et al., 05b] H. Zgaya, S. Hammadi, K. Ghédira *Evolutionary method to optimize Workplan mobile agent for the transport network application*, IEEE SMC'2005, Hawaii, USA 10-12 October 2005.
- [Zidi, 06] K. Zidi. *Système interactif d'aide au déplacement multi-modal (SIDAM)*. PhD thesis, Ecole centrale de Lille, 2006.

Appendix A

- Servers Table:

Serveurs	
RefServer	ServerName
1	S1
2	S2
3	S3
4	S4
5	S5
6	S6
7	S7
8	S8
9	S9
10	S10
11	S11
12	S12
13	S13
14	S14
15	S15
16	S16
17	S17
18	S18
19	S19
20	S20

- Tasks Table

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
1	1	Travel service	0,5	4	0,2
1	3	Travel service	0,3	2	0,2
1	4	Travel service	0,1	10	0,45
1	5	Travel service	0,8	4	0,15
1	6	Travel service	0,1	4	0,15
1	7	Transport service	0,3	10	0,2

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
1	8	Travel service	0,1	3	0,3
1	9	Transport service	0,8	3	0,15
1	10	Travel service	0,1	10	0,3
1	11	Transport service	0,3	4	0,45
1	12	Travel service	0,4	10	0,35
1	14	Transport service	0,1	10	0,3
1	15	Travel service	0,1	1	0,15
1	17	Transport service	0,3	10	0,25
1	18	Travel service	0,1	3	0,45
2	1	Transport service	0,8	2	0,35
2	3	Transport service	0,2	3	0,25
2	4	Transport service	0,5	4	0,3
2	7	Transport service	0,1	10	0,45
2	8	Transport service	0,1	2	0,3
2	9	Transport service	0,4	6	0,15
2	10	Travel service	0,2	5	0,1
2	12	Travel service	0,3	6	0,15
2	13	Travel service	0,4	3	0,05
2	14	Travel service	0,8	10	0,3
2	15	Travel service	0,4	5	0,2
2	17	Travel service	0,4	6	0,3
2	18	Travel service	0,5	2	0,35
2	19	Travel service	0,4	6	0,05
2	20	Transport service	0,1	5	0,15
3	1	Travel service	0,1	10	0,05
3	2	Travel service	0,3	1	0,1
3	3	Transport service	0,1	4	0,05
3	4	Travel service	0,2	2	0,3
3	5	Travel service	0,1	4	0,15
3	7	Travel service	0,1	1	0,2
3	8	Transport service	0,5	6	0,45
3	9	Transport service	0,1	3	0,25
3	10	Travel service	0,1	6	0,3
3	13	Travel service	0,2	1	0,1
3	14	Transport service	0,1	4	0,05
3	15	Travel service	0,1	3	0,35
3	16	Travel service	0,5	8	0,05

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
3	20	Geographic service	0,1	1	0,3
4	1	Travel service	0,5	6	0,1
4	3	Travel service	0,5	4	0,45
4	4	Travel service	0,5	3	0,2
4	5	Geographic service	0,5	2	0,15
4	7	Travel service	0,5	1	0,05
4	8	Travel service	0,8	6	0,1
4	9	Travel service	0,5	5	0,2
4	10	Travel service	0,3	2	0,15
4	11	Geographic service	0,5	7	0,05
4	12	Travel service	0,4	6	0,25
4	14	Travel service	0,5	3	0,15
4	16	Travel service	0,5	7	0,1
4	17	Geographic service	0,5	8	0,05
4	18	Transport service	0,8	10	0,35
4	19	Geographic service	0,5	6	0,1
4	20	Travel service	0,5	3	0,25
5	1	Travel service	0,2	7	0,05
5	3	Geographic service	0,3	3	0,1
5	4	Travel service	0,2	8	0,2
5	5	Travel service	0,2	7	0,35
5	7	Geographic service	0,5	5	0,15
5	8	Transport service	0,2	7	0,05
5	9	Geographic service	0,2	5	0,1
5	10	Travel service	0,2	3	0,05
5	11	Geographic service	0,7	10	0,25
5	12	Transport service	0,2	9	0,2
5	15	Travel service	0,3	7	0,3
5	16	Geographic service	0,2	5	0,15
5	17	Travel service	0,2	2	0,05
5	18	Travel service	0,7	7	0,4
5	19	Geographic service	0,2	7	0,3
5	20	Geographic service	0,2	5	0,15
6	1	Travel service	0,6	2	0,2
6	3	Travel service	0,3	5	0,15
6	4	Travel service	0,6	8	0,05
6	7	Travel service	0,3	10	0,35

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
6	8	Travel service	0,6	7	0,05
6	9	Transport service	0,6	3	0,2
6	10	Travel service	0,1	6	0,45
6	12	Geographic service	0,2	6	0,15
6	13	Travel service	0,5	8	0,15
6	14	Geographic service	0,5	9	0,05
6	15	Travel service	0,9	3	0,25
6	16	Travel service	0,3	7	0,35
6	17	Geographic service	0,6	8	0,35
6	18	Transport service	0,1	2	0,45
6	20	Travel service	0,2	6	0,25
7	4	Travel service	0,1	7	0,15
7	8	Geographic service	0,3	10	0,15
7	9	Transport service	0,2	5	0,4
7	10	Travel service	0,1	1	0,15
7	11	Geographic service	0,2	4	0,1
7	12	Transport service	0,9	1	0,4
7	13	Travel service	0,5	6	0,1
7	15	Travel service	0,8	8	0,2
7	17	Travel service	0,1	4	0,4
7	20	Geographic service	0,9	1	0,25
8	3	Transport service	0,1	6	0,35
8	4	Travel service	0,3	10	0,15
8	5	Travel service	0,1	9	0,4
8	7	Geographic service	0,6	7	0,05
8	8	Transport service	0,7	2	0,15
8	9	Travel service	0,8	3	0,2
8	10	Travel service	0,3	2	0,45
8	12	Travel service	0,1	1	0,05
8	14	Geographic service	0,7	3	0,35
8	15	Transport service	0,3	9	0,01
8	16	Geographic service	0,1	2	0,45
8	17	Travel service	0,9	1	0,05
8	18	Travel service	0,5	10	0,05
8	19	Travel service	0,7	3	0,3
8	20	Geographic service	0,6	8	0,05
9	1	Transport service	0,7	6	0,2

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
9	3	Geographic service	0,1	3	0,1
9	4	Travel service	0,9	8	0,1
9	5	Travel service	0,2	1	0,05
9	10	Geographic service	0,9	8	0,4
9	11	Transport service	0,9	5	0,2
9	13	Geographic service	0,4	3	0,45
9	15	Travel service	0,8	5	0,05
9	16	Travel service	0,5	10	0,15
9	17	Travel service	0,1	9	0,2
9	18	Geographic service	0,9	8	0,4
9	20	Transport service	0,6	10	0,45
10	1	Travel service	0,1	7	0,2
10	3	Travel service	0,9	6	0,25
10	4	Travel service	0,1	6	0,15
10	5	Geographic service	0,3	4	0,1
10	8	Transport service	0,8	7	0,1
10	9	Travel service	0,4	5	0,05
10	10	Geographic service	0,9	5	0,4
10	11	Travel service	0,4	8	0,15
10	12	Geographic service	0,8	4	0,3
10	15	Travel service	0,7	2	0,3
10	16	Travel service	0,4	6	0,1
10	18	Transport service	0,7	10	0,2
10	19	Geographic service	0,9	7	0,25
11	4	Travel service	0,2	10	0,4
11	7	Travel service	0,6	7	0,05
11	10	Travel service	0,2	10	0,25
11	11	Transport service	0,6	10	0,1
11	12	Geographic service	0,3	3	0,1
11	13	Travel service	0,6	1	0,45
11	14	Travel service	0,5	2	0,15
11	16	Transport service	0,1	5	0,05
11	17	Transport service	0,3	2	0,35
11	18	Travel service	0,6	1	0,05
11	19	Travel service	0,1	4	0,2
11	20	Transport service	0,3	8	0,35
12	5	Travel service	0,3	5	0,3

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
12	7	Transport service	0,5	7	0,15
12	19	Geographic service	0,2	8	0,45
13	1	Travel service	0,2	1	0,5
13	6	Transport service	0,6	2	0,25
13	19	Travel service	0,5	5	0,3
13	20	Transport service	0,9	6	0,9
14	2	Travel service	0,8	1	0,2
14	3	Transport service	0,3	2	0,7
14	4	Transport service	0,6	9	0,9
14	9	Travel service	0,6	10	0,4
14	10	Travel service	0,9	5	0,4
14	19	Geographic service	0,2	6	0,5
14	20	Transport service	0,3	3	0,25
15	1	Travel service	0,8	4	0,1
15	5	Travel service	0,2	6	0,3
16	3	Travel service	0,2	2	0,9
16	4	Geographic service	0,2	3	0,1
16	17	Geographic service	0,2	3	0,5
16	18	Travel service	0,8	8	0,7
16	19	Travel service	0,8	1	0,4
16	20	Geographic service	0,4	8	0,4
24	9	Transport service	0,3	2	0,6
24	19	Travel service	0,2	3	0,55
25	4	Transport service	0,5	5	0,5
25	16	Travel service	0,3	6	0,25
25	17	Travel service	0,2	7	0,3
25	20	Transport service	0,1	9	0,15
26	2	Geographic service	0,1	5	0,4
26	3	Travel service	0,2	9	0,8
26	11	Travel service	0,3	7	0,7
26	12	Geographic service	0,8	5	0,25
26	18	Transport service	0,2	9	0,2
27	18	Travel service	0,3	2	0,5
27	19	Travel service	0,6	1	0,4
27	20	Travel service	0,3	2	0,4
28	3	Travel service	0,9	5	0,25
28	5	Travel service	0,3	4	0,8

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
28	10	Transport service	0,8	2	0,35
28	12	Geographic service	0,2	7	0,25
28	15	Transport service	0,8	4	0,9
28	19	Geographic service	0,2	5	0,85
28	20	Travel service	0,1	6	0,8
29	1	Travel service	0,3	4	0,5
29	2	Travel service	0,3	2	0,4
29	11	Transport service	0,5	3	0,15
29	13	Transport service	0,4	4	0,2
30	17	Geographic service	0,2	10	0,05
30	18	Travel service	0,3	9	0,1
30	19	Travel service	0,3	8	0,25
30	20	Travel service	0,4	7	0,3
31	1	Transport service	0,5	4	0,2
31	3	Geographic service	0,3	2	0,2
31	4	Travel service	0,1	10	0,45
31	6	Travel service	0,1	4	0,15
31	8	Travel service	0,1	3	0,3
31	12	Transport service	0,4	10	0,35
31	14	Geographic service	0,1	10	0,3
31	18	Transport service	0,1	3	0,45
32	1	Travel service	0,8	2	0,35
32	3	Travel service	0,2	3	0,25
32	4	Transport service	0,5	4	0,3
32	7	Travel service	0,1	10	0,45
32	8	Travel service	0,1	2	0,3
32	9	Travel service	0,4	6	0,15
32	12	Travel service	0,3	6	0,15
32	14	Travel service	0,8	10	0,3
32	18	Transport service	0,5	2	0,35
33	1	Geographic service	0,8	5	0,3
33	3	Transport service	0,3	4	0,4
33	4	Transport service	0,8	3	0,45
33	7	Travel service	0,3	11	0,2
33	12	Travel service	0,3	1	0,35
33	16	Travel service	0,8	6	0,15
34	1	Transport service	0,1	10	0,05

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
34	2	Geographic service	0,3	1	0,1
34	3	Transport service	0,1	4	0,05
34	7	Travel service	0,1	1	0,2
35	1	Travel service	0,5	6	0,1
35	3	Travel service	0,5	4	0,45
35	8	Transport service	0,8	6	0,1
35	18	Transport service	0,8	10	0,35
36	1	Travel service	0,2	7	0,05
36	3	Travel service	0,3	3	0,1
36	18	Travel service	0,7	7	0,4
37	3	Transport service	0,6	7	0,45
37	18	Geographic service	0,6	10	0,05
37	20	Travel service	0,6	3	0,1
38	1	Travel service	0,6	2	0,2
38	3	Travel service	0,3	5	0,15
38	10	Transport service	0,1	6	0,45
38	14	Geographic service	0,5	9	0,05
38	17	Travel service	0,6	8	0,35
39	10	Travel service	0,1	1	0,15
39	17	Transport service	0,1	4	0,4
39	20	Geographic service	0,9	1	0,25
40	11	Geographic service	0,8	1	0,05
40	16	Travel service	0,4	1	0,4
40	17	Transport service	0,9	10	0,05
41	3	Travel service	0,1	6	0,35
41	5	Travel service	0,1	9	0,4
41	18	Geographic service	0,5	10	0,05
42	1	Transport service	0,7	6	0,2
42	3	Travel service	0,1	3	0,1
42	11	Travel service	0,9	5	0,2
42	18	Transport service	0,9	8	0,4
43	1	Transport service	0,1	7	0,2
43	3	Geographic service	0,9	6	0,25
43	9	Travel service	0,4	5	0,05
43	10	Travel service	0,9	5	0,4
43	11	Travel service	0,4	8	0,15
43	19	Transport service	0,9	7	0,25

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
44	11	Transport service	0,6	10	0,1
44	17	Travel service	0,3	2	0,35
45	1	Travel service	0,3	7	0,4
45	3	Travel service	0,1	7	0,2
45	8	Transport service	0,2	8	0,25
45	18	Transport service	0,9	2	0,35
46	9	Geographic service	0,4	5	0,2
46	19	Geographic service	0,2	8	0,45
47	1	Travel service	0,2	1	0,5
47	6	Travel service	0,6	2	0,25
47	19	Travel service	0,5	5	0,3
48	2	Transport service	0,8	1	0,2
48	3	Transport service	0,3	2	0,7
48	4	Travel service	0,6	9	0,9
48	19	Travel service	0,2	6	0,5
49	1	Transport service	0,8	4	0,1
50	3	Geographic service	0,3	5	0,2
50	15	Travel service	0,8	3	0,1
51	3	Travel service	0,2	2	0,9
51	18	Travel service	0,8	8	0,7
52	17	Geographic service	0,2	3	0,5
52	19	Transport service	0,8	1	0,4
53	1	Travel service	0,5	1	0,9
53	13	Travel service	0,2	9	0,5
53	16	Travel service	0,1	5	0,8
54	19	Geographic service	0,2	3	0,55
55	16	Transport service	0,3	6	0,25
55	17	Transport service	0,2	7	0,3
56	2	Travel service	0,1	5	0,4
56	3	Travel service	0,2	9	0,8
56	11	Travel service	0,3	7	0,7
56	18	Transport service	0,2	9	0,2
57	18	Geographic service	0,3	2	0,5
57	19	Travel service	0,6	1	0,4
58	2	Transport service	0,3	9	0,9
58	3	Travel service	0,9	5	0,25
58	10	Transport service	0,8	2	0,35

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
58	15	Travel service	0,8	4	0,9
58	19	Travel service	0,2	5	0,85
59	1	Travel service	0,3	4	0,5
59	2	Transport service	0,3	2	0,4
59	11	Transport service	0,5	3	0,15
59	13	Transport service	0,4	4	0,2
60	17	Geographic service	0,2	10	0,05
60	18	Transport service	0,3	9	0,1
60	19	Transport service	0,3	8	0,25
61	3	Transport service	0,1	2	0,1
61	15	Geographic service	0,1	1	0,25
61	17	Transport service	0,2	10	0,25
62	8	Geographic service	0,1	2	0,3
62	14	Travel service	0,8	10	0,3
62	18	Travel service	0,5	2	0,15
63	1	Geographic service	0,8	5	0,3
63	3	Travel service	0,3	4	0,4
63	4	Geographic service	0,8	3	0,25
63	5	Travel service	0,5	7	0,05
63	7	Travel service	0,3	2	0,2
63	8	Travel service	0,8	7	0,15
63	9	Transport service	0,8	1	0,15
63	16	Transport service	0,8	6	0,15
63	18	Geographic service	0,5	5	0,35
64	1	Transport service	0,2	10	0,15
64	2	Travel service	0,3	1	0,1
64	3	Travel service	0,1	4	0,05
64	4	Travel service	0,2	2	0,3
64	5	Transport service	0,1	4	0,15
64	9	Travel service	0,1	3	0,25
65	1	Travel service	0,5	6	0,1
65	14	Transport service	0,4	3	0,15
65	18	Transport service	0,8	10	0,35
66	1	Travel service	0,2	7	0,05
66	3	Travel service	0,1	3	0,1
66	15	Travel service	0,3	8	0,2
67	3	Transport service	0,6	7	0,45

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
67	13	Transport service	0,6	6	0,3
67	19	Travel service	0,6	7	0,25
68	1	Travel service	0,6	2	0,2
68	3	Travel service	0,3	5	0,1
68	6	Geographic service	0,3	5	0,15
68	8	Transport service	0,6	7	0,05
68	17	Transport service	0,6	8	0,25
69	9	Transport service	0,2	5	0,4
69	11	Transport service	0,2	4	0,1
70	9	Geographic service	0,9	8	0,15
70	11	Transport service	0,8	1	0,05
70	14	Transport service	0,1	3	0,1
71	3	Geographic service	0,7	6	0,35
71	7	Transport service	0,4	7	0,05
71	14	Transport service	0,3	3	0,25
72	1	Travel service	0,8	6	0,2
72	3	Travel service	0,2	3	0,1
72	10	Travel service	0,5	8	0,4
72	13	Travel service	0,5	3	0,15
72	18	Transport service	0,5	8	0,4
73	15	Transport service	0,7	2	0,3
73	16	Geographic service	0,4	2	0,1
73	19	Transport service	0,9	7	0,25
74	7	Geographic service	0,6	7	0,05
74	14	Travel service	0,5	2	0,25
74	16	Travel service	0,1	5	0,15
75	1	Travel service	0,3	7	0,4
75	3	Geographic service	0,1	7	0,2
75	10	Transport service	0,8	1	0,05
75	12	Geographic service	0,9	2	0,25
76	2	Transport service	0,1	5	0,4
76	3	Geographic service	0,2	9	0,8
76	18	Travel service	0,2	9	0,25
77	18	Travel service	0,3	2	0,5
77	20	Transport service	0,3	2	0,2
78	2	Geographic service	0,3	9	0,8
78	12	Transport service	0,3	7	0,35

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
79	1	Transport service	0,5	4	0,5
79	2	Travel service	0,2	2	0,4
80	17	Travel service	0,2	10	0,05
80	20	Travel service	0,1	7	0,3
81	1	Transport service	0,5	4	0,2
81	3	Transport service	0,4	2	0,2
81	4	Geographic service	0,2	10	0,45
81	12	Travel service	0,3	10	0,15
81	18	Travel service	0,1	3	0,45
82	1	Transport service	0,8	2	0,35
82	3	Transport service	0,2	3	0,25
82	7	Travel service	0,1	5	0,45
82	8	Transport service	0,1	2	0,3
82	9	Travel service	0,4	6	0,15
82	14	Travel service	0,5	2	0,3
83	5	Transport service	0,2	7	0,15
83	7	Transport service	0,9	3	0,2
83	12	Transport service	0,3	1	0,35
83	14	Travel service	0,3	5	0,25
84	1	Geographic service	0,1	10	0,05
84	2	Transport service	0,3	1	0,1
84	3	Travel service	0,1	4	0,05
85	1	Travel service	0,5	6	0,1
85	3	Travel service	0,5	4	0,45
85	18	Transport service	0,8	10	0,35
86	1	Transport service	0,2	7	0,05
86	3	Transport service	0,1	3	0,1
86	5	Travel service	0,2	7	0,15
86	18	Travel service	0,7	7	0,45
87	3	Travel service	0,6	7	0,45
87	20	Geographic service	0,2	3	0,15
89	12	Transport service	0,9	1	0,4
89	15	Travel service	0,7	8	0,3
89	17	Travel service	0,3	4	0,4
90	9	Travel service	0,5	8	0,05
90	14	Transport service	0,1	3	0,1
90	16	Geographic service	0,4	1	0,4

Tasks					
RefTask	RsfServeur	Labels	Time	Cost	Size
90	17	Transport service	0,3	10	0,05
91	6	Transport service	0,1	9	0,4
91	18	Travel service	0,5	5	0,05
91	19	Travel service	0,7	3	0,3
92	3	Travel service	0,1	3	0,1
92	13	Geographic service	0,4	3	0,45
92	17	Transport service	0,1	9	0,2
92	18	Travel service	0,9	8	0,4
93	1	Travel service	0,1	7	0,2
93	3	Travel service	0,9	6	0,15
93	5	Transport service	0,3	4	0,1
93	9	Transport service	0,4	5	0,05
94	11	Travel service	0,6	10	0,15
94	12	Travel service	0,3	3	0,1
94	13	Geographic service	0,6	1	0,35
94	14	Transport service	0,5	2	0,15
94	18	Transport service	0,6	1	0,05
95	5	Travel service	0,6	8	0,35
95	9	Travel service	0,5	9	0,2
95	11	Transport service	0,3	4	0,35
95	13	Travel service	0,2	2	0,3
96	19	Transport service	0,3	8	0,35
97	1	Geographic service	0,1	1	0,5
97	19	Transport service	0,5	5	0,3
98	2	Travel service	0,8	2	0,2
98	3	Travel service	0,3	2	0,7
98	9	Transport service	0,6	10	0,4
98	20	Transport service	0,3	2	0,25
99	1	Geographic service	0,8	4	0,15
99	5	Travel service	0,2	6	0,3
100	3	Travel service	0,3	4	0,1
100	5	Transport service	0,3	3	0,5
100	20	Travel service	0,2	2	0,5

- Latencies

Latencies		
RefServer1	RefServer2	SL
0	1	0,25
0	2	0,13
0	3	0,71
0	4	0,49
0	5	0,39
0	6	0,4
0	7	0,19
0	8	0,79
0	9	5,29
0	10	0,52
0	11	0,67
0	12	0,88
0	13	0,89
0	14	0,25
0	15	0,85
0	16	0,2
0	17	0,89
0	18	0,11
0	19	0,38
0	20	0,02
1	2	0,25
1	3	0,36
1	4	0,46
1	5	0,46
1	6	0,68
1	7	0,28
1	8	0,13
1	9	0,14
1	10	0,82
1	11	0,49
1	12	0,87
1	13	0,41
1	14	0,02
1	15	0,1
1	16	0,79

Latencies		
RefServer1	RefServer2	SL
1	17	0,95
1	18	0,94
1	19	0,11
1	20	0,34
2	3	0,98
2	4	0,61
2	5	0,76
2	6	0,74
2	7	0,82
2	8	0,44
2	9	0,28
2	10	0,98
2	11	0,26
2	12	0,81
2	13	0,17
2	14	0,71
2	15	0,42
2	16	0,44
2	17	0,52
2	18	0,21
2	19	0,96
2	20	0,97
3	4	0,16
3	5	0,28
3	6	0,16
3	7	0,93
3	8	0,49
3	9	0,04
3	10	0,9
3	11	0,82
3	12	0,74
3	13	0,53
3	14	0,96
3	15	0,13
3	16	0,16
3	17	0,1
3	18	0,04

Latencies		
RefServer1	RefServer2	SL
3	19	0,16
3	20	0,25
4	5	0,95
4	6	0,11
4	7	0,46
4	8	0,48
4	9	0,5
4	10	0,76
4	11	0,17
4	12	0,23
4	13	0,89
4	14	0,27
4	15	0,14
4	16	0,29
4	17	0,33
4	18	0,74
4	19	0,31
4	20	0,24
5	6	0,16
5	7	0,9
5	8	0,34
5	9	0,33
5	10	0,89
5	11	0,36
5	12	0,31
5	13	0,58
5	14	0,17
5	15	0,15
5	16	0,38
5	17	0,1
5	18	0,45
5	19	0,75
5	20	0,13
6	7	0,35
6	8	0,5
6	9	0,37
6	10	0,34

Latencies		
RefServer1	RefServer2	SL
6	11	0,51
6	12	0,68
6	13	0,6
6	14	0,39
6	15	0,87
6	16	0,3
6	17	0,62
6	18	0,61
6	19	0,19
6	20	0,51
7	8	0,71
7	9	0,28
7	10	0,09
7	11	0,69
7	12	0,96
7	13	0,01
7	14	0,56
7	15	0,52
7	16	0,51
7	17	0,21
7	18	0,1
7	19	0,84
7	20	0,64
8	9	0,04
8	10	0,67
8	11	0,72
8	12	0,42
8	13	0,4
8	14	0,2
8	15	0,33
8	16	0,46
8	17	0,15
8	18	0,17
8	19	0,73
8	20	0,41
9	10	0,7
9	11	0,78

Latencies		
RefServer1	RefServer2	SL
9	12	0,85
9	13	0,73
9	14	0,54
9	15	0,12
9	16	0,31
9	17	0,5
9	18	0,63
9	19	0,37
9	20	0,49
10	11	0,27
10	12	0,34
10	13	0,88
10	14	0,15
10	15	0,58
10	16	0,08
10	17	0,2
10	18	0,23
10	19	0,14
10	20	0,86
11	12	0,16
11	13	0,89
11	14	0,49
11	15	0,74
11	16	0,58
11	17	0,08
11	18	0,7
11	19	0,47
11	20	0,4
12	13	0,37
12	14	0,36
12	15	0,47
12	16	0,41
12	17	0,12
12	18	0,59
12	19	0,06
12	20	0,69
13	14	0,98

Latencies		
RefServer1	RefServer2	SL
13	15	0,49
13	16	0,88
13	17	0,2
13	18	0,16
13	19	0,19
13	20	0,47
14	15	0,2
14	16	0,16
14	17	0,66
14	18	0,2
14	19	0,9
14	20	0,32
15	16	0,16
15	17	0,96
15	18	0,18
15	19	0,94
15	20	0,1
16	17	0,84
16	18	0,81
16	19	0,96
16	20	0,57
17	18	0,14
17	19	0,81
17	20	0,4
18	19	0,72
18	20	0,6
19	20	0,11