



# Personalizable architecture model for optimizing the access to pervasive ressources and services: Application in telemedicine

Ebrahim Nageba

## ► To cite this version:

Ebrahim Nageba. Personalizable architecture model for optimizing the access to pervasive ressources and services: Application in telemedicine. Other [cs.OH]. INSA de Lyon, 2011. English. NNT: 2011ISAL0128 . tel-00694445

**HAL Id: tel-00694445**

**<https://theses.hal.science/tel-00694445>**

Submitted on 4 May 2012

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

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



Thèse

**Personalizable Architecture Model for Optimizing the  
Access to Pervasive Ressources and Services.**

**Application in Telemedicine**

**Modèle d'architecture personnalisable pour l'optimisation de l'accès à  
des ressources et services pervasifs. Application à la télémédecine**

Présentée devant  
L'Institut National des Sciences Appliquées de Lyon

Pour obtenir  
Le grade de docteur

Formation Doctorale  
INFORMATIQUE

ÉCOLE DOCTORALE  
INFORMATIQUE ET MATHÉMATIQUES

Par  
Ebrahim Nageba

SOUTENUE LE 07 DECEMBRE 2011 DEVANT LA COMMISSION D'EXAMEN

Jury MM.

---

|                   |   |                       |
|-------------------|---|-----------------------|
| Jacques DUCHENE   | Professeur, Institut Charles Delaunay, STMR,<br>Université de Technologie de Troyes | Président du jury     |
| Jocelyne FAYN     | Ingénieur de Recherche, HDR, SFR Santé Lyon<br>Est, Bron                            | Co-directeur de thèse |
| Franck MORVAN     | Professeur, IRIT, Université Paul Sabatier,<br>Toulouse                             | Rapporteur            |
| Paul RUBEL        | Professeur Emérite, MTIC, INSA de Lyon  | Directeur de thèse    |
| Christine VERDIER | Professeur, LIG, Université Joseph Fourier,<br>Grenoble                             | Rapporteur            |

---

Laboratoire MTIC (EA4171) & SFR Santé Lyon Est (US7-UMS3453)



---

## **Modèle d'architecture personnalisable pour l'optimisation de l'accès à des ressources et services pervasifs. Application à la télémédecine**

---

### **Résumé**

Le développement et l'usage croissants de systèmes pervasifs, dotés de fonctionnalités et de moyens de communication de plus en plus sophistiqués, offrent de fantastiques potentialités de services, en particulier pour l'e-Santé et la télémédecine, au bénéfice de tout citoyen, patient ou professionnel de santé. L'un des challenges sociétaux actuels est de permettre une meilleure exploitation des services disponibles pour l'ensemble des acteurs impliqués dans un domaine donné. Mais la multiplicité des services offerts, la diversité fonctionnelle des systèmes, et l'hétérogénéité des besoins nécessitent l'élaboration de modèles de connaissances de ces services, des fonctions de ces systèmes et des besoins. En outre, l'hétérogénéité des environnements informatiques distribués, la disponibilité et les capacités potentielles des diverses ressources humaines et matérielles (instrumentation, services, sources de données, etc.) requises par les différentes tâches et processus, la variété des services qui fournissent des données aux utilisateurs, et les conflits d'interopérabilité entre schémas et sources de données sont autant de problématiques que nous avons à considérer au cours de nos travaux de recherche.

Notre contribution vise à optimiser la qualité de services en environnement ambiant et à réaliser une exploitation intelligente de ressources ubiquitaires. Pour cela, nous proposons un méta-modèle de connaissances des principaux concepts à prendre en compte en environnement pervasif : Acteur, Tâche, Ressource, Objet, Service, Localisation, Organisation, etc. Ce méta-modèle est basé sur des ontologies décrivant les différentes entités précitées dans un domaine donné ainsi que leurs relations. Puis, nous l'avons formalisé en utilisant un langage standard de description des connaissances. A partir de ce modèle, nous proposons alors une nouvelle méthodologie de construction d'un framework architectural, que nous avons appelé ONOF-PAS (ONtology Oriented Framework for Pervasive Applications and Services). ONOF-PAS est basé sur des modèles ontologiques, une base de règles, un moteur

d'inférence, et des composants orientés objet permettant la gestion des différentes tâches et le traitement des ressources. Il s'agit d'une architecture générique, applicable à différents domaines. ONOF-PAS a la capacité d'effectuer un raisonnement à base de règles pour gérer les différents contextes d'utilisation et aider à la prise de décision dans des environnements hétérogènes dynamiques, tout en tenant compte de la disponibilité et de la capacité des ressources humaines et matérielles requises par les diverses tâches et processus exécutés par des systèmes d'information pervasifs. Enfin, nous avons instancié ONOF-PAS dans le domaine de la télémédecine pour traiter le scénario de l'orientation des patients ou de personnes victimes de problèmes de santé en environnement hostile telles que la haute montagne ou des zones géographiquement isolées. Un prototype d'implémentation de ces scénarios, appelé T-TROIE (Telemedicine Tasks and Resources Ontologies for Inimical Environments), a été développé afin de valider notre approche et le framework ONOF-PAS que nous avons proposé.

Le résultat attendu de notre contribution est d'aboutir à une communication efficace des informations disponibles, procurant une qualité de services de télémédecine qui permet d'assurer une réponse prompte et fiable à chaque acteur demandeur en lui fournissant des données exhaustives et pertinentes à sa demande, et par conséquence, d'obtenir une prise en charge appropriée, plus rapide, et optimale des patients, quel que soit l'endroit où ils se trouvent, en fonction des moyens et des technologies disponibles.

**Mots Clés:** Systèmes Pervasifs, Contexte-Awareness, Modélisation des Connaissances, Ontologie, Raisonnement à base des règles, Systèmes d'aide à la décision, Télémédecine, eSanté.

---

## **Personalizable Architecture Model for Optimizing the Access to Pervasive Resources and Services. Application in Telemedicine.**

---

### **Abstract**

The growing development and use of pervasive systems, equipped with increasingly sophisticated functionalities and communication means, offer fantastic potentialities of services, particularly in the eHealth and Telemedicine domains, for the benefit of each citizen, patient or healthcare professional. One of the current societal challenges is to enable a better exploitation of the available services for all actors involved in a given domain. Nevertheless, the multiplicity of the offered services, the systems functional variety, and the heterogeneity of the needs require the development of knowledge models of these services, systems functions, and needs. In addition, the distributed computing environments heterogeneity, the availability and potential capabilities of various human and material resources (devices, services, data sources, etc.) required by the different tasks and processes, the variety of services providing users with data, the interoperability conflicts between schemas and data sources are all issues that we have to consider in our research works.

Our contribution aims to empower the intelligent exploitation of ubiquitous resources and to optimize the quality of service in ambient environment. For this, we propose a knowledge meta-model of the main concepts of a pervasive environment, such as Actor, Task, Resource, Object, Service, Location, Organization, etc. This knowledge meta-model is based on ontologies describing the different aforementioned entities from a given domain and their interrelationships. We have then formalized it by using a standard language for knowledge description. After that, we have designed an architectural framework called ONOF-PAS (ONtology Oriented Framework for Pervasive Applications and Services) mainly based on ontological models, a set of rules, an inference engine, and object oriented components for tasks management and resources processing. Being generic, extensible, and applicable in different domains, ONOF-PAS has the ability to perform rule-based reasoning to handle various contexts of use and enable decision making in dynamic and heterogeneous environments while taking into account the availability and capabilities of

the human and material resources required by the multiples tasks and processes executed by pervasive systems. Finally, we have instantiated ONOF-PAS in the telemedicine domain to handle the scenario of the transfer of persons victim of health problems during their presence in hostile environments such as high mountains resorts or geographically isolated areas. A prototype implementing this scenario, called T-TROIE (Telemedicine Tasks and Resources Ontologies for Inimical Environments), has been developed to validate our approach and the proposed ONOF-PAS framework.

The final objective of our contribution is to achieve an efficient communication of the available information, enabling high quality telemedicine services that ensure quick and reliable answers to every requesting actor by providing him with data that are relevant to his request, and consequently, to obtain an appropriate and optimal patients' medical assistance, anywhere and anytime, depending on the means and the available technologies.

**Keywords:** Pervasive Computing, Context Modelling, Knowledge Modelling, Ontology, Rule-based Reasoning, Decision Support Systems, Telemedicine, eHealth.

| SIGLE            | ECOLE DOCTORALE  | NOM ET COORDONNEES DU RESPONSABLE  |
|------------------|--|--|
| <b>CHIMIE</b>    | <b>CHIMIE DE LYON</b><br><a href="http://www.edchimie-lyon.fr">http://www.edchimie-lyon.fr</a><br><br>Insa : R. GOURDON  | <b>M. Jean Marc LANCELIN</b><br>Université de Lyon – Collège Doctoral<br>Bât ESCPE<br>43 bd du 11 novembre 1918<br>69622 VILLEURBANNE Cedex<br>Tél : 04.72.43 13 95<br><a href="mailto:directeur@edchimie-lyon.fr">directeur@edchimie-lyon.fr</a>                                  |
| <b>E.E.A.</b>    | <b>ELECTRONIQUE, ELECTROTECHNIQUE, AUTO-MATIQUE</b><br><a href="http://edeea.ec-lyon.fr">http://edeea.ec-lyon.fr</a><br><br>Secrétariat : M.C. HAVGOUDOUKIAN<br><a href="mailto:eea@ec-lyon.fr">eea@ec-lyon.fr</a> | <b>M. Gérard SCORLETTI</b><br>Ecole Centrale de Lyon<br>36 avenue Guy de Collongue<br>69134 ECULLY<br>Tél : 04.72.18 60 97 Fax : 04 78 43 37 17<br><a href="mailto:Gerard.scorletti@ec-lyon.fr">Gerard.scorletti@ec-lyon.fr</a>  |
| <b>E2M2</b>      | <b>EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION</b><br><a href="http://e2m2.universite-lyon.fr">http://e2m2.universite-lyon.fr</a><br><br>Insa : H. CHARLES  | <b>Mme Gudrun BORNETTE</b><br>CNRS UMR 5023 LEHNA<br>Université Claude Bernard Lyon 1<br>Bât Forel<br>43 bd du 11 novembre 1918<br>69622 VILLEURBANNE Cédex<br>Tél : 04.72.43.12.94<br><a href="mailto:e2m2@biomserv.univ-lyon1.fr">e2m2@biomserv.univ-lyon1.fr</a>                |
| <b>EDISS</b>     | <b>INTERDISCIPLINAIRE SCIENCES-SANTE</b><br><a href="http://ww2.ibcp.fr/ediss">http://ww2.ibcp.fr/ediss</a><br><br>Sec : Safia AIT CHALAL<br><br>Insa : M. LAGARDE   | <b>M. Didier REVEL</b><br>Hôpital Louis Pradel<br>Bâtiment Central<br>28 Avenue Doyen Lépine<br>69677 BRON<br>Tél : 04.72.68 49 09 Fax : 04 72 35 49 16<br><a href="mailto:Didier.revel@creatis.uni-lyon1.fr">Didier.revel@creatis.uni-lyon1.fr</a>                                |
| <b>INFOMATHS</b> | <b>INFORMATIQUE ET MATHEMATIQUES</b><br><a href="http://infomaths.univ-lyon1.fr">http://infomaths.univ-lyon1.fr</a>  | <b>M. Johannes KELLENDONK</b><br>Université Claude Bernard Lyon 1<br>INFOMATHS<br>Bâtiment Braconnier<br>43 bd du 11 novembre 1918<br>69622 VILLEURBANNE Cedex<br>Tél : 04.72. 44.82.94 Fax 04 72 43 16 87<br><a href="mailto:infomaths@univ-lyon1.fr">infomaths@univ-lyon1.fr</a> |
| <b>Matériaux</b> | <b>MATERIAUX DE LYON</b><br><br>Secrétariat : M. LABOUNE<br>PM : 71.70 –Fax : 87.12<br>Bat. Saint Exupéry<br><a href="mailto:Ed.materiaux@insa-lyon.fr">Ed.materiaux@insa-lyon.fr</a>                              | <b>M. Jean-Yves BUFFIERE</b><br>INSA de Lyon<br>MATEIS<br>Bâtiment Saint Exupéry<br>7 avenue Jean Capelle<br>69621 VILLEURBANNE Cédex<br>Tél : 04.72.43 83 18 Fax 04 72 43 85 28<br><a href="mailto:Jean-yves.buffiere@insa-lyon.fr">Jean-yves.buffiere@insa-lyon.fr</a>           |
| <b>MEGA</b>      | <b>MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE</b><br><br>Secrétariat : M. LABOUNE<br>PM : 71.70 –Fax : 87.12<br>Bat. Saint Exupéry<br><a href="mailto:mega@insa-lyon.fr">mega@insa-lyon.fr</a>                | <b>M. Philippe BOISSE</b><br>INSA de Lyon<br>Laboratoire LAMCOS<br>Bâtiment Jacquard<br>25 bis avenue Jean Capelle<br>69621 VILLEURBANNE Cedex<br>Tél : 04.72.43.71.70 Fax : 04 72 43 72 37<br><a href="mailto:Philippe.boisse@insa-lyon.fr">Philippe.boisse@insa-lyon.fr</a>      |
| <b>ScSo</b>      | <b>ScSo*</b><br><br><b>M. OBADIA Lionel</b><br><br>Sec : Viviane POLSINELLI<br>Insa : J.Y. TOUSSAINT   | <b>M. OBADIA Lionel</b><br>Université Lyon 2<br>86 rue Pasteur<br>69365 LYON Cedex 07<br>Tél : 04.78.69.72.76 Fax : 04.37.28.04.48<br><a href="mailto:Lionel.Obadia@univ-lyon2.fr">Lionel.Obadia@univ-lyon2.fr</a>   |

\*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie





*À mes très chers parents,  
J'aurais tellement aimé que vous soyez à mes côtés ...  
Sans vous je n'aurais jamais été ce que je suis ...*

*Maman,  
Tu as beaucoup souffert de ta maladie. De loin, je partageais sans répit ta souffrance. Même lors de la rédaction de cette thèse, je n'ai pas cessé de penser à toi. J'aurais voulu que ton cœur continue à battre pour toujours. Tu es partie, mais ton image, ta voix, et ton esprit m'accompagneront tout au long du chemin de ma vie.*

*Papa,  
Je garderai le souvenir de ta fierté et de ta confiance en moi.*

*À mes frères et sœurs Farouque, Faten, Fatema, Dïaa, et Alaa*

*À ma femme, Nour*

*À mon grand, Adam*

*À tous,*

*je dédie ce travail*

## Remerciements

Cette thèse a bénéficié d'une bourse du gouvernement Syrien et a été réalisée au sein du laboratoire MTIC commun à l'Université Lyon 1, l'INSA de Lyon et l'INSERM. Grâce à cette opportunité, j'ai pu découvrir le monde de la recherche scientifique et exercer le métier d'enseignant-chercheur au sein de l'Université de Lyon 1 et de l'INSA de Lyon.

J'exprime ma plus sincère reconnaissance à Mr. Paul Rubel et Mme. Jocelyne Fayn qui ont dirigé ma thèse. Leurs conseils avisés ainsi que la motivation et la confiance qu'ils m'ont prodigués me furent très précieux pour mener à bien cette thèse.

Je tiens également à remercier :

Mr. Franck Morvan, Professeur à l'IRIT, Université Paul Sabatier de Toulouse, et Mme Christine Verdier, Professeur au LIG, Université Joseph Fourier de Grenoble d'avoir accepté d'évaluer ma thèse. J'ai particulièrement apprécié leurs rapports détaillés et approfondis témoignant de l'intérêt porté à mon travail.

Mr. Jacques Duchêne, Professeur à l'Université de Technologie de Troyes, d'avoir accepté de participer au jury en tant que Président.

Toutes les personnes avec lesquelles j'ai travaillé, toute l'équipe de MTIC, l'équipe du projet PAIRSE, l'équipe SOC au LIRIS, les enseignants-chercheurs du Département Informatique de l'Université Claude Bernard Lyon 1, l'IUT A, et l'IUT B.

Mes amis et mes collègues pour les bons moments passés ensemble.

Tous les membres de ma famille, notamment ma femme Nour, pour leur encouragement, leur soutien, leur confiance, et leur amour indéfectibles.

# Table of Contents

|   |           |
|---|-----------|
| <b>Chapter I Introduction.....</b>  | <b>21</b> |
| I.1 Context and Motivation .....  | 21        |
| I.2 Scenario Example in Telemedicine.....                                     | 23        |
| I.2.1 A Real Episode.....   | 24        |
| I.2.2 Patient Orientation Scenario Description .....                          | 24        |
| I.3 Challenges and Research Questions .....                                   | 26        |
| I.4 Goals.....  | 28        |
| I.5 Document Structure.....   | 29        |
| <b>Chapter II Background.....</b>   | <b>33</b> |
| II.1 Telemedicine .....   | 33        |
| II.1.1 What is Telemedicine?.....   | 33        |
| II.1.2 eHealth & pHealth.....   | 35        |
| II.1.3 Personal, Shared, and Distributed Electronic Health Record (EHR) ..... | 36        |
| II.2 Pervasive Computing.....   | 37        |
| II.3 The Context-awareness Paradigm.....                                      | 39        |
| II.4 Semantic Web .....   | 40        |
| II.5 Knowledge Representation .....   | 41        |
| II.5.1 Ontology .....   | 42        |
| II.5.2 Ontology Types .....   | 43        |
| II.5.3 Ontology Description Languages.....                                    | 44        |
| II.5.3.1 OWL & OWL 2 .....  | 45        |
| II.5.3.2 OWL sublanguages .....   | 46        |
| II.5.4 Description Logic Based Reasoning.....                                 | 47        |
| II.6 Conclusion.....  | 49        |
| <b>Chapter III State of the Art.....</b>                                      | <b>53</b> |
| III.1 Telemedicine Systems.....   | 53        |
| III.1.1 Introduction.....   | 53        |
| III.1.2 Data Types and Standards in Telemedicine .....                        | 53        |
| III.1.3 Telemedicine Systems Modeling Approaches and Applications.....        | 55        |
| III.1.3.1 Multi-Agent based Telemedicine Systems .....                        | 55        |
| III.1.3.2 Mobile Personal Health Systems.....                                 | 56        |

|                   |   |           |
|-------------------|---|-----------|
| III.1.3.3         | Other Web-based Telemedicine Systems and Applications.....                        | 58        |
| III.1.4           | Ontologies and Telemedicine.....  | 61        |
| III.1.5           | Telemedicine Networks Quality of Service .....                                    | 62        |
| III.1.6           | Analysis of Telemedicine Systems.....   | 64        |
| III.2             | Context Management, Modelling and Computing.....                                  | 66        |
| III.2.1           | Ontology-based Context Modelling Approaches.....                                  | 66        |
| III.2.1.1         | Service Oriented Context-aware Middleware.....                                    | 66        |
| III.2.1.2         | Context Broker Architecture.....  | 67        |
| III.2.1.3         | Context Adaptation and Semantic Interoperability .....                            | 68        |
| III.2.1.4         | Other Context Modeling Approaches .....   | 68        |
| III.2.2           | Current Context-awareness in Healthcare and Telemedicine Applications .....       | 70        |
| III.2.3           | Analysis of Context Research Works .....  | 73        |
| III.3             | Task/Activity-Based Computing.....  | 74        |
| III.4             | Discussion.....   | 77        |
| III.5             | Conclusion .....  | 79        |
| <b>Chapter IV</b> | <b>ONOF-PAS Framework Methodology, Construction, and Formalization.....</b>       | <b>83</b> |
| IV.1              | Rationale and Scope of the Problem .....  | 83        |
| IV.2              | Pervasive Tasks and Contextual Situation Definition.....                          | 84        |
| IV.2.1            | Pervasive Tasks .....   | 84        |
| IV.2.2            | Contextual Situation .....  | 86        |
| IV.3              | Knowledge Integration for Enabling Decision Making Processes .....                | 87        |
| IV.4              | ONOF-PAS Design Methodology .....   | 89        |
| IV.5              | Knowledge Meta-Model Definition .....   | 91        |
| IV.5.1            | General Concepts Identification.....  | 91        |
| IV.5.2            | Meta-Model Formalization Using OWL .....  | 93        |
| IV.5.2.1          | Why Ontologies? .....   | 93        |
| IV.5.2.2          | Ontologies Life Cycle .....   | 94        |
| IV.6              | Ontological Models instantiation and Description in the Telemedicine Domain ..... | 96        |
| IV.6.1            | Telemedicine Domain Analysis and Concepts Extraction.....                         | 96        |
| IV.6.2            | Telemedicine Ontological Models Instantiation.....                                | 97        |
| IV.6.3            | Telemedicine Ontological Models Description .....                                 | 98        |
| IV.6.3.1          | Actor Ontological Model.....  | 98        |
| IV.6.3.2          | Task Ontological Model .....  | 99        |
| IV.6.3.3          | Resources Ontological Model .....   | 100       |
| IV.6.3.4          | Data Objects Ontological Model .....  | 102       |
| IV.6.3.5          | Example of Task Model Instantiation in Telemedicine.....                          | 102       |
| IV.6.4            | Formal Representation of Ontological Models.....                                  | 104       |
| IV.6.4.1          | Ontology Description Language Selection.....                                      | 104       |

|                   |  |            |
|-------------------|--|------------|
| IV.6.4.2          | Ontological Models Encoding .....                                | 105        |
| IV.6.4.3          | Restrictions Definition .....                                    | 108        |
| IV.7              | Rules Description .....  | 109        |
| IV.8              | Conclusion .....   | 113        |
| <b>Chapter V</b>  | <b>Task-centred ONOF-PAS Architecture .....</b>                  | <b>117</b> |
| V.1               | Introduction .....   | 117        |
| V.2               | Actor/System Interactions in the Patient Transfer Scenario ..... | 117        |
| V.3               | ONOF-PAS Architecture Design Requirements .....                  | 119        |
| V.3.1             | Genericity .....   | 120        |
| V.3.2             | Interoperability .....   | 120        |
| V.3.3             | Scalability .....  | 121        |
| V.3.4             | Reusability .....  | 121        |
| V.4               | Task-centred ONOF-PAS Architecture .....                         | 121        |
| V.4.1             | Why Task Centred ? .....   | 121        |
| V.4.2             | ONOF-PAS Architecture Overview .....                             | 122        |
| V.5               | Tasks and Resources Management .....                             | 124        |
| V.5.1             | Profile Manager .....  | 124        |
| V.5.2             | Tasks Manager and Execution Algorithm .....                      | 125        |
| V.5.3             | SPARQL Requestor .....   | 127        |
| V.5.4             | Heterogeneous Resources Processing .....                         | 129        |
| V.5.4.1           | Mapping to ONOF-PAS Ontologies .....                             | 129        |
| V.5.4.2           | Using Web Services for Resources Data Updating .....             | 133        |
| V.5.4.3           | Discussion about Resources Data Update Services Discovery .....  | 136        |
| V.5.4.3.1         | Capability Representation .....                                  | 137        |
| V.5.4.3.2         | Matching Services Capabilities with Requestor Needs .....        | 138        |
| V.5.4.4           | SPARQL Queries Based Resources Filtering .....                   | 139        |
| V.6               | Messaging .....  | 142        |
| V.6.1             | Messages Management Model .....                                  | 142        |
| V.6.2             | Messages Transmission Policy .....                               | 144        |
| V.6.2.1           | Messages Prioritization .....                                    | 144        |
| V.6.2.2           | Messages Stratification .....                                    | 145        |
| V.7               | Conclusion .....   | 147        |
| <b>Chapter VI</b> | <b>ONOF-PAS System Implementation .....</b>                      | <b>151</b> |
| VI.1              | Development Tools .....  | 151        |
| VI.1.1            | Ontology Development Tools .....                                 | 151        |
| VI.1.1.1          | Protégé Integrated Development Environment .....                 | 151        |
| VI.1.1.2          | SWRLTab .....  | 152        |
| VI.1.1.3          | Inference Engines .....  | 153        |

|                    |  |            |
|--------------------|--|------------|
| VI.1.2             | Prototype Application Implementation Tools .....                   | 154        |
| VI.1.2.1           | Java .....   | 154        |
| VI.1.2.2           | Jena API .....   | 154        |
| VI.2               | ONOF-PAS Ontologies Editing .....                                  | 155        |
| VI.3               | Task Execution Example Using Protégé .....                         | 160        |
| VI.3.1             | Patient Orientation Task Scenario .....                            | 160        |
| VI.3.2             | Task Description .....   | 161        |
| VI.3.3             | Resources Data Filtering Using the SPARQL Language .....           | 167        |
| VI.4               | TTROIE Prototype .....   | 169        |
| VI.4.1             | Different TTROIE Use Cases .....                                   | 170        |
| VI.4.2             | TTROIE Software Architecture .....                                 | 171        |
| VI.4.3             | Task Activities Diagram .....                                      | 174        |
| VI.4.4             | Telemedicine Task Execution Algorithm .....                        | 175        |
| VI.4.5             | TTROIE User Interface .....  | 176        |
| VI.5               | Conclusion .....   | 178        |
| <b>Chapter VII</b> | <b>Conclusion and Perspectives .....</b>                           | <b>181</b> |
| VII.1              | General Remarks .....  | 181        |
| VII.2              | Conclusion .....   | 182        |
| VII.3              | Perspectives .....   | 184        |
|                    | Bibliography .....   | 189        |
|                    | Appendix A: Comparison of Different Inference Engines .....        | 203        |
|                    | Appendix B: Extract of the OWLManager Class Java Source Code ..... | 206        |

## List of Figures

|            |   |     |
|------------|---|-----|
| Figure 1.  | Overview of Telemedicine Environments.....                                  | 25  |
| Figure 2.  | Pervasive Computing Overview. P2P stands for “Peer to Peer”.....            | 38  |
| Figure 3.  | The Structure of OWL 2, from W3C.....                                       | 46  |
| Figure 4.  | End-to-end X73/EN13606 Design Scheme, from (Matinez 2007).....              | 54  |
| Figure 5.  | Simplified EPI-MEDICS Model. From (Rubel, 2005).....                        | 57  |
| Figure 6.  | Typical Sensors and Medical Web Services Architecture.....                  | 60  |
| Figure 7.  | The Context Broker Architecture CoBrA. From (Chen 2004).....                | 67  |
| Figure 8.  | Snapshot of the ERMHAN’s Architecture. From (Paganelli 2008). ....          | 72  |
| Figure 9.  | The Structure of U-TCA, from (Jiang 2008).....                              | 76  |
| Figure 10. | Main Steps of Situation Handling Modeling.....                              | 88  |
| Figure 11. | The Main Phases of our Methodology for ONOF-PAS Construction. ....          | 89  |
| Figure 12. | Graphical Representation of the Knowledge Meta-model.....                   | 92  |
| Figure 13. | Ontology Construction Methodology.....                                      | 95  |
| Figure 14. | Generic View of the Global Telemedicine Ontology.....                       | 98  |
| Figure 15. | Actor Ontology Concepts in the Telemedicine Domain.....                     | 99  |
| Figure 16. | Example of Task Ontology Concepts in the Telemedicine Domain.....           | 100 |
| Figure 17. | Example of Resources Ontology Concepts in the Telemedicine Domain.....      | 101 |
| Figure 18. | Concepts of a Typical Data Objects Ontology in the Telemedicine Domain..... | 102 |
| Figure 19. | Example of a Telemedicine Task Model Instantiation.....                     | 103 |
| Figure 20. | Example of a Complex Relationship Between Different Concepts.....           | 110 |
| Figure 21. | ONOF-PAS Rules Classification.....  | 110 |
| Figure 22. | Sequence Diagram for a Typical Patient Transfer Scenario.....               | 118 |
| Figure 23. | ONOF-PAS Architecture Overview.....   | 123 |
| Figure 24. | Profile Manager Model.....  | 125 |
| Figure 25. | Task Manager Model.....   | 125 |
| Figure 26. | Task Execution Algorithm.....   | 127 |
| Figure 27. | Graphic Representation of the SPARQL Query Described in Fragment 9.....     | 128 |
| Figure 28. | Mapping to ONOF-PAS Resources and Services Ontologies.....                  | 131 |
| Figure 29. | Mapping Definition Conforming to OWL-DL.....                                | 131 |
| Figure 30. | Entity Mapping Example.....   | 132 |
| Figure 31. | The ONOF-PAS Resources Processor.....                                       | 134 |
| Figure 32. | An Example of an Ontology Specifying the Taxonomy of Some Services.....     | 138 |
| Figure 33. | Messages Management Model.....  | 144 |
| Figure 34. | Messages Transmission Algorithm.....  | 146 |
| Figure 35. | Graphical Representation of the Main ONOF-PAS Ontologies.....               | 152 |
| Figure 36. | Extract of the ONOF-PAS Ontologies Classes, edited by Protégé.....          | 156 |
| Figure 37. | Extract of the Object Type Properties of the ONOF-PAS Global Ontology.....  | 157 |
| Figure 38. | Extract of the Data Type Properties of the ONOF-PAS Global Ontology.....    | 158 |
| Figure 39. | Patient Orientation Task Instance.....                                      | 162 |
| Figure 40. | Rescue Team Member Instance.....  | 162 |
| Figure 41. | Extract of a Patient Instance.....  | 163 |
| Figure 42. | Extract of the List of SWRL Rules edited using SWRLTab in Protégé.....      | 163 |
| Figure 43. | Extract of SWRL Rules concerning the Patient Orientation Task.....          | 164 |
| Figure 44. | Reasoning with the SWRLJESSTab Protégé Plug-in.....                         | 166 |
| Figure 45. | Number of Axioms inferred by JESS.....                                      | 166 |
| Figure 46. | Snapshot of the Patient Orientation Task Instance.....                      | 167 |
| Figure 47. | SPARQL Query Example 1.....   | 168 |



|            |  |     |
|------------|--|-----|
| Figure 48. | SPARQL Query Example 2 .....                                     | 168 |
| Figure 49. | SPARQL Query Example 3 .....                                     | 169 |
| Figure 50. | SPARQL Query Example 4 .....                                     | 169 |
| Figure 51. | Different use cases of TTROIE.....                               | 171 |
| Figure 52. | TTROIE main classes .....  | 172 |
| Figure 53. | TTROIE Components Diagram.....                                   | 173 |
| Figure 54. | Telemedicine Task Activities Diagram .....                       | 174 |
| Figure 55. | Main TTROIE Graphical User Interface.....                        | 176 |
| Figure 56. | Confirmation Message of the Initialization of New Instances..... | 177 |
| Figure 57. | Heterogeneous Resources Data generated by TTROIE.....            | 177 |

## List of Tables

|           |  |     |
|-----------|--|-----|
| Table 1.  | Different Use Cases of Telemedicine, from (Beolchi 2003) .....               | 34  |
| Table 2.  | Basic Technology Modules Used in Current Telemedicine Systems .....          | 65  |
| Table 3.  | Restrictions on the Classes of the Global Telemedicine Ontology Domain. .... | 108 |
| Table 4.  | SWRL Atoms Description. ....   | 111 |
| Table 5.  | Typical Resources Services Examples .....                                    | 135 |
| Table 6.  | Comparison of Resources Filtering Methods.....                               | 141 |
| Table 7.  | Examples of Messages Priority Level .....                                    | 145 |
| Table 8.  | Object Type Properties Domain and Range.....                                 | 157 |
| Table 9.  | Data Type Properties, Domain and “Range” .....                               | 159 |
| Table 10. | Task Execution Instructions .....  | 175 |



# *Chapter I*

## *Introduction*



# Chapter I Introduction

## Overview

This thesis proposes a knowledge-based framework to support the decision making processes in non pre-defined scenarios taking place in inimical environments. The proposed framework enables decision processes to efficiently adapt to the resources discovered in a dynamic context, where the availability and usability conditions of the heterogeneous resources required for performing tasks constantly change.

In the next sections, we introduce our motivation for this research work. Then, we illustrate the target of our thesis through an example of application scenario in the telemedicine domain. We expose the challenges related to telemedicine and the research questions for which we should provide answers through this dissertation. After that, we present the objectives of our research activities. Finally, we present the structure of our thesis.

---

### I.1 Context and Motivation

Since the last two decades, we witness an increasing evolution of information and communication technologies. Almost every year, new devices, technologies, standards, protocols, and languages appear. Many efforts of scientists, researchers, and industrial actors, have been employed to increase the performance of information systems in terms of services that these systems provide. Today, the availability of ubiquitous technologies, like smart cell phones, and the easy accessibility to the magic World Wide Web make the informatics services omnipresent. However, industrial actors still focus on tools and they often neglect the final user needs.

Thus, the first crucial issue is how to convince the end users to trust in the availability, continuity, and reliability of these omnipresent services. On the other hand, scientists and professionals in the ICT domain agree that information systems must provide high quality services. Unfortunately, this objective has not yet been perfectly achieved. For example, we still lack sufficiently high performance telemedicine systems that are able to infer and suggest several solutions to handle contextual situations in pervasive environments.

The second issue is that in pervasive environments we are surrounded by many sensors that capture our gestures, movement, sounds, and our biological data (e.g. elderly persons monitoring). We already know well which data should be acquired. At the same time, services are usually based on well defined protocols that mainly provide predefined information. However, what about other environments where there are no sensors, no predefined protocols, no predefined needs. To which extent can the available services be used? In fact, pervasive systems architects focus on the data that can be captured by distributed sensors in pervasive environments. Based on the captured data, they define the services that the information systems should offer. Additionally, system designers already predict the user needs and base the services on predefined protocols (e.g., a hotel room reservation, flight tickets reservation, online payment, online shopping, etc). However, it is almost impossible to meet all user needs by predefined services. In some scenario we do not know the entire user needs, services may meet part of these needs but users still have needs non met by the services they are using.

The third issue of the current pervasive information systems' context resides in the fact that in a given domain and within a business process, actors having different attitudes and skills perform multiple tasks according to their profiles. These tasks usually require heterogeneous resources, i.e., devices, services, knowledge sources, etc. Generally, distributed resources and knowledge are owned or managed by different organizations or institutes. As we know, resources conditions, in terms of availability, accessibility and capability, constantly change. On the other side, each task has its own specifications and may be run in dynamic environments, and sometimes in hostile environments. The aforementioned contexts limit the ability of pervasive systems to provide users with a high quality information, especially in application areas such as the eHealth or the telemedicine domain where the decision making processes may be very complex, are usually not expressible by exact algorithms, and may require intelligent compromise solutions combining both the knowledge about the objects treated by the tasks and their context.

The above described context motivated us to create a solution permitting an intelligent operation of the available heterogeneous resources required for performing tasks in both non-predefined scenarios and inimical environments. The proposed solution will be based on a framework we will

call ONOF-PAS (ONtology Oriented Framework for Pervasive Applications and Services) that involves ontological models to enable decision making processes to efficiently adapt to the dynamic conditions of the discovered resources. The system architecture we propose in this dissertation is task-centred and based on ontologies to describe the different entities of pervasive environments as well as the associations among these entities.

The proposed architecture will also comprise a rule base including a set of statements that specify how to link each task with the required resources, an inference engine to perform rule-based reasoning, and some additional object-oriented components related to tasks management and resources processing.

---

## **1.2 Scenario Example in Telemedicine**

A common objective among the different telemedicine applications is to provide healthcare professionals with appropriate decision support and solutions which meet their profiles, needs and context.

The most popular telemedicine scenarios are based on predefined protocols, for example follow up of patients at home, Tele-expertise in the cardiology domain between an emergency physician and a remote cardiologist, and follow up of rural populations or elderly in nursing homes. However, other scenarios are contextual, and not subject to predefined protocols. For example, finding the most appropriate hospital in which a patient can be hospitalized in case of an accident or the transfer of a patient from an hospital to another that has the required material, logistical and human resources available to achieve some specific treatments to the patient. These scenarios become more complex when tasks are performed in inimical environments, such as geographically critical or isolated areas like high mountains resorts where these environments lack the required healthcare infrastructure.

All the aforementioned scenarios require models that shall enable an intelligent management of the tasks, resources and exchanged messages among the different actors to increase the quality of services delivered by the healthcare providers.



---

### **I.2.1 A Real Episode**

Monday 29th Dec. 2008, according to AFP (Agence France-Presse), in the Paris area, a man suffering from a heart attack died after waiting nearly six hours for a place in an intensive care unit. The ambulance which rescued him at home had sought in vain a place in 27 hospitals. By default, the ambulance team brought the patient to the emergency service of the Longjumeau hospital which has the resuscitation equipment but was not able to perform a coronarography to the patient. It was only at 06:00 am that a place was released for an intervention at Lariboisière hospital in Paris, but the patient died when he arrived at this establishment.

Additionally, in high mountains or ski resorts, every year, a significant number of persons are victims of a heart attack. For example, 250 to 300 myocardial infarctions occur each year in the ski resorts of the French Nord-Alpin area. In such difficult environments, which lack healthcare infrastructures, it is a vital issue to provide rescue team members, first aid persons, or ordinary people surrounding the patient, with a decision support system able to propose them solutions about the patient's transfer in case of a health problem, especially a heart attack.

The medical literature claims for several years the importance of reducing the delay between the onset of the first symptoms and the hospital admission (Luepker 2005), for increasing the life expectancy of the patients and improving their health status, and thus for mastering the societal costs (McGinn 2005).

---

### **I.2.2 Patient Orientation Scenario Description**

To clarify the context of our thesis we present a typical telemedicine application scenario in pervasive environment concerning medical assistance in inimical environments, i.e. high mountains resorts. In such a scenario, according to the French context, the regulation center (Figure 1) acts as a mediator between the patient site and the healthcare providers, e.g. hospitals, medical centers, general practitioners or specialist offices, etc.

Let us consider a person, Mr. John Smith, who has a health problem while mountain climbing, skiing, or staying in a refuge or a resort. The victim needs immediate assistance and healthcare that might be provided by rescue team members, emergency physicians or a first aid

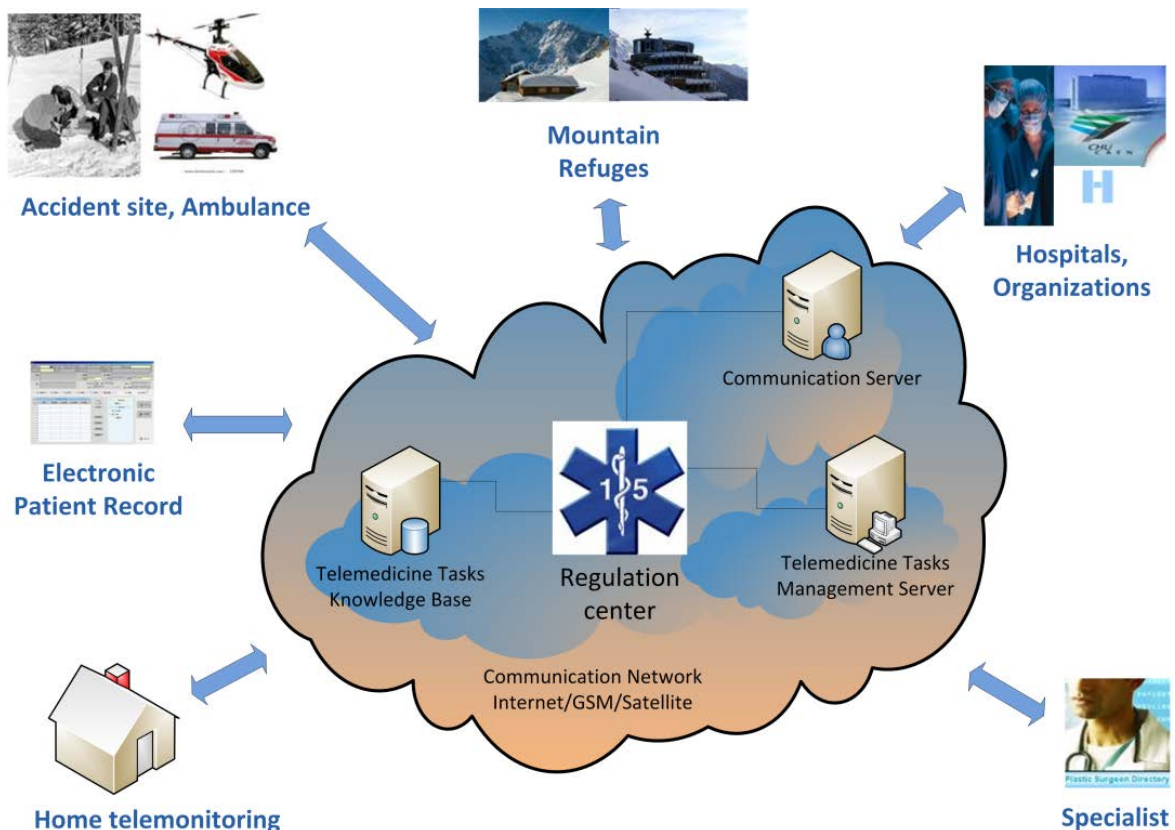


Figure 1. Overview of Telemedicine Environments.

person, who are in geographically distributed locations. The rescue team members, or the first aid person, who should provide a primary medical assistance to the person, need to know if they need and where to transfer the victim, taking into consideration multiple contextual factors like the person's clinical and social conditions, e.g. risk factors, medical history, etc, and also the hospitals abilities to receive the patient, in terms of accommodation, competence and availability of the different required resources, e.g. a bed or a place in an intensive care unit or a surgery unit, specific equipments or devices to perform additional investigations, staff able to provide the needed treatment, etc. In addition to the previous contextual factors, the location should be taken into consideration so that the patient can be transported to an hospital or a medical center located in the same geographic zone where the accident took place.

An effective collaboration framework is needed to save the patient's life. To start this collaboration, communication devices such as smart-phones or pocket PCs with wireless or satellite connectivity can be

used so that the user can access to a telemedicine system anywhere and anytime. An ideal pervasive telemedicine system should propose a set of tasks, accessible by domain actors according to their profiles. Depending on the geographic location and the available material and human resources, a telemedicine system will possibly have to infer several recipient addresses and to set up different messages that shall be addressed to one or more of these recipients.

The required telemedicine system should perform rule-based reasoning on the facts related to the healthcare professional profile, the type of tasks he/she can perform, and the clinical conditions of the patient. The results of the reasoning process should be related to the contextual knowledge about the environment and the material resources, including logistics and medical equipments, which are required by the telemedicine task being performed.

---

### **I.3 Challenges and Research Questions**

According to the scenario presented above, we can conclude that telemedicine applications require automated, available and continuous services and real time collaboration among software agents and actors who are geographically distributed in heterogeneous environments. However, telematic services are increasingly built on ad hoc systems models. Moreover, the expected functionalities of a telemedicine system are different and depend on application scenarios. Most existing systems are dedicated to a particular pathology type or to specific clinical situations.

Therefore, the common challenge facing researchers, designers, developers, and organizations is to adopt uniform technical frameworks that permit to define specific implementations of established standards. The objectives of such uniform technical frameworks are to achieve effective systems integration, facilitate appropriate sharing of medical information, and support optimal patient care.

Actually, the current systems' deficiency resides in the lack of personalisability and adaptability to the users' profiles and needs and to different contexts of use. The effectiveness of the services provided by pervasive systems is determined by many factors. The most important one is the availability of pertinent information anytime and anywhere. On the other side, pervasive systems' functionalities should take into consideration

the specificity of the different tasks performed by the users, the circumstances of the objects treated by the tasks, the conditions of the heterogeneous resources required to perform various tasks in terms of availability, accessibility and capability, as well as the environment in which the tasks are performed. In the following, we set out the challenges to which we should provide answers through this thesis.

- The scenario presented in the previous section concerns patient medical tele-assistance in difficult or inimical environment (i.e. high mountains resorts, ski resorts, deserts, jungles, submarines, arctic, etc) that are potentially dangerous and physically challenging to both humans and electronics. First aid persons or rescue team members are not skilled enough to take the right decision for the patient orientation or transfer. Therefore, the first challenge is:

How can we enable telemedicine tasks management in non-predefined scenarios occurring in unfavorable environments?

- Resources management is still a problem in pervasive computing. For instance, if a mobile user wants to perform a task in inimical environments, the task execution is limited by the availability and capability of the different required resources, i.e., data sources, services, devices, etc. Thus, the users have to manually find the best resources needed by the task they want to perform. Such a situation is not desirable in the pervasive computing world because it doesn't scale with the increasing amount of software services, user mobility, and resources dynamism. It also demands that users waste time on non task-related activities, for example making calls or sending faxes to verify that the required resources are available and accessible. Ideally, as users change environments, they should be able to quickly take advantage of local resources to continue working on computing tasks without a long manual set-up process. This entails two main requirements: firstly, the system should be able to take advantage of the available resources; secondly, the system should be able to automatically use local resources to accomplish tasks. Thus, the second challenge is:

How can we adapt the decision making process to the constantly changing conditions of resources required to perform telemedicine tasks?

We should also take into consideration other challenges like resources data updating and filtering. Moreover, we should find a strategy for messages transmission in terms of priorities and delivery, in addition to provide the user with exhaustive and pertinent information related to the task that he/she performs.

---

#### **I.4 Goals**

To face the aforementioned challenges related to decision making in non predefined scenarios and to meet the diversity of the user's needs in different contextual situations, there is an essential need for models able to capture knowledge about the following elements: the actors who perform multiple tasks in a given domain and in different contexts, the heterogeneous types of resources required for performing the tasks, and the knowledge related to the organizations that own or manage these resources. It is however a critical issue to design such models, because they should be effective and synergetic to realize knowledge acquisition from actors and other main entities in a given domain. Moreover, to achieve a successful knowledge modeling we should focus on well defined objectives and exploit the existing technologies, standards, and protocols.

Thus, we precise below our objectives in this thesis as follows:

- Create a system architecture permitting an intelligent operating of available heterogeneous resources, i.e. services, data and knowledge sources, devices, etc.
- Provide a solution to support decision making processes in non predefined scenarios that take place in inimical environments (e.g. patient orientation decision making in high mountain domains resorts).
- Meet user needs by providing him with pertinent data, taking into consideration the continuous changing of resources conditions and environment specificities. In other words, to achieve more effective communication of relevant information and to make best use of available services for all actors, particularly in a domain such as telemedicine, where our goal is to provide non skilled telemedicine services users with a faster, more reliable and automated patient medical tele-assistance, and consequently, to obtain a better level of telemedicine service quality.

---

## **I.5 Document Structure**

The remainder of this dissertation is organized as follows.

Chapter 2 includes a general background about Telemedicine and other related domains such as eHealth, pHealth, as well as the Electronic Health Record. In this chapter we also present the key concepts related to our research work, such as pervasive computing, context-awareness, knowledge representation and other semantic web technologies such as Description Logic, OWL, and SPARQL.

In chapter 3, we provide a rich state of the art in the following three main areas: telemedicine systems and applications, context-aware modeling, and task/activity-based computing. We describe and analyse the most interesting projects and approaches proposed by the research community.

In chapter 4, we describe our ONOF-PAS construction methodology and define the knowledge meta-model including general concepts identification, ontological models representation and formalization using an ontology description language such as OWL. In this chapter, we also describe the ontological models instantiation in the telemedicine domain. Finally, we give some detail about the rule base construction.

In chapter 5, we present our task-centred ONOF-PAS architecture. We explain in detail each component of the proposed architecture including the task manager, the profile manager, and the resources processor. Finally, we propose a message management model and a message transmission policy.

In chapter 6, we describe the implementation process of our proposal. In the first part of the implementation process, we edit and test all the ontological models using the Protégé Integrated Development Environment. We illustrate through an example, e.g. the Patient Orientation Task, how to infer the required heterogeneous resources by using the JESS inference engine and the SWRL rules to effectuate rule-based reasoning. In the second part of the implementation process, we describe, using the patient orientation task as an example, the TTROIE (Telemedicine Tasks Resources Ontologies for Inimical Environments) prototype we have

developed, including its software architecture, task execution algorithm and TTROIE user interface.

In chapter 7, we summarize our scientific contributions and discuss some perspectives for future research.

# *Chapter II*

## *Background*





# Chapter II Background

## Overview

In this chapter, we present the main concepts related to our research works, including pervasive computing, context & context-awareness, and the Task Computing paradigm. We also provide a background about knowledge representation, standard knowledge description languages, and semantic web technologies. As we have focused on the telemedicine application domain, we start this chapter by a general introduction of telemedicine. The other concepts and standards will be presented one by one in the state of the art (chapter III).

---

### II.1 Telemedicine

---

#### II.1.1 What is Telemedicine?

Early telemedicine initiatives date back to the beginning of the 20th century, where the Dutch physiologist Einthoven transferred ECG signals via telephone lines. In Norway and Sweden in the 1920's, telemedicine was applied to aid troubled seamen from the shore (using radio to give advice). In 1935, the International Radio-Medical Centre, which provides advice and assistance to seaman during medical emergencies, was founded. In 1955, the Nebraska Psychiatric Institute used closed-circuit television to provide care from the university medical centre to a state hospital over distance. A new boost to telemedicine was given by NASA in the 1960's and 70's. They transmitted measured psychological and physiological parameters from the spacecraft and space suits during missions (Doran 1996). In his turn, the European Union has launched the information society projects recommending the creation of healthcare networks based on common standards linking general practitioners, hospitals and social centers on a European scale (Bangemann 1994).

Digital transmission and compression empowered in the 1980's a new generation of telemedicine, mostly based on point-to-point videoconferencing (Wootton 2006). Currently, due to improved technological capabilities, real-time 24 hours monitoring and treatment of patients over distance has become feasible.

Today, Telemedicine systems are supported by advanced technologies like high speed computer networks and switching systems, telecommunications superhighways including cellular telephony, fiber optics and satellites, interactive video, and high resolution monitors.

Telemedicine, or medicine at distance, is defined as the utilization of information and telecommunication technologies to transfer medical information for diagnosis, therapy and education. The medical information may include images, bio-signals, live video and audio, video and sound files, patient medical records, and output data from medical devices (Norris 2002). Telecare is a related term and refers to the provision, at a distance, of nursing and community support to a patient by using information and communication technologies. Similarly, Telehealth refers to public health services, i.e., clinical, administrative and education, delivered at a distance, to people who are not necessarily unwell, but who wish to remain well and independent (Wootton 2006). Other definitions of telemedicine have been proposed. However, all these telemedicine definitions focus on the same aim which is to provide expert-based healthcare to understaffed remote sites and advanced emergency care through modern telecommunication and information technologies.

Table 1. Different Use Cases of Telemedicine, from (Beolchi 2003)

| Actors  | Applications Types                | Interaction Types                 | Current Technologies                        | Future technologies                                   |
|---|-----------------------------------|-----------------------------------|---|---|
| Doctor ↔ Doctor                                     | Medical opinion request           | Real-time, Data store and forward | Videoconferencing, E-mail, Internet         | Mobile devices, Virtual Reality, wireless connections |
| Doctor ↔ Doctor + Patient                           | Tele-consultation                 | Real-time,                        | Videoconferencing, Unmanned Robots          | Sensitive Technologies                                |
| Doctor ↔ Healthcare Professional ↔ doctor + Patient | Tele-assistance                   | Real-time                         | Videoconferencing, Orally data transmission | Sensors, Automatic transmission                       |
| Doctor ↔ Patient                                    | Advice request, Patient follow up | Real-time, Data store and forward | Telephone, Internet, E-mail, Television     | Intelligent monitoring, Embedded micro-sensors        |
| Doctor ↔ Institution                                | Tele-work                         | Real-time                         | Videoconferencing, Internet, E-mail         | Mobile Technologies,                                  |
| Doctor ↔ University                                 | Tele-education                    | Real-time                         | Videoconferencing, Internet                 | Virtual Reality, Data mining                          |

The Telemedicine glossary edited by the European commission (Beolchi 2003) lists the different use cases of telemedicine according to the interaction between the concerned actors and the used technologies. Table 1 represents this inventory.

---

### II.1.2 eHealth & pHealth

The technological evolution is expected to have a great impact on the healthcare process by increasing intercommunication among the actors from the scientific, medical, administrative and technical domains, as well as on the medical decision support domain by providing medical expertise anywhere. The requirements for a safe and high quality care as well as the increasing efficiency and productivity of the health systems are expected to be realized by increasingly introducing distributed and specialized health services. However, these services should become strongly oriented towards improving the management of the patient's personal health status and towards responding to the needs of the subject of care. These health services should be provided independent of time, localization, and distribution of resources in a high communicative and collaborative way called eHealth (Pharow 2009).

eHealth is defined through the use of internet and other electronic medias by healthcare professionals and patients in order to share, exchange, and access medical data (Wyatt 2005), (Blobel 2007). In fact, the telemedicine we witness today, is one of the eHealth operation modes, such as multimedia data and medical information (Electronic Health Record) exchange among the general practitioners and specialists for consultation and expertise (Telisson 2006), (Verdier 2006), and also decision support through an intelligent information system (Fayn 2010).

A report produced by the World Health Organization (WHO) and the European Commission for the world summit on the information society highlights the opportunities for eHealth in the different countries as well as the need for a global, long-term and collaborative approach, so that all citizens may benefit (Helay 2006) (WHO 2005).

The development of a cooperative medical assistance requires an effective partnership around the patient, involving several health professionals to improve the quality of care and to better interact with the patient. Accordingly, an adaptive communication system should be

developed. These developments have led to the concept of pHealth, which stands for Personal Health, where the patient is the major element of the healthcare plan since the systems adapt better to his needs. The patient is thus becoming the center of the healthcare systems. Enabling a common use of very popular, advanced Personal Health Systems (PHS) is the first issue to be addressed for improving the patient's health support, at any time, whatever he/she is at work, at home or during a trip (Fayn 2010). The objective of pHealth is to design, develop, and evaluate systems to ensure that citizens are closely involved in taking care of their own health. Regardless the technological development, the pHealth objectives involve to create and to evaluate new ways of communication and cooperation between the public health sector, private companies, and the citizen, as well as to develop technological systems that enable citizens to take care of their health problems. In the long term, healthcare systems should integrate pervasive technologies in hospitals, at the patient's home, on the patient himself, etc.

---

### **II.1.3 Personal, Shared, and Distributed Electronic Health Record (EHR)**

The EHR is a computerized file that intends to gather all relevant information of the patient clinical trial. In fact, EHR is designed in order to improve healthcare quality by providing the patients with medical assistance throughout their life. Additionally, EHR permits a better coordination among healthcare providers, avoids duplication of examinations and enables an efficient and rapid exchange of patient medical data among the health domain actors. Therefore, EHR is an essential entity in any telemedicine application.

Since a few years, the French Ministry of Health has been trying to promote the implementation of the Electronic Health Record to improve the coordination, the continuity and therefore the quality of care, as well as to reduce healthcare costs. In 2009, a new reviving plan of EHR has started and it is now being implemented. In his report addressed to the French ministry of health including recommendations concerning the personal electronic health record, Michel Gagneux (Gagneux 2008), has considered the personal electronic health record as a particular view of a shared EHR. It constitutes a space of shared data concerning a patient who has direct access to it without the need of any medical intermediate. On the other side,

healthcare professionals may have access to this shared data space under the patient's authorization.

Currently, the EHR is more a patient record than a medical one. Indeed, many various entities can provide information likely to be used in the EHR. These entities may be healthcare providers, such as Public and private health institutions, physicians, specialists, medical laboratories, and radiology centers, in addition to health insurance organizations, institutions and professionals involved in a social or medico-social framework. We should not forget also the patients themselves. On the other side, the characteristic "shared" resides in the capability of each healthcare professional or entity, at the condition of the patient's consent, to access information that are produced by other entities and eventually stored elsewhere. Additionally, healthcare and different other actors should be able to feed the shared EHR, from their workstation, with data needed to coordinate the healthcare process, while respecting the actor's liability.

In the same way, EHR enables healthcare professionals to use services in order to optimize their practice. It should also propose to the patients services that meet their needs. For instance, if we want that patients become real actors of their health and participate in the medical assistance process, the EHR should provide the patients not only with information concerning, for example, screening tests results, information about his disease, and prevention data, but also services allowing them to follow their treatment and to monitor their clinical status, particularly in case of chronic diseases (Rubel 2005).

The role of EHR related knowledge bases lies in providing the technical basis for optimizing the level of information needed for the physician's decision of treatment and in positively influencing everyday clinical processes. One of these processes is the clinical documentation of the treatment of every single patient (Ammon 2008).

---

## **II.2 Pervasive Computing**

Nowaday, technologies pervade our environment and make the information available anytime and anywhere, as Mark Weiser already expected in (Weiser 1999). Meanwhile, information systems, applications and services increasingly become pervasive. Thanks to networking, personal computing evolved to distributed computing. As computers became connected, they

began to share capabilities over the network. Distributed computing marked the next step towards pervasive computing by introducing seamless access to remote information resources and communication with fault tolerance, high availability, and security (Saha 2003).

Currently, cloud computing transforms the way we design, build and deliver applications. Enterprises must take into account architectural considerations when adopting and using the cloud computing technology. Sun and IBM have taken the initiative to support and provide this new technology (IBM 2009), (Sun Microsystems 2009).

The notion of pervasive computing refers to user-centric provisioning of services and applications that are adaptive to user preferences and monitored conditions, namely the related context information, in order to constantly and consistently offer value-added and high level services. The concept of pervasive computing further denotes that services and applications are available to the user anywhere and anytime.

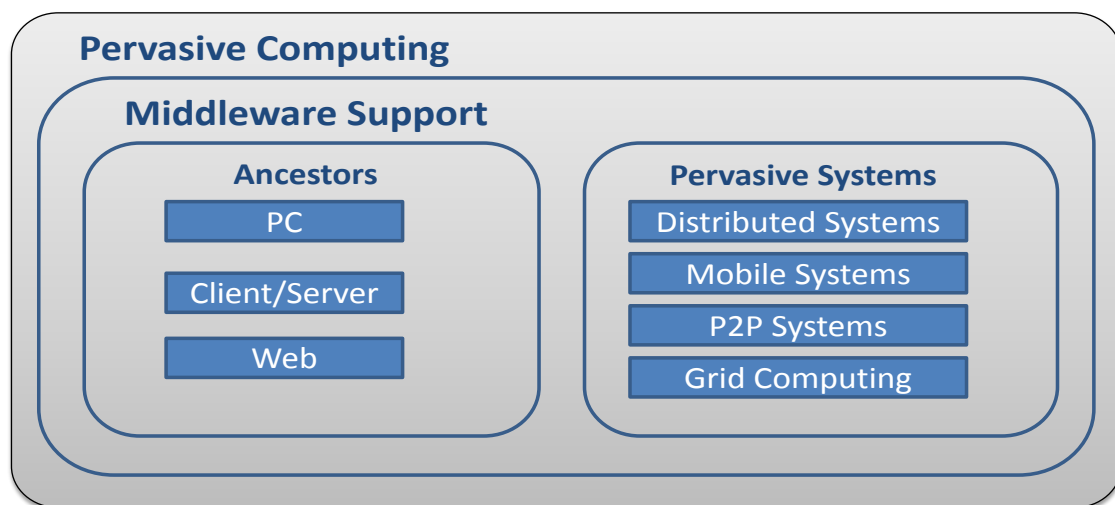


Figure 2. Pervasive Computing Overview. P2P stands for "Peer to Peer".

Mobile computing emerged from the integration of cellular technology with the Web (Satyanarayanan 2001). Both the size and price of mobile devices are falling everyday and could eventually support the vision of pervasive computing devices easily available to users in any human

environment. Figure 2 shows a general view of different pervasive computing aspects.

The mobile computing goal of “anytime anywhere” connectivity is extended to “all the time everywhere” by integrating pervasiveness support technologies such as interoperability, scalability, smartness, and visibility.

Now, telecommunication technologies have evolved, e.g., 3G, 3GS, the coming 4G, and WiMax (Worldwide Interoperability for Microwave Access), IEEE 802.16. On the other side, advanced terminals with enormous computing and storage capabilities, such as the iPhone and other smartphones, have appeared. Thus, more and more, information systems become ubiquitous and intelligent. However, they must be adaptable to the terminal type and to the user’s profile to ensure a suitable application use in pervasive environments. Doing that, pervasive systems will be able to meet user’s needs in different and dynamic environments.

In the next few years, nanotechnologies will be embedded in almost everything, not only in electrical devices. Therefore, pervasive computing environment will become saturated with computing and communication capabilities. In such a pervasive environment, which seamlessly integrates a large number of devices and services that help users perform different kind of tasks, the intelligent objects, e.g., user’s smart phone, watch, clothes, etc, will acquire information about the users’ context and profile.

A taxonomy of mobile and pervasive applications has been realized by (Dombroviak 2007) independently of the characteristics of the middleware or the infrastructure that support the application. This study lists the different abstracting application characteristics, e.g. transitionality, time constraints, goal, collaboration, lifetime, centricity and context-awareness related characteristics such as absolute location awareness, space awareness, proximity awareness, transition awareness, event awareness, object awareness, and operational awareness. In the next section, we clarify what context awareness means.

---

### **II.3 The Context-awareness Paradigm**

Context-awareness is a concept that appears in different disciplines such as philosophy, linguistic, psychology and computer sciences. More recently, with advances in mobile computing, context has become a topic of interest



to other areas of computer science, such as ubiquitous computing and ambient intelligence. In these areas, context is usually regarded as a set of environmental conditions that can be used to adapt applications to their user's current situation and needs. These areas are particularly interested in context due to mobility, which generates opportunities to explore the environment. As the user context frequently changes, applications running on the user's terminal may adapt their behavior according to these changes.

Many definitions of context in pervasive computing have been proposed (Schilit 1994), (Brown 1997), (Bradley 2005). The most referred definition of context was given by Dey: "Context is any information that can be used to characterize the situation of entities (i.e. a person, a place, or an object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves" (Dey. 2001). Nevertheless, according to Chaari (Chaari 2007), this definition is a source of conflict, because it does not distinguish between what is a part of context and what does not. They provide context definition as "Context is a set of external parameters that can affect the behavior of an application by defining new views on its data and functionalities. These parameters can be dynamic and may change during the use of the application".

The proposed definitions are however either very abstract or very specific to a particular area. These various definitions make the context formalization difficult due to the lack of precision or lack of generality.

---

## II.4 Semantic Web

The semantic web is a set of technologies which aim to make the content of web resources accessible and usable by software or agents through a system of formal meta-data using a family of languages developed by W3C<sup>1</sup> (World Wide Web Consortium). Tim Berner-Lee (Berners-Lee 2001) has defined the Semantic Web as an extension of the classic Web containing not only texts in natural language (English, French, Spanish, etc), but also formalized information and automated services to be automatically processed by the machine. In other words, the semantic web is the WWW

---

<sup>1</sup> <http://www.W3.org>

with inference capabilities. It aims at a better control of content and not only the syntax. Adding explicit semantics to the content of web resources transforms the web into a global source of knowledge which is useful for sharing and reuse data from different applications. In addition to dealing with smart applications, the semantic web makes the data more intelligent by using technologies for semantic web representation such as ontologies (Daconta 2003). Thus, the semantic web largely contributes to the development of new, more flexible approaches for data integration. The semantic web is based mainly on the use of standardized languages permitting to describe the content of web resources. For example, the formal ontology language OWL (Ontology Web Language) has been designed to provide high level descriptions of these resources (Bechhofer 2004). OWL is one of the most known languages in the field of semantic web.

The goal of the semantic web is also to provide a mechanism of reasoning on data, with less human intervention, by allowing to infer facts from a document, even if these facts have not been explicitly created by the author. Its syntax is based mainly on the use of URIs (Uniform Resource Identifier) which link up resources and terms in a document to other documents. Thus, the concepts of a document do not consist in a simple syntactic definition, but they are linked to shared resources which provide the context in which these concepts are defined (Fensel 2008).

In the eHealth domain, semantic web technologies can be used to exchange data about actors and different human, communication, and material resources in a more efficient and reliable way. However, if different vocabularies are used to describe data in eHealth applications, the ontology provides a mechanism which permits to detect the equivalence of terms. This implies that the models can eventually be shared with minor or no human intervention and especially with less data lost during the transfer process (Ganguly 2005).

---

## II.5 Knowledge Representation

People, organizations, and software systems need to communicate among themselves. However, due to different requirements and background contexts, there are widely varying viewpoints and assumptions regarding what is essentially the same subject matter. Everyone uses a different

language and may have differing, overlapping and mismatched concepts, structures and methods. The way to solve these problems is to reduce or eliminate conceptual and terminological confusion and come to a shared understanding. Therefore, a tool is needed to enable this shared understanding and facilitate the communication among people and organizations, as well as to support the interoperability among software systems. One of the best tools to realize these requirements is an ontology. In the next sections we define what an ontology is, and we present the different ontology types and description languages.

---

### II.5.1 Ontology

Researchers in the computer sciences field have borrowed the term “Ontology” from the philosophy which is a branch of metaphysics concerned with the nature and relations of being (Meriam Webster). The term “Ontology” is used to refer to the shared understanding of some domain of interest which may be used as a unifying framework to solve problems (Uschold 1996). It is now widely used in fields such as knowledge representation, information retrieval and knowledge extraction (Guarino 1998).

In the literature, several definitions of ontology have been proposed. But, the definition that characterizes the essence of an ontology in the artificial intelligence domain is based on a consensus definition given by (Gruber 1993) “An ontology is a formal and explicit specification of a shared conceptualization”. The conceptualization refers to an abstract model of a phenomenon in the real world, which identifies the appropriate concepts and their interrelations with this phenomenon. Explicit means that the type of the used concepts and the constraints on their use must be explicitly defined. Formal refers to the fact that an ontology should be comprehensible by computers, viz computers should be able to interpret the semantics of the provided information. Shared indicates that the ontology supports the consensual knowledge, shared understanding, and it is not restricted to certain individuals but accepted by a group (Broekstra 2001). Therefore, an ontology is used to represent knowledge of a particular domain by specifying different concepts and interrelations. Whatever the domain is, an ontology consists of several components. The most important

are: the concepts, the relationships, the attributes, the instances and the axioms (Gruber 1993).

Ontology engineering and management in the healthcare domain have a long tradition. It started with controlled vocabularies with restricted lists of terms such as catalogs, unstructured glossaries, structured arrangements of words such as dictionaries, thesauri and taxonomies including structured glossaries (LOINC 1994) (SNOMED 1999) (UMLS 2009). Today we witness the most advanced ontology types which are formal ontologies using mathematical procedures and logical statements (Blobel 2009a), (Blobel 2009b).

---

### II.5.2 Ontology Types

Ontologies can be classified according to the degree of representation formality in addition to the operational objectives including the communication among the users, the interoperability among the information systems, and knowledge domain (Guarino 1998). We can distinguish three main categories of ontologies as follows.

- The foundational ontology or the formal upper level ontology. These ontologies have a large granularity because they fairly involve generic concepts such as space, time, events, etc. In fact, this class of ontology is a general one which integrates philosophical foundations such as the principles that should be respected to design such an ontology. An example of such an ontology is BFO (Basic Foundation Ontology) (BFO 2011), (SUMO 2011).
- The core ontology, also called mid level ontology, which provides a bridge between the very abstract upper layer ontology models, hard to understand, and very specific domain ontologies. It is less generic and therefore its content (real-world examples) is easier to understand. It facilitates the construction of a consistent domain ontology for modelers less knowledgeable about upper level ontologies (Heijst 1997), (Gomez-Peroz 1999).
- The domain ontology is an ontology where the content focuses on a specific aspect or facet of the world/reality and on our human knowledge/representation of it. It governs a set of vocabularies and concepts that describe an application domain (Mizoguchi 2000). This type

of ontology can be obtained by the specialization of the mid level ontology. In addition, domain ontologies are not specific to a particular process or task. In the next chapter, we will focus on this type of ontology to explore the existing telemedicine domain ontologies.

---

### II.5.3 Ontology Description Languages

An ontology language provides a mean to store the knowledge as well as a format to represent the acquired knowledge. Different ontology description languages have been defined. Each one defines a syntax and an expression associated with it. In the following we briefly mention existing ontology description languages.

KL-One (Schmolze 1985) and F-Logic (Kifer 2005) are frame-based languages which use “Frames” to represent things, actions, and concepts, and “Slots” to represent relationships to other things, actions, and concepts, including properties and their values. Description logic based languages, such as KIF (Knowledge Interchange Format), are powerful in terms of expressiveness of knowledge representation and they are also appropriate for the specification of concepts and of their hierarchies (Baader 2003).

In addition to the previous categories, the OMD (Ontology Definition Meta-model) (ODM 2005), specified by OMG (Object Management Group), is a family of MOF (Meta Object Facility) which enables ontology modeling through the use of UML-based tools. The metamodels that comprise the ODM reflect the abstract syntax of several standard knowledge representations and conceptual modeling languages that have been recently adopted by other international standards (e.g., RDF and OWL by the W3C). The ODM is applicable to knowledge representation, conceptual modeling, formal taxonomy development and ontology definition. Additionally, it enables the use of a variety of enterprise models as starting points for ontology development through mappings to UML and MOF. ODM-based ontologies can be used to support:

- Interchange of knowledge among heterogeneous computer systems.
- Representation of knowledge in ontologies and knowledge bases.
- Specification of expressions that are the input to or output from infer-

ence engines.

Today, the most known languages are RDF (Resource Description Framework), RDFS (RDF Schema) and OWL (Ontology Web Language), which use the XML syntax. These languages have been standardized by W3C. In the next section we focus on OWL which we have used to formalize our proposed ontological models.

---

#### II.5.3.1 OWL & OWL 2

Recommended by W3C since 2004, OWL has become the standard language for ontology description. It has been created from its predecessor language DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer) (DAML 2000). OWL permits to define terminologies in order to describe concrete domains. A terminology is made up of concepts and properties (also called roles in description logic). A domain is represented by instances of concepts. In fact, OWL is primarily intended to facilitate the representation and automatic processing of Web content by providing, if compared with RDF and RDF-S, additional vocabulary and formal semantics that permit to create semantic relationships between concepts. Thus OWL allows us to explicitly represent the meaning of terms and relations between these terms. It reuses some of the constructors of RDF / RDFS, such as *rdfs: subClassOf*, *rdfs: domain*, and *rdfs: range*. But it also adds additional vocabulary to represent the logical combinations between classes, i.e., *intersection*, *union*, and *complement*. Furthermore, OWL specifies certain characteristics of properties, such as *symmetry* and *transitivity*. Additionally, OWL specifies restrictions on the membership of a class, constraints of cardinality, and enumerated classes.

Moreover, OWL aims to make web resources easily accessible to the automated processes, on the one hand by structuring these resources in a comprehensible and standardized manner, taking advantage of the marked structuring of the XML syntax, and on the other hand by adding meta-information to these resources (Bechhofer 2004). Hence, OWL has more powerful means to express meaning and semantics than RDF and RDFS. Furthermore, OWL takes account of the diversity of distributed knowledge sources and provides a method to gather information coming from several sources by enabling ontologies importation and linking.

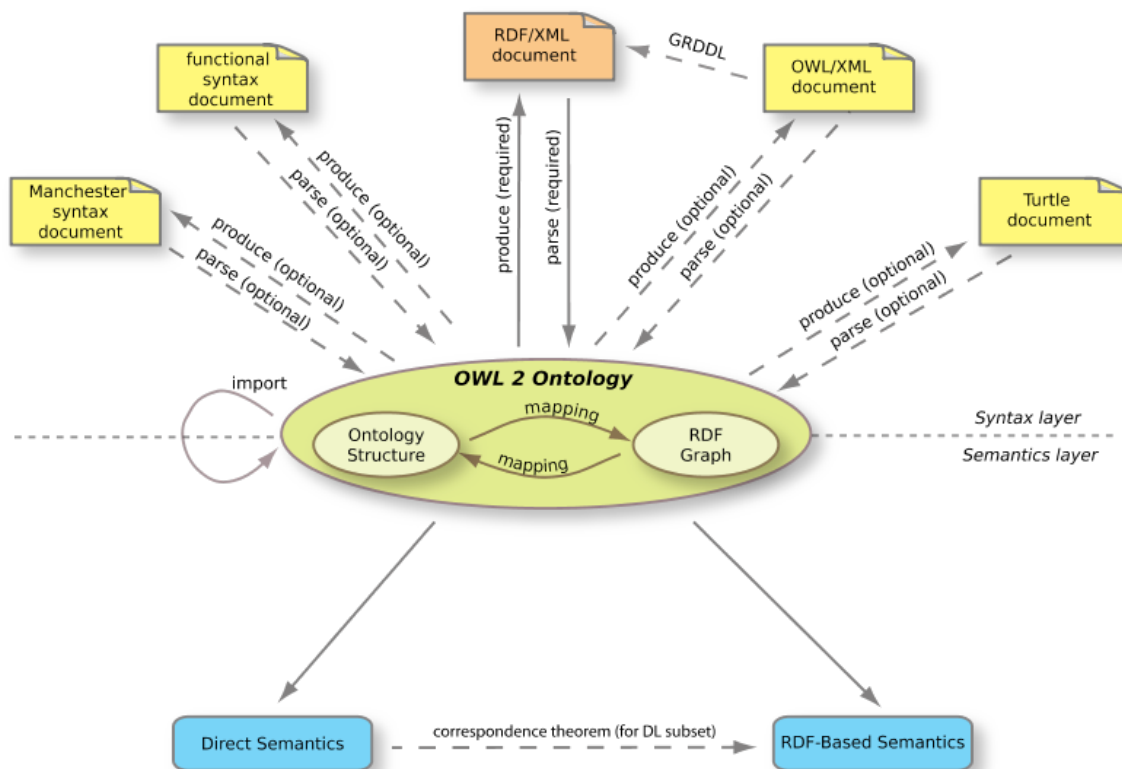


Figure 3. The Structure of OWL 2, from W3C.

OWL 2 is an extension and revision of OWL. At the time of this thesis redaction, OWL 2 is still under development by the W3C Web Ontology Working Group. Figure 3 shows the structure of OWL 2. It is very similar to the OWL one. All OWL ontologies are compliant with OWL 2 ontologies, with identical inferences in all practical cases. OWL 2 adds new functionalities when comparing to OWL, such as features related to the properties, for example, rich data and the data type range, i.e. properties *asymmetric*, *reflexive*, and *disjoint*.

### 11.5.3.2 OWL sublanguages

OWL defines three sublanguages, from the less expressive to the most expressive one, i.e. OWL-Lite, OWL-DL, and OWL-Full. Each sublanguage is an extension of its predecessor. In the following, we briefly present these three sublanguages.

- OWL-Lite is the most simple language of OWL. It only supports a subset of the OWL constructors and has a formal complexity which is lower

than the OWL-DL one. OWL-Lite is intended for users mainly needing a classification hierarchy and simple constraints. For example, it allows only the cardinality values 0 & 1. It does not allow to use the union and complement constructors, nor the class description through the enumeration of its individuals.

- OWL-DL is based on, as its name suggests, the Description Logic, DL. It is intended for users who demand a maximal expressivity for keeping the computational completeness of reasoning. It includes all OWL constructors, which can be used under certain restrictions such as the disjunction of classes, properties, individuals and data values.
- OWL-Full is the most complex version of OWL. It enables the highest level of expressiveness. OWL-Full and OWL-DL have the same constructors as OWL. The difference resides in the fact that OWL-Full permits free mixing of OWL with RDFS. As RDFS does not impose a strict separation of classes, properties, individuals, and data values, it is unlikely that any reasoning software will be able to support a complete reasoning for each OWL-Full feature. In other words, using OWL-Full, on the contrary of OWL-DL, does not allow the reasoning, because currently, a complete implementation of OWL-Full does not yet exist.

---

#### II.5.4 Description Logic Based Reasoning

One of the key features of ontologies is that they can be processed by a Reasoner which supports the decision-making process and provides the knowledge base with the capacity of reasoning by applying defined rules. Comparing with classic data bases, the open world assumption involves that the response to a request in a DL-based system needs to perform logical reasoning, often more complex than a simple search to verify the presence of the information, because the system often has to consider several possible interpretations. Therefore, the lack of information in a classic data base may be compared with the lack of knowledge in DL-based systems (Baader 2003). A Reasoner allows among others to provide a subsumption test for determining the hierarchy of classes between different ontology concepts. In addition, a Reasoner can check the consistency of the ontology gradually during its construction.



Generally, a typical ontology consists of a taxonomy of classes and a set of inference rules. A rule describes a conclusion which can be triggered depending on one or more conditions. It may be a statement processed by an inference engine that can generate an inference from a defined rule. The inference engines can deduce new knowledge, which is generated from already existing knowledge, using rules in the knowledge base. Thus, it is now possible to let the machine manipulate data in a more significant way. In our work, we are interested in the reasoning performed in the framework of ontologies based on the description logic such as those represented with OWL DL.

In the Description Logic formalism, a concept represents a set of individuals with common characteristics, while a role represents a binary relationship between individuals. Thus, a concept corresponds to a generic entity, while an individual corresponds to a particular entity, i.e. an instance of a concept. The concept and the role are defined by a structured description, elaborated from a number of description constructors such as union ( $\cup$ ), intersection ( $\cap$ ), negation ( $\neg$ ), and quantifiers ( $\forall, \exists$ ). These constructors allow the definition of new concepts and complex roles depending on simpler entities.

The concepts may be primitive or defined. A primitive ( $C \subseteq D$ ) concept is used for the construction of a defined concept. Its description contains only necessary conditions. A necessary condition may be: if an individual belongs to a class, then it must satisfy this condition. On the other hand, a concept is defined or complete ( $C \equiv D$ ) if its description contains at least a set of necessary and sufficient conditions. Consequently, these conditions are sufficient to determine that any individual who meets these conditions must be an instance of the defined class. It is also important to understand that a Reasoner can classify only defined classes.

Indeed, the knowledge in Description Logic is generally defined through two levels (Baader 2003).

- The terminological level (T-Box): At this level, we can carry out the concepts representation and manipulation, as well as the roles or the relationships between these concepts. Thus, terminological axioms of inclusion  $\subseteq$  or equality  $\equiv$  can be defined between concepts or roles. For example  $A \subseteq B$  or  $A \equiv B$  for the two concepts A and B, and  $S \subseteq R$   $S \equiv R$  for the two roles R and S.

- The assertion level (A-Box): This is the individuals description and manipulation level, which represents the factual instances. For example,  $A(x)$  and  $R(x,y)$  respectively represent the assertion of concept  $A$  and the role  $R$  (relation between instances  $x$  and  $y$ ).

---

## II.6 Conclusion

In this chapter, we have synthetically presented the main key words that our thesis is based on. We have introduced to the reader the central notions related to telemedicine, pervasive computing, context-awareness, and knowledge engineering. To have a more detailed description of some telemedicine systems and their applications, as well as of the different methods of context modeling, we invite the reader to have a look on the next chapter which provides a more profound state of the art of the aforementioned areas.



# *Chapter III*

## *State of the Art*



## Chapter III State of the Art

### Overview

In this chapter, we explore the most interesting research works including approaches and projects that are relevant to the subject of this thesis. We provide an analysis of the related works and take position in terms of research problems and objectives. The state of the art we present in this chapter, concerns the three following main domains: telemedicine systems, context modelling, and task/activity modelling.

---

### III.1 Telemedicine Systems

---

#### III.1.1 Introduction

Telemedicine systems allow an exchange of information with remote mobile sites and enable better utilization of existing healthcare resources. They also help patients to stay in their communities, to maintain a normal lifestyle while being treated, to reduce the hardships (physical and economic) associated with travel (for patients or physicians) and enable early diagnosis, intervention and treatment, where needed and when needed. Indeed, tele-consultation, tele-monitoring, tele-surgery and tele-education can provide an effective way for distant interaction between different actors who can communicate together by means of messages exchange and/or tele-presence solutions to perform different tasks aiming at improving healthcare (Wootton 2006), (Norris 2002). The novel technologies of fiber optics, digital switches, and multimedia terminals guarantee a high quality of service (QoS) in conjunction with reasonable operation cost. Such advanced technologies offer a good chance to healthcare providers to introduce high quality telemedicine services. In the following sections, we specify the medical data type that can be exchanged within the healthcare domain and the standards that govern the representation and the exchange of medical data.

---

#### III.1.2 Data Types and Standards in Telemedicine

According to our literature review in telemedicine, we can classify the exchanged data among telemedicine systems in four main categories:

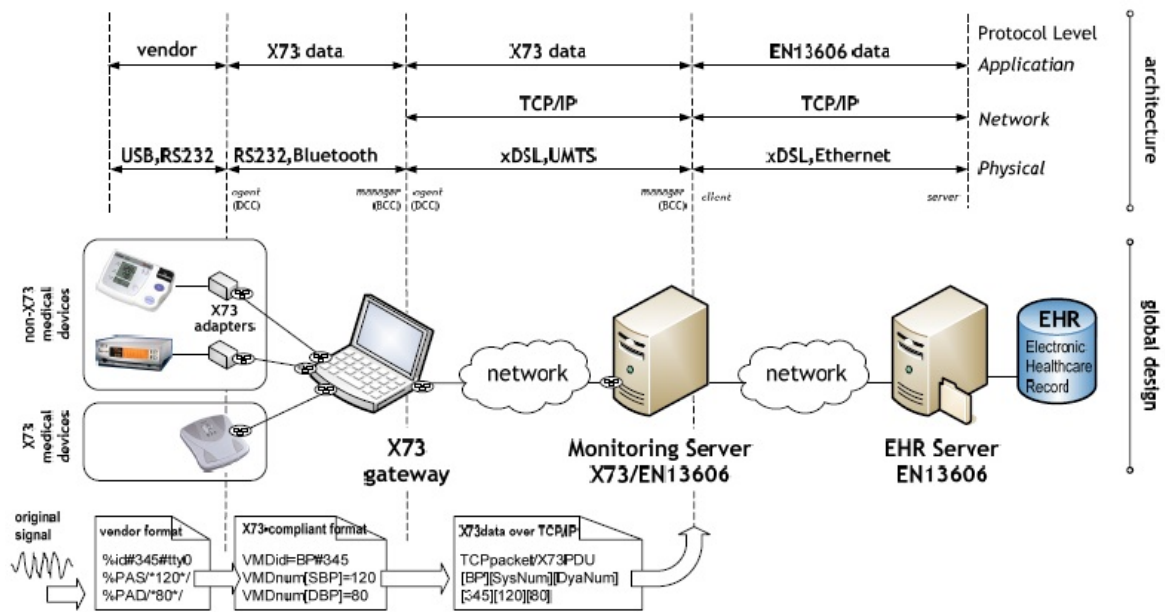


Figure 4. End-to-end X73/EN13606 Design Scheme, from (Matinez 2007).

- Textual data, e.g. the results of biomedical examinations, performed by analysis laboratories, and diagnostic reports which can be transferred using Portable Document Format (PDF).
- Audiovisual data based on the MPEG format, which are transferred during a videoconferencing session using the H 320 standard.
- Imaging data, i.e. X-ray, Magnetic Resonance Imaging (MRI), ultrasounds, which are transferred using DICOM<sup>2</sup>, the HL7<sup>3</sup> standard, or other standards of CEN/TC 251 WG4.
- Physiological signals, i.e. ECG, EEG, ... provided in their own standard interchange format, for example SCP-ECG<sup>4</sup> for the ECG, which has been set up by the European normalization committee (CEN).

Additionally, standards such as ISO/IEEE 11073 (X73) and EN13606 are used to provide tele-monitoring solutions. (Galarraga 2007) and (Martinez 2007) have conducted a cooperative research effort devoted to the

<sup>2</sup> Digital Imaging and Communications in Medicine, "medical.nema.org"

<sup>3</sup> Health Level Seven International "www.hl7.org"

<sup>4</sup> SCP-ECG has been recognized in 2007 as an ISO standard (ISO/DIS 11073-91064)

development of an end-to-end solution based on the aforementioned standards, see Figure 4.

---

### **III.1.3 Telemedicine Systems Modeling Approaches and Applications**

According to the bibliographic study we have done, we synthesize in the following sections the different types of proposed telemedicine systems and applications.

---

#### **III.1.3.1 Multi-Agent based Telemedicine Systems**

Agent-based modelling and simulation (ABMS) is a new modelling paradigm to model systems composed of interacting autonomous agents (Macal 2006). Moreover, agent-based modelling has become so widespread, because we live in an increasingly complex world where systems that we need to analyse and to model are becoming more complex in terms of their interdependencies.

Although there is no universal agreement on the precise definition of the term “Agent”, definitions tend to agree on more points than they disagree. Some modellers consider any type of independent component (software, model, individual, etc.) to be an agent (Bonabeau 2002). An independent component’s behaviour can range from primitive reactive decision rules to complex adaptive artificial intelligence (AI). Other modellers claim that a component’s behaviour must be adaptive in order for it to be considered as an agent, which means that the agents can learn from their environments and change their behaviour in response. The fundamental feature of an agent is the capability of the component to make independent decisions. This requires agents to be active rather than purely passive. For agent-based system development, JADE (Java Agent Development framework), an open-source software framework, is used to assist the development and execution of agent applications in compliance with the Foundation for Intelligent Physical Agent (FIPA) specification for interoperable multi-agent systems. Additionally, the FIPA ACL (Agent Communication Language) standard language is used to represent the messages exchanged among agents of the multi-agent system.

Computational advances make possible a growing number of agent-based applications across many fields. Fox, (Fox 2003), demonstrated that distributed agents systems have the potential to improve



the operation of healthcare organizations, where failures of communication and coordination are important sources of error. Lhotska conducted a study to model the management of medical documentation in a hospital using a multi-agent system and JADE for its implementation (Lhotska 2007). The process of medical documentation has been decomposed into four groups of tasks: storing and retrieving stored data, interaction with the user, data archiving, and security. Each group of tasks is fulfilled by a specific agent. Koutkias proposed a multi-agent system to model patient interaction with a medical contact center offering homecare services (Koutkias 2005). The multi-agent system is designed as a cooperative agent team, where agents share a common goal and each one adopts a request to do its share towards achieving the goal of the team. uEyes is a ubiquitous system, proposed by (Takahashi 2007), for supervising care-support services that fulfil the users' requirements based on their physical locations and on the status of the system components in ubiquitous computing environments. uEyes is implemented as a set of agents which are prepared to build applications for supervision.

---

#### III.1.3.2 Mobile Personal Health Systems

In the near future, mobile health systems will have an important role to play in improving the likelihood of detecting ailments at an early stage, when the need for in-hospital treatment can be averted, as well as in easing the increasing overwork on healthcare infrastructures.

For these reasons, the European Community is supporting a large number of research projects including the recently, successfully finalized EPI-MEDICS project which has developed an intelligent Personal ECG Monitor (PEM) for the early detection of cardiac events (Rubel 2005), (Fayn 2010). The PEM embeds advanced decision making techniques, generates different alarm levels and forwards alarm messages to the relevant care providers by means of new generation wireless communication, i.e. GPRS/EDGE/UMTS, etc. The PEM embeds itself a web server to facilitate the reviewing and/or updating of the embedded EHR (Electronic Health Record). The messages and the EHR are encoded in XML. Major alarm messages are automatically transmitted to the nearest emergency call centre. Data leading to medium or minor alarms are temporarily stored on a central web server and the health professionals are

informed by a SMS to consult these data. The web server embarked in the PEM automatically generates HTML pages consultable by means of any web browser. Thus the health professional will be able to reach the information stored in the PEM (list of contacts, data files, ECGs, configuration parameters) in the same way as he/she accesses the information on Internet. Figure 5 shows a simplified model of the EPI-MEDICS concept.

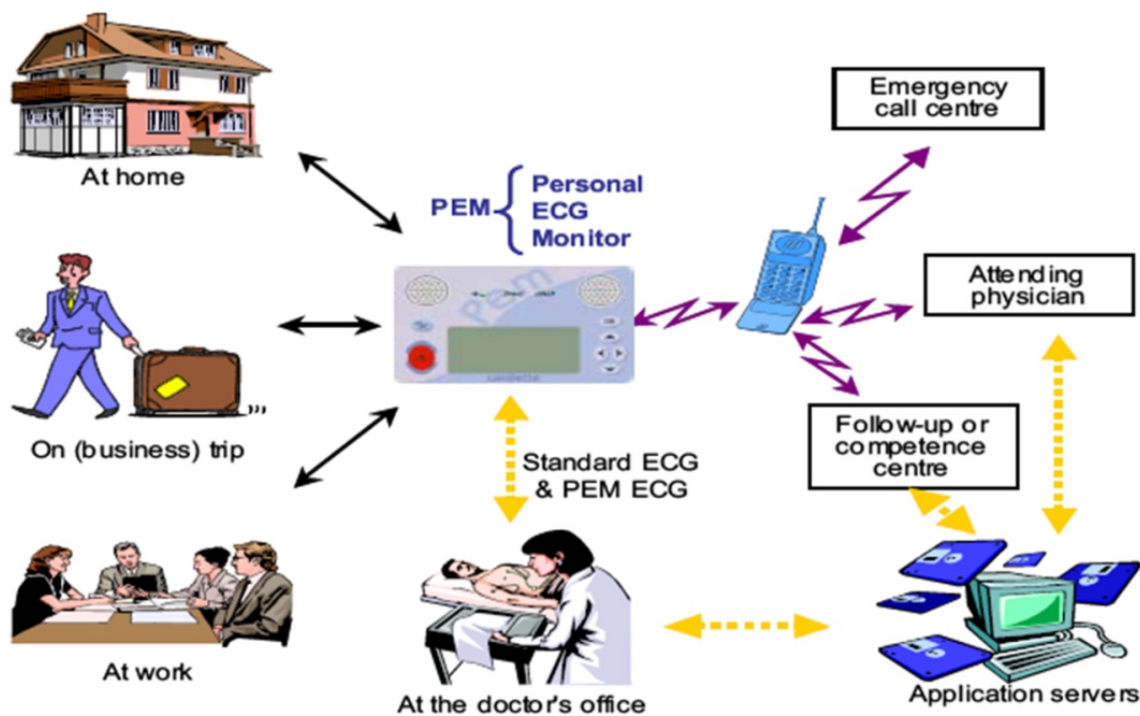


Figure 5. Simplified EPI-MEDICS Model. From (Rubel, 2005).

Today, mobile phones being extremely widespread have become ever more powerful. Modern mobile phones possess a set of cutting-edge features and technology in terms of processor, memory and screen capabilities as well as communication abilities. Phones provide socially appropriate means of displaying timely information to the user and enable the sensors to transmit measurements directly to healthcare providers (O'Brien 2010). A mobile healthcare network has been developed by (Kariyawasam 2010) to provide facilities for patients to upload their medical information to a database and to maintain their profiles, in addition to enable doctors to check their patients' profiles by means of a hardware

component developed to acquire the signals including medical information of the patient and update the central database via mobile phone.

---

#### III.1.3.3 Other Web-based Telemedicine Systems and Applications

The most sophisticated telemedicine systems involve interactive videos. These systems require extensive investments towards the design, implementation, and maintenance of telecommunications infrastructures. In addition to laying long distance fibre optic networks and setting up satellite links, these facilities are provided with specialized equipment and software at all locations where human experts interact (Sankaran 2000).

Many patients today have internet connections already set up in their homes. The ability of internet to carry multimedia such as voice, video and data combined with its low operating cost features makes it an attractive way for telemedicine consultations. Additionally, advances in data compression, security, interoperable platform software, and modeling techniques of designing distributed networks make the internet a promising tool for future telemedicine applications.

Kolovou presented a model for telemedicine service providers (TM-Provider) based on two controllers for the management of the user requests: the “Transaction Controller” and the “Process Controller” (Kolovou 2004). The first controller receives the requests and manages them applying authentication and access security procedures. The second serves the processes of the transaction, executes the corresponding events upon the medical data and covers the integrity and protection principles for the application data of the telemedicine system.

The Citizen Health System (CHS) project (Maglaveras 2005) aims at the monitoring, treatment, and management of chronically ill patients at home. CHS consists of databases, medical signals archiving systems, user interfaces, and additional processing modules. Based on the results of the CHS project, the INTERLIFE project was conducted by (Maglaveras 2006). INTERLIFE aims at helping Congestive Heart Failure (CHF) patients to prevent acute episodes and expensive hospitalizations, to educate them to a healthy life and to monitor them during a follow-up.

For the areas that cannot be served by the existing telemedicine solutions, an approach has been proposed for creating scalable telemedicine networks based on Delay Tolerant Networking (DTN) techniques using

store-and-forward Voice-Over-IP (VoIP) (Scholl 2009). DTN operates by leveraging mobility and local communications between participants in the network. Each member of the network communicates with the other members when possible, for example when they are close enough for local wireless communication (using WiFi, Bluetooth, etc.), or when a long range link becomes available. Members store messages from each other and forward them later on when they establish connectivity with other members. This type of telemedicine network allows communication of non time-critical information between participants. However, in emergency scenarios where the time is a critical factor, store-and-forward using VoIP cannot be used.

In the videoconferencing application domain, the OTE-TS (Hellenic Telecommunication Organization Telemedicine Service) project provides the user with integrated electronics which handle diverse types of conversational audiovisual information and medical records. It includes a stack of functions, protocols, and interfaces for data acquisition, processing, and display (Karavatselou 2001). Wen-Pai has defined a High Definition Telemedicine system (HDTM) architecture that leverages the network as a platform for intelligent transport and services to support high-definition videoconferencing and high-quality video and audio telemetry (Wen-Pai 2010). Another video conference based monitoring system for remote patient disease diagnosis and treatment using real time protocols (MPEG4/H.26X) for the video and audio sessions and for connecting sophisticated medical equipments, has been proposed by (Din 2010).

Several emergency telemedicine systems have been developed and tested for emergency physicians to communicate with remote specialists for consultations, treatment, diagnosis, and to transfer instructions through the telemedicine equipments (K. YOO 2004). Most of these systems employ existing standards for data transmission such as DICOM and MPEG. (Kang 2007) and (Jung 2007) have developed a system for emergency care using an HSDPA (High Speed Downlink Packet Access) modem. The proposed system is used by emergency rescuers to get directives from medical doctors and to provide emergency medical services to the patients in the ambulance. The medical data are transferred by using a relay server which is a multi-threaded, scalable, and stand-alone application based on the C# WinSock (Windows Socket) library. The relay server used by Jung operates similarly to a P2P (Peer-to-Peer) scheme for the content-delivery, but

provides additional and enhanced functions such as client session management, performance monitoring, and control of the effectiveness of data distribution. The platform RES@MU developed by IFREMMONT (L'Institut de Formation et de Recherche en Médecine de Montagne) (Ifremmont 2007) offers an open source solution using technologies such as SOAP and XML to exchange medical data and information and to manage in real-time and in a safe mode all the exchange processes triggered by the medical emergency.

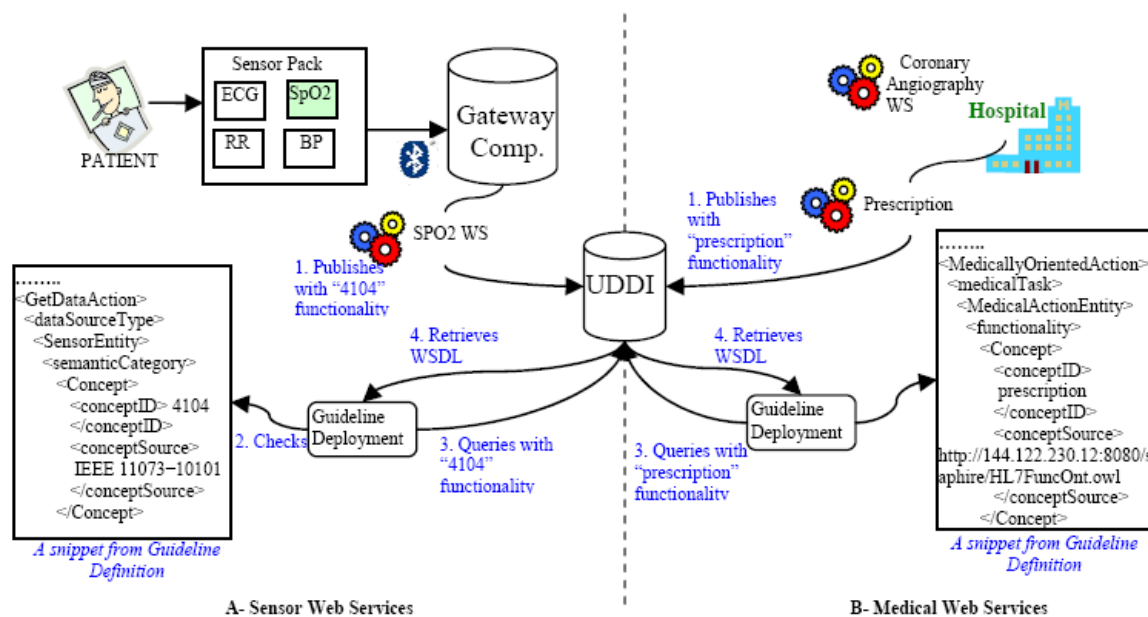


Figure 6. Typical Sensors and Medical Web Services Architecture, from (Nee. 2008).

Paula developed a web-based telemedicine system using application server pages (ASP) and Javascript for the homecare management of patients suffering from chronic obstructive pulmonary disease (Paula 2006). It runs on a Microsoft Internet Information Server and uses Open Database Connectivity (ODBC) to access a Sybase database management system. Also, in the context of homecare, a coordination platform to support the work of all the actors involved in the homecare processes has been proposed by (Beuscart 2004). An ongoing Taiwan project conducted by (Lu 2010) is designing and implementing pervasive health services infrastructures based on a grid system which integrates P2P

resources sharing mechanisms to provide homecare monitoring systems for preventive medicine.

The SAPHIRE project (SAPHIRE 2008) provides a clinical decision support system for healthcare monitoring based on computerized clinical guidelines, i.e., definitions of medical plans for the study of medical problems, following the European Cardiology Guidelines (Nee 2008). Figure 6 shows the data transmission mechanism, based on the IEEE 11073 standard, from wireless sensors to the gateway computer where they are exposed as web services. When a guideline is to be deployed, first of all the relevant web services are discovered from the UDDI (Universal Description Discovery and Integration) registry through their semantics, and saved as a guideline definition to be used for the guideline execution.

In neuroscience, (Healy 2010) proposed a web-based telemedicine application for the remote monitoring of the EEG (Electroencephalography) data, the electrical activity of the brain. The proposed system reduces the delay that can arise when offsite expert analysis is required and allows EEG data to be streamed to a remote location nearly real-time and viewed while acquisition is ongoing.

---

#### **III.1.4 Ontologies and Telemedicine**

Software interoperability is a major issue in telemedicine due to the variety and diversity of telemedicine applications and services which are required for the patient care. Today, ontology has emerged as a significant technique for software interoperation. An ontology-driven agent system has been proposed for diabetes patient (Ganguly 2005). This system uses an ontology to share common understanding of the structure of information. For example, if the General Practitioner GP uses the term “drug” to indicate the medicine that a patient is taking, and the specialist uses the term “treatment” to indicate the same term, a simple multi-agent framework will not be able to interpret the context of each message although it has the same meaning. Ontology can facilitate the conceptual understanding of terms, and an agent can query this ontology to perform an intermediate translation of the terms to be used by other agent systems.

From the EHR sharing point of view, the project ARTEMIS (A Semantic Web Services-based P2P Infrastructure for the Interoperability of Medical Information Systems) has as objective to share the patient

electronic health records among different enterprises by enabling the interoperability between the different HL7, OpenEHR, and EHRcom (EN 13606) standards thanks to ontology mapping and data exchange web services (ARTEMIS 2004). The project developed an engineering approach to provide the exchange of meaningful clinical information among healthcare institutes through semantic mediation (Bicer 2005), (Oemig 2009). The framework proposed by ARTEMIS involves the mapping of a source ontology into a target ontology with the help of a mapping tool which produces a mapping definition. This mapping definition is then used to automatically transform the source ontology message instances into target message instances.

An ontology-based model, called Smart Telehealth Home (STH) (Latfi 2007), takes advantage of the full potential of ontologies to describe the smart home domain, in order to provide an effective base for the development, configuration and execution of software applications. The ontologies of STH (i.e. habitat ontology, person and medical history ontology, equipment ontology, behavior ontology, and decision ontology) are employed to initialize the Bayesian networks used for recognizing which activity is most likely to be performed by the patient at a given time and in a given place. Another ontology-based model has been proposed for monitoring and assisting the patient at home by (Paganelli 2007). The proposed model consists of several ontologies describing the patient domain, the home domain, alarm management, and the social context (Paganelli 2008). The components of the proposed ontology-based model have been implemented by adopting standard technologies such as internet protocols, XML, and Web Services.

---

### III.1.5 Telemedicine Networks Quality of Service

The performance of pervasive services is mainly related to two critical factors which are the quality of service QoS and Resources Management. In the telecommunication domain, the International Communication Union (ITU) refers to QoS as “a set of quality requirements on the collective behaviour of one or more objects”. A number of QoS Parameters, i.e. throughput, transfer delay, jitter, error rate, and packet loss, describe the speed and reliability of data transmission. However, in the Web Services

domain, the parameters of QoS can be for example processing time, response time, latency, reliability, security, etc, (Le Blevec 2008).

In fact, guaranteeing a sustained QoS in dynamic situations requires accurate system awareness through monitoring and proactive QoS and resources management. Any QoS solution must scale as the number of users and applications increase or reduce. It also must be able to seamlessly adapt to dynamically changing conditions and priorities (Shirazi 2004). Thus, in dynamically changing environments, a successful QoS and resource manager must:

- Be aware of the current state of resources, e.g. network load.
- Know the application requirements.
- Be able to predict QoS violation.
- Be able to diagnose the cause of violations.
- Succeed in managing and allocating the resources.
- Have the tools to carry out QoS and resources management decisions.

A middleware services layer was proposed by the PICO (Pervasive Information Community Organization) project (Kumar 2003) where resources management services are implemented as software agents.

Another QoS framework for telemedicine applications has been proposed to provide and manage QoS in a telemedicine system intended to allow tele-diagnosis and get second medical opinion from consultants from a central hospital (Lage 2004). Several QoS parameters, e.g. capacity, delay, jitter, and loss have been associated to the standards concerned in this study such as DICOM, JPEG2000, MPEG-2, Audio G.711 and Video H.261.

An adaptive multimedia transmission mechanism for telemedicine applications to enhance the bandwidth efficiency and reduce the network influence has been proposed by (Lee 2004). The mechanism builds on different transmission policies, i.e. messages classification, messages prioritization and messages queues, for different kinds of the media objects such as transmission priorities, protocol, and bandwidth. A model for implementing QoS through Asynchronous Transfer Mode (ATM) has been suggested by (Nanda 2007), including a telemedicine scenario for a virtual critical care unit with given traffic specifications.



Qureshi proposed an approach which aims to build a high-quality mobile telemedicine system that overcomes the limitations of individual public wireless data networks, where data channels are unable to handle the demand of telemedicine applications, by aggregating several physical channels to construct a high-bandwidth virtual channel (Qureshi 2005a). To achieve this process, a software middleware called HORDE was developed to facilitate flexible network striping over unstable dissimilar channels (Qureshi 2005b). This software middleware provides an application which has the ability to Stripe data streams over a dynamically varying set of heterogeneous network channels, to control per-network-channel congestion and explicit flow control feedback for each stream.

The availability of WiMax/IEEE 802.16 technology will offer new opportunities to provide broadband access solutions as an alternative to cable and digital subscriber lines (DSL). This technology is expected to provide high data rates over large areas and also to provide a large number of users in the rural and developing areas with connectivity. The Broadband wireless access (BWA) possibilities of the WiMax/IEEE 802.16 technology makes it an obvious and feasible choice for telemedicine applications using both fixed and mobile environments. A simulation study using WiMax/IEEE 802.16 for bandwidth-efficient and secure transmission of medical records for telemedicine application has been effectuated by (Kumar 2008).

---

### **III.1.6 Analysis of Telemedicine Systems**

Regardless of the technical advances, any telemedicine system is constructed with four basic modules: data transmission, content, terminals and instruments, data and information processing. Several technology examples for each one are schematically summarized in Table 2.

An interesting statistical study done by (AOKI 2003) has demonstrated the effectiveness of telemedicine practice through the evaluation of some specific aspects of telemedicine, such as training and needs assessments, in terms of methods and outcomes. This study has concerned one hundred and four articles evaluating either clinical or non clinical outcomes of telemedicine. According to this study, almost all articles demonstrated that telemedicine is useful and that it can play an important role in the future healthcare. Another evaluation study of

telemedicine systems has been conducted by (Thiyagarajan 2006) to critically analyze the evaluation and assessment frameworks that have been applied to telemedicine systems. The analysis made by this study differentiates the papers according to different stages: information acquisition and information display, network transmission, and systems functionality. Network transmission includes time delay measurement (e.g. still image transfer time), video-clip transfer time, live video transmission rate and live video latency. Systems functionality includes the quality of network transmission and the software functionality. Thiyagarajan concluded that there has been insufficient rigorous, objective technical evaluations and assessments of telemedicine systems.

Table 2. Basic Technology Modules Used in Current Telemedicine Systems

| Module                          | Examples  |
|---------------------------------|---|
| Data transmission               | Telephone, GPRS-EDGE, 2G, 3G, WiMax, DSL, ISDN, ATM, leased lines, optical network  |
| Content                         | Voice, videoconference, 2D and 3D gray level and colour images, grey level and colour videos, physiological signals, physiological measurement, clinical data, ...    |
| Terminals and instruments       | Television, personal computer, PDA, mobile phone, scopes (e.g., dermascope, otoscope, stethoscope, ophthalmoscope), scanner, photo camera, video camera, ...          |
| Data and Information Processing | File exchange, data base construction and maintenance, security, interactivity, signal and image processing for diagnosis support, augmented and virtual reality, ... |

Giansanti et al. have proposed a set of tools and procedures to perform a telemedicine quality control process (Giansanti 2007). These tools are: 1) an informative questionnaire which serves to acquire the information about the examined telemedicine product/service; 2) a classification form allowing to classify a telemedicine product/service as belonging to one application area of telemedicine; 3) a technical file intended to act as a technical dossier of the product, forcing the telemedicine supplier to deliver the only requested documentation of its product in order to avoid redundant information; 4) a quality assessment checklist of the requirements regarding all the essential aspects of the telemedical applications that each telemedicine product/service must meet. By assessing the information provided by the previous components, a quality level of the telemedicine application can be assigned.

---

## **III.2 Context Management, Modelling and Computing**

There are several approaches to model context. These approaches vary in the way that context is structured, i.e. how contextual information is represented and exchanged. Strang has classified the context modelling approaches according to the data structures schemes which are used to exchange contextual information in the respective systems (Strang 2004). According to Strang's classification, context modelling approaches can be grouped in six categories: key-value models, mark-up schema models, graphical models, object oriented models, logic based models and ontology based models. Strang concludes that ontology-based context models show the most promising approach for context modelling in the pervasive computing domains with respect to handling context, validating context, providing quality of context indicators, supporting incompleteness and ambiguity of context, providing a formal definition of the domain and applying context to existing domains.

Nowadays, ontologies have become a significant instrument to model the context in pervasive systems. As we have based the methodology we propose in this thesis on an ontology, we focus, in the following sections, on the ontology-based approaches that have adopted ontological models to handle the context.

---

### **III.2.1 Ontology-based Context Modelling Approaches**

We notice that ontologies are particularly suitable to describe the context data and information used in our daily life and represent them by data structures that can be processed by computers. Moreover, Ontologies provide a uniform way for core concepts specification and enable contextual knowledge sharing and reuse in a given domain. In the next section, we report about the most relevant ontology based context modelling approaches in order to establish a contrast with our research work.

---

#### **III.2.1.1 Service Oriented Context-aware Middleware**

The SOCAM project, Service Oriented Context-aware Middleware, aims to develop a prototype of context-aware services in smart environments (Gu

2004), (Gu 2005). The middleware of SOCAM is based on ontologies to model the context. As a first step, the SOCAM middleware uses a high level ontology to describe the information characterizing in general the applications execution environment. Then, a specific application domain ontology must be defined, e.g. an ontology that describes a user in a smart home. The SOCAM architecture comprises several other components such as context provider, context interpreter and context database to be used for capturing contextual information, representing these information by OWL, running a rule-based reasoning on the OWL context representation, and finally for storing the different instances in the context database. In addition, the SOCAM middleware integrates context-aware services using the information stored in the database to modify the services behavior according to the current context.

### III.2.1.2 Context Broker Architecture

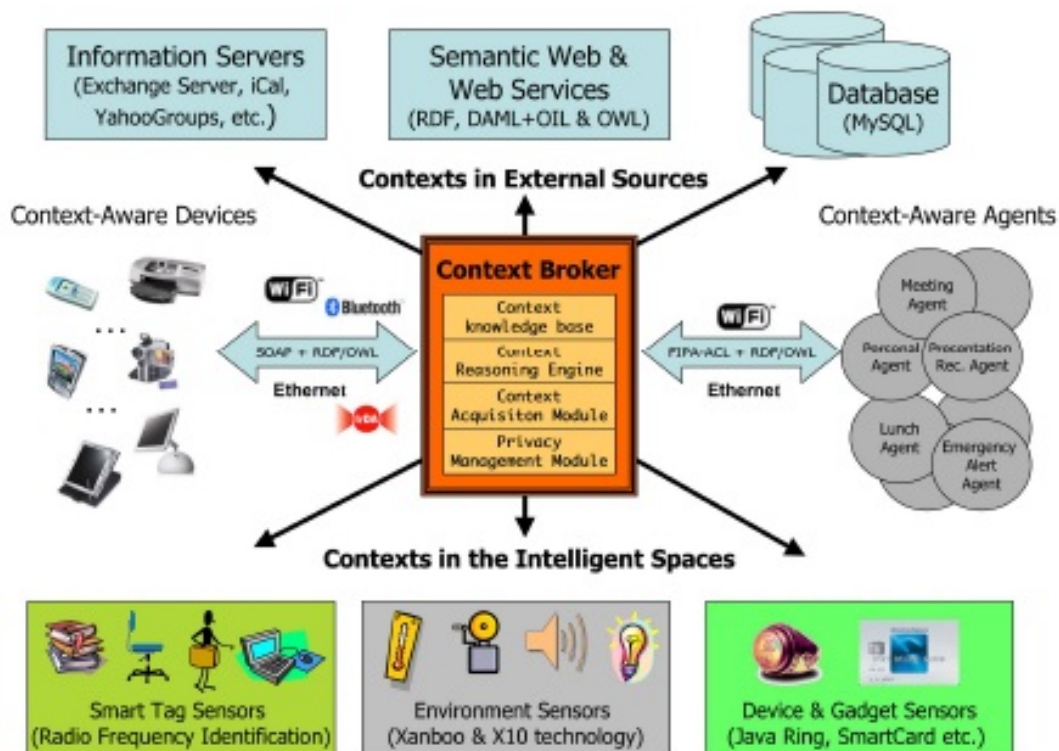


Figure 7. The Context Broker Architecture CoBrA. From (Chen 2004).

The Context Broker Architecture (CoBrA), Figure 7, proposed by (Chen 2004) implements an agent oriented approach for context-aware systems in smart environments, i.e. smart homes, smart vehicles, smart meeting rooms, etc. The core element of the CoBrA architecture is an intelligent agent called context broker that maintains a shared context model for a community of agents, services, and sensors.

It acquires and interprets the contextual information coming from different sources. It can also detect and resolve interpretation conflicts of the context stored in the shared model. Based on the security policy defined by each user, the broker provides the confidentiality of the data exchanged among users. The intelligent agent relies on inferences based on logical rules for context management and interpretation (Chen 2004).

---

#### III.2.1.3 Context Adaptation and Semantic Interoperability

The vast majority of the existing contributions in the field of context-awareness have explained the way to capture, interpret and provide the contextual information to the application without considering how applications can adapt to context. In addition, the existing works mainly focused on an incremental creation or prototyping of context-aware applications, by incorporating the adaptation code in the business logic layer. This limits the capability to take into account new contexts that were not expected during the application development (Chaari 2007). Other frameworks have been introduced by (Strimpakou 2006) and (Malatras 2009), to enable a transparent and unobtrusive contextualization based on three components for users, devices, and space context management, as well as a policy based management. According to Malatras, the adaptation of pervasive environments is guided by the enforcement of policies that are triggered by context information, e.g. battery levels falling below a certain threshold for a particular device. Thus, it is not necessary to change existing software applications to realize pervasive behavior, but instead, configuration changes guided by context information and policy rules can satisfy a great degree of the desired adaptation.

---

#### III.2.1.4 Other Context Modeling Approaches

CONON (CONtext ONtology) defines generic concepts concerning the context in pervasive computing environments such as “Home”, “Office” or

“Vehicule”, and provides extensibility for adding domain specific concepts (Wang 2004). In his performance study of the prototype implementing CONON, the author quantitatively evaluated the feasibility of logic-based context reasoning for non-time critical applications in pervasive computing environments.

In order to provide collaboration among ubiquitous computing devices, (Christopoulou 2005) has proposed the so called Gadgetware Architectural Style (GAS) Ontology, having the goal to provide a common language for the communication and collaboration among the heterogeneous devices that constitute the environments. It also supports the service discovery mechanisms required by the pervasive computing environments.

A semantic middleware which aims to provide ubiquitous computing applications with high level contextual knowledge has been proposed by (Chibani 2008). The proposed middleware is based on agent entities offering reusable services for context semantics discovery, capture and aggregation. To model the contextual knowledge aggregation, the proposed middleware integrates a services composition mechanism with dynamic discovery of the available service agents.

According to Lee et al., for independent systems to manage, share, correlate and reason over context, the contextual information must be modeled in an homogeneous fashion (Lee 2009). The authors propose an hybrid approach for modeling contextual information that incorporates the management and the communication benefits of traditional object oriented context models, while also taking advantage of the semantic and inference benefits of ontology based context models. Similarly, another hybrid approach to model context has been proposed by (Dedefa 2007). The platform proposed by Dedefa provides a collaborative context-aware service based on an hybrid context management model. It performs reasoning and decisions based on context data, context semantics, and related rules and policies (Dedefa 2008).

Gerogalas et al. have employed the OMG’s Model Driven Architecture for context modeling and proposed a Model Driven Integration Architecture (MDIA) which has been designed to integrate GUI models, processes models and ontology models to build a platform independent Context Aware Application (CAA) model representing user interfaces,

business logic, and technology-based context data involved in the context aware applications (Georgalas 2007).

The situation awareness issue in social networks has been tackled by (Sutterer 2008) by presenting an ontology-based user profile model that allows users to set up situation aware social networks, by controlling how reachable they are for specific categories of people in a given situation.

In the robotic domain, a robot system has in general three functional components of sensing, processing and acting. Therefore, a software infrastructure is required to support context-aware applications development and execution. Hong et al. have used an ontology-based context modeling and reasoning approach to build a context-aware infrastructure for a network-based intelligent robot system (Hong 2007).

---

### **III.2.2 Current Context-awareness in Healthcare and Telemedicine Applications**

Context-aware computing is a research field which is, with the wide deployment of pervasive computing infrastructures, becoming more and more important. It often refers to healthcare as an interesting and rich application area. Conversely, the eHealth research community is increasingly interested in the advantages of context-aware computing to meet the requirements of telemedicine applications. Among the main types of contextual information that are considered relevant to the user, are the profile, activity, time, and location, which are designated as primary context.

Telemedicine workers are highly mobile, which means that they constantly change location to perform their daily tasks, which includes patient assisting and transferring. The information required by the telemedicine actors being highly dependent on their location, role, time and task, it is thus essential to provide information that is relevant to the context of use of both the patients and the healthcare professionals. Therefore, telemedicine has been recognized as an important and promising field of context-aware research. In the literature, several research proposals have recently appeared that suggest models and solutions which are able to record and analyze the patients' behavior patterns to monitor the user's mobility and to assist individuals with special needs in their daily activities, such as self-care. The basic assumption of these research works is based on

the visibility of the patient's context information, e.g., the patient's physical conditions and cognitive status, to determine the best suitable assistive operation to implement, e.g., remind elders to take the prescribed medicines, avoid the risk of falling in elderly community-dwellings (Hewson 2007). The vast majority of available solutions relay on sensors platforms to obtain context information about the users' conditions. The set of data that should be monitored depends on the application scenario and may include, for instance, user location and behavior patterns.

An initiative was conducted by (Lee 2006) to enable the management of emergency departments by integrating location-aware services in the hospital information system. The PEACH (Pervasive Environment for AffeCtive Healthcare) framework proposed a context-aware middleware-level solution that integrates bio-sensors able to detect alteration of the patient's psycho-physical conditions, to aggregate the sensed information and to detect potentially dangerous situations for the patient (Taleb 2009). Similarly, for the early detection of risk situations in an intensive care unit, (Leite 2009) have developed a service-oriented context-aware middleware for real-time data management using mobile devices. The proposed middleware tracks data in real time, e.g., vital signs, provides services such as real-time access to databases from the mobile device, constant database updating from continuously sampling data acquisition devices, and sending of alerts to mobile devices.

The issue of context-aware computing based financial and human resources allocation in healthcare systems has been addressed by (Hsieh 2007). The author has employed Petri Nets to model the resources allocation and to monitor the medical processes in healthcare systems. Recently, (Anya 2010) proposed a conceptual model for context-awareness in e-health, taking into account the different contexts involved in the healthcare process, such as context of information resources available for supporting decision making, context of knowledge providers and healthcare professionals working in different organizations, context of patient receiving treatment, and context of the end users. The goal of Anya's work was to integrate explicit information and tacit expert experiences across various work domains into a knowledge resource adequate for supporting the operational context of the work setting in which it is to be used.



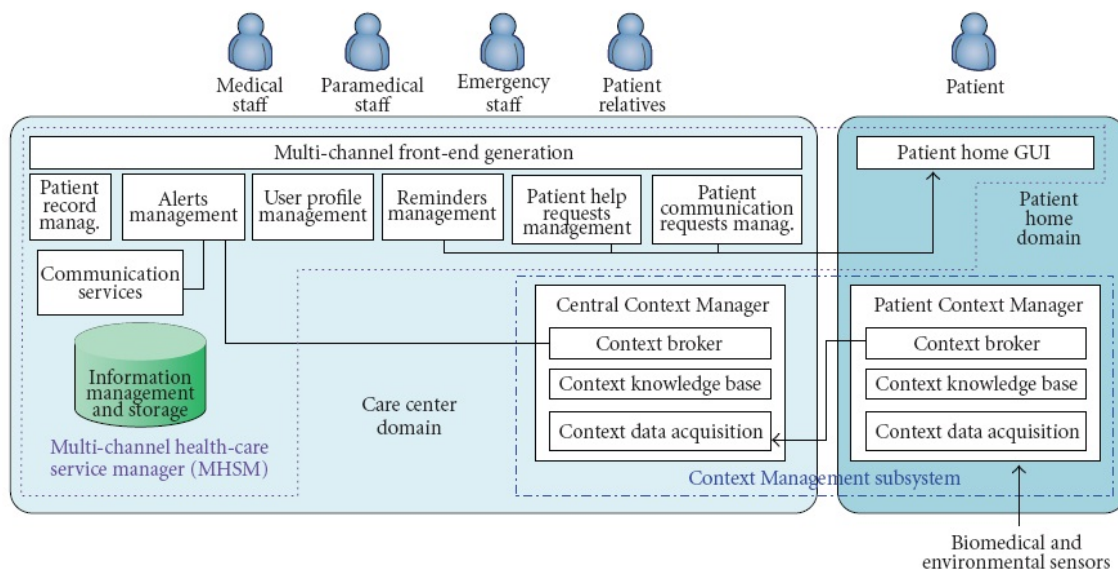


Figure 8. Snapshot of the ERMHAN's Architecture. From (Paganelli 2008).

Sain et al. have applied context-awareness to design a personal health information system named MUHIS (Middleware for Ubiquitous Healthcare Information System), which gathers sensor data from personal healthcare devices and sensors, and then transforms these data into useful information (Sain. 2010). The conceptual architecture of MUHIS contains a hardware layer that is responsible for retrieving raw context data by querying the sensors, a middleware layer responsible for providing context to the third layer which is the application layer.

Paganelli et al. have conducted a research on the design and development of a service-oriented architecture for continuous care provision leveraging on context-aware and mobile technologies, called Emilia Romagna Mobile Health Assistance Network (ERMHAN) (Paganelli 2008). The proposed architecture aims to enable the development and delivery of a set of care services which allow the patients to be assisted at home in a familiar environment and which support the activity and mutual collaboration of the care providers who are involved in patient care and assistance. Figure 8 shows the components of the ERMHAN system architecture.

---

### III.2.3 Analysis of Context Research Works

The research community has not agreed on the best way to model context and architectures to support its use. Despite the diversity of the context-dependent applications in the healthcare and telemedicine domains, we must admit that there is still a lack of systems development and implementation in this area. Most of the previous projects are prototypes and real applications are still difficult to develop. This is obviously due to the difficulty of managing distributed and complex context-aware systems. Even if we could identify some interesting prototypes, there still exists a gap between the requirements expressed in the literature and the developed prototypes. This seems realistic given that context modelling and computing in healthcare and telemedicine is a new, evolving research area.

An interesting review of context awareness in healthcare has been elaborated in (Bricon-Souf 2007). The aim of this survey is to derive an objective view of the actual dynamism of context awareness in healthcare and to identify strengths and weaknesses in this field.

Concerning systems contextualization, we notice that the underlying architecture, middleware, systems and applications have to implement additional components in order to be adapted to the context. Consequently, there is still a need for a transparent solution operating in the background and that exploits the programming interfaces of existing applications, e.g. browsers, navigation panels, network protocols, services, database systems, etc, to provide ubiquitous computing functionalities to the end users.

Object-oriented modeling techniques provide support for storing, managing and accessing context in a scalable manner. However, they lack semantic and ability to reason over context. On the other hand, ontologies enhance context with semantic for knowledge representation, autonomic processing and reasoning. However, research in the fields of efficient ontology storage, access and management is still at an early stage. We notice that researchers in the pervasive computing domain believe that the scalable management of context information is not a core feature of ontology-based context management. They consider that object oriented and data modeling techniques are still more advanced in these tasks.

Therefore, an ontology based approach alone is not sufficient to effectively model context in pervasive environments.

### **III.3 Task/Activity-Based Computing**

Task/Activity-based computing is a computing infrastructure which supports suspending and resuming activities across heterogeneous execution environments and supports activity-based collaboration (Bardram 2004). A task represents a user's computing intention. A task may range from simple, short-life activities such as sending an e-mail, to complex, long-life activities involving multiple steps, such as patient orientation to an hospital. A task is platform-independent and application-independent. This paradigm represents a new view towards system building. The basic observation here is that, having an explicit representation of the computing tasks used to build the software services, it is possible to build middleware infrastructures that enable automatic instantiation of the task context into a collection of dynamic computing devices with a minimum amount of user interference.

Software services should be built with the inherent expectation that they are going to be used with other services to achieve high-level goals. Users should interact with computing devices in terms of high-level tasks instead of individual services or applications. The following elements constitute Task-driven computing:

- Application/environment-independent specification of tasks.
- Explicit management of computing context, which includes task specifications, task states and user preferences. A user's computing context is globally accessible and updated automatically as progress is made on tasks.
- Services with well-defined interfaces and meta-data specifications. Services are required to have the interfaces supporting automatic configuration. Legacy applications can be wrapped to support these interfaces.
- Session management: automatic service configuration, composition & coordination. Services are automatically composed and configured to accomplish user tasks. The system automatically adapts service configurations when service availability changes.

- Proactive guidance. When a task can't be executed because of resource restrictions, the system offers suggestions to the user to help him accomplish what he wants to do.

The idea of Task/Activity-based computing is that the whole computing environment is focused on a single activity. Rather than opening several programs, one opens an activity, and this activity is linked to the programs necessary to make progress on that activity. There are five core principles for Activity-based Computing:

- Activity-centered resource aggregation and bundling resources together according to activity.
- Activity suspension and resumption, saving the state of the activity.
- Activity roaming, accessing the activity from multiple locations.
- Activity sharing, synchronous and asynchronous multi-person access to an activity.
- Activity awareness, location-based computing, resources in the activity are made available based on the location of the user.

In the following, we highlight the most pertinent research works related to task-based computing.

The Aura project, (Aura 2002), (Judd 2003), introduces Task-level computing and outlines an architecture that allows computing context to migrate between devices. It relies on the availability of the peoples' location information to provide location-specific services. Aura is specifically intended for pervasive computing environments involving wireless communication, wearable or handheld computers, and smart spaces. (Bardram 2005) proposed a framework of Activity-Based Computing (ABC) which addresses the mobility and cooperation in human work activities. The aim of this framework is to support human activity by managing its collections of work tasks on a computer, to support the mobility by distributing activities across heterogeneous computing environments, and to support asynchronous collaboration by allowing several people to participate in an activity. The ABC approach has been designed to address activity-based computing support for clinical work in hospitals (Bardram 2009). In addition, a role-task model enabling users to navigate and perform real-world activities has been proposed in (Fukazawa

2006). The goal of the proposed model is to recommend tasks relevant to the current role of the user and his current location.

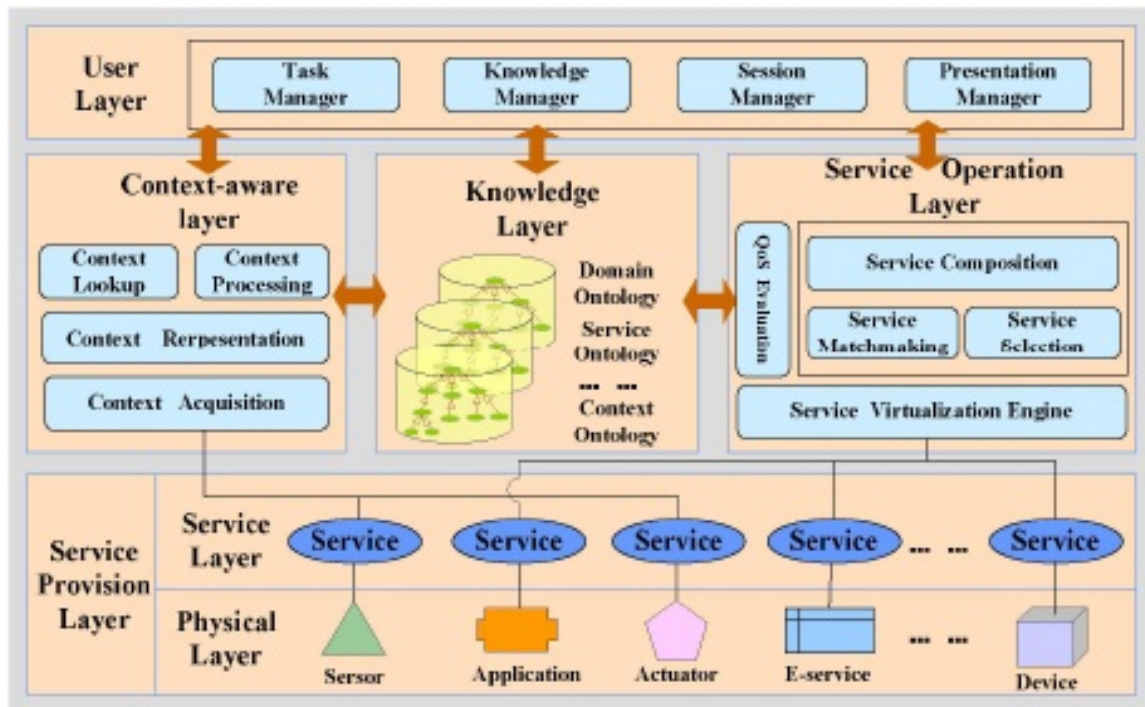


Figure 9. The Structure of U-TCA, from (Jiang 2008).

In pervasive computing, services discovery is an essential technique to provide the applications with the appropriate resources. It aims to guarantee transparent development environments to application developers. Kim et al. have proposed a framework that helps applications to find appropriate services even though the required ones are not available or not found (Kim 2006). They proposed a fine-grained effect ontology which is used to reasonably evaluate the functional similarity among different services and a policy-based query coordination which is used to dynamically apply different resources constraints according to the involved human activities.

The notion of task has also been used in the web services composition domain. An ontology based on OWL, called OWL-T where T stands for Task, has been proposed by (Tran 2007). The OWL-T can be used for users describing and specifying, formally and semantically, their needs. It aims at facilitating the system modeling without regarding the

details of the low-level and technical aspects of the underlying infrastructure. Jointly, in order to optimize the discovery process and to save time and effort of the users in task computing, (Huang 2008) has proposed a web-based, semantic mechanism to filter and rank the matching services according to their dynamic context information to provide the most suitable and relevant services for users. From the query optimization point of view, a synthesis of the main dynamic optimization methods has been realized by (Morvan 2009).

Since a user task in a high abstraction level represents the user's intention, there is a gap between the user task and the services surrounding the users. To fill this gap, Jiang et al. have proposed a user centric task computing architecture for heterogeneous and dynamic pervasive computing environments called U-TCA (Figure 9) which focuses on sensing and abstracting the user requirements using a task model as well as a middleware to mask the heterogeneity of the computing environments (Jiang 2008). Finally, Sasa et al. (Sasa 2008) have proposed a service oriented architecture framework for human tasks execution, which improves their execution by automating and semi-automating decision making based on ontologies and agent technologies.

---

#### III.4 Discussion

The models and systems cited above present solutions based on the data captured by distributed sensors in well equipped pervasive environments, using common scenarios and standardized protocols. For example, the multi-agent based telemedicine systems modeling approaches proposed by (Koutkias 2005), (Takahashi 2007) have only addressed some specific scenarios and they did not clarify how the different agents can access and share the resources. Moreover, the scenario of a patient tele-assistance in hostile environments, where there are neither sensors nor pre-defined protocols to exchange data, has been neglected by most of the existing context-aware telemedicine systems. That is why we have focused in our thesis on the diversity and specificities of healthcare scenarios occurring in non favorable and non equipped environments.

Because the emergency scenarios are largely various and because of the diversity of illnesses, the data to be transferred in each scenario are partly heterogeneous, and thus the emergency telemedicine applications

that have been reported in the literature are limited to specific clinical situations. For this reason, we still need a telemedicine system that takes into consideration the scenario diversity. Moreover, although the RES@MU platform developed by (Ifremmont 2007) is interesting in terms of medical services provided in case of an emergency, it however lacks a strategy to manage the availability and capability of the used resources, i.e. logistical, material and even human resources. Similarly, for the system developed by (Kang 2007) and (Jang 2007), it is not clear how the proposed system could behave in case of the non availability of the emergency physician who should provide directives to the ambulance team. More generally, the most current telemedicine systems do not handle the resources management problem. They do not consider the continuously changing conditions, i.e. availability, accessibility, capability, etc, of the heterogeneous resources required to perform different telemedicine tasks. In this thesis, we have treated all the above issues to provide a more general and knowledge-based solution taking into account the dynamic context of the resources and environments.

The use of an ontology by the ARTEMIS project and the other ontology-based telemedicine systems cited in section III.1.4., are restricted to the data semantic interoperability level. The other ontology potentialities have not been very well exploited by the proposed approaches. In ONOF-PAS, we will maximally take profit from the ontology features at both the semantic interoperability level and the inference level to deduce new object type properties between the different concepts by using SWRL rules.

The architectures and conceptual models proposed by (Taleb 2009), (Leite 2009), (Hsieh 2007), (Paganelli 2008) and (Anyà 2010) are based on Service-Oriented Architecture to provide healthcare services in static environments, e.g. patients' homes, intensive care units, etc. Conversely, in our ONOF-PAS framework we will focus on the Task notion, which has not been used before for the modeling of context-aware healthcare systems, to enable healthcare professionals to carry out multiple tasks in dynamic environments. Additionally, the works cited above, ignore the need for knowledge management in order to provide the healthcare actors, who lack the knowledge required to perform different telemedicine tasks, with efficient decision making support tools, as we will do in this thesis.

---

### III.5 Conclusion

Overall, in this chapter we have presented the results of the state of the art that we have respectively achieved in the telemedicine systems, context modeling approaches, and task-based computing domains. We have presented the different systems, applications, and projects related to telemedicine. As the domain of eHealth is extremely large, it was not possible to cover all eHealth and Telemedicine topics and aspects in the present state of the art. Nevertheless, we have presented the most interesting and related systems and projects proposed by the relevant research community and as we are interested by the ontology aspects, we have focused, in this literature review, on the approaches that have used or integrated ontologies to model telemedicine systems and context.

Finally, as our proposed solution, which is presented in the next chapters, is founded on the task-based computing paradigm, we have provided the reader with a simple background about this notion and about also the several research works related to task-based computing.





# *Chapter IV*

## *ONOF-PAS Framework*

### *Methodology, Construction, and Formalization*



# Chapter IV    ONOF-PAS Framework Methodology, Construction, and Formalization

## Overview

In this chapter, we firstly remind the reader of the main problem that we are trying to solve through this dissertation. We also define the principal notions, i.e., pervasive task, contextual situation, and decision making process on which our proposal is based. Then, we present our methodology to design the ONOF-PAS framework and the included knowledge meta-model construction. After that, we will instantiate the meta-model in the eHealth domain and we will formalize the instantiated models using the Ontology Web Language OWL. At the end of this chapter, we will provide details on the rules base construction and formalization using the Semantic Web Rule Language SWRL.

---

### IV.1    Rationale and Scope of the Problem

In a given domain and within a business process, actors having different attitudes and skills, perform multiple tasks that depend on their profiles. These tasks may require heterogeneous resources, i.e., devices, services, knowledge sources, etc. Generally, these distributed resources and knowledge are owned or managed by different organizations. In some application scenarios, the resources conditions, in terms of availability, accessibility and capability, change constantly. On the other side, each task has its own specifications and may be run in dynamic environments, and sometimes in non equipped or hostile environments. Additionally, in pervasive systems application domains, such as telemedicine, data communication services should be as convenient as possible and adapted to the context of use. However, the diversity and specificity of applications and use contexts make services adaptability and composition a complex process.

The aforementioned context limits the ability of pervasive systems to provide users with a high quality information, especially in application areas such as the eHealth domain where the decision making processes may be very complex, are usually not expressible by exact algorithms, may

require intelligent compromise solutions combining both the knowledge about the queried objects and their context, taking into account the lack of precision, the risks attached to the objects, and some a priori knowledge about the potentialities of each available resource. Thus, the main question is:

How to adapt the decision making process to the contextual situation represented by the pervasive tasks specifications, the users profiles and aptitudes, the environment conditions, and the constantly changing resources required to perform the pervasive tasks ?

Therefore, to face this challenge of decision making in non predefined scenarios taking place in hostile environments and to meet the users' needs diversity in different contextual situations, there is an essential need of models able to capture knowledge of the following elements:

- The actors who perform multiple tasks in a given domain and in different contexts.
- The tasks objectives.
- The conditions of the objects treated by the tasks.
- The heterogeneous types of resources required for performing the tasks.
- The knowledge related to the organizations that own or manage these resources.

We recall that the main goal of our thesis is to provide a solution to support decision making processes in non pre-defined scenarios that take place in inimical environments (e.g. decision making for patient orientation in hostile environment such as high mountain resorts).

---

## IV.2 Pervasive Tasks and Contextual Situation Definition

---

### IV.2.1 Pervasive Tasks

The notion of Pervasive Task will be often used all over the next sections and chapters. For this reason, we prefer to start the present section by defining the Pervasive Task notion within the scope of this dissertation.

One of the main motives of focusing on the definition of a *Task* in the healthcare domain resides in the fact that the tasks in the healthcare or

the telemedicine domain are independent, flexible and achieved by an healthcare professional to accomplish a specific objective, while in other domains, e.g. the industrial domain, tasks may not be independent which means that the execution of one task may rely on the results of another task in the same workflow. In the healthcare domain, actors are not forced to respect a defined workflow or process. This is the major reason that conducted us to focus on the task notion in our modelling process.

In the common language, a Task is “a usually assigned piece of work often to be finished within a certain time”. According to our experience in the telemedicine domain, the task notion is used in terms of mission to be accomplished by the healthcare worker, which means a human task. In computing, a task is “*a set of program instructions that are loaded in memory and executed by the processing unit*” (Stair 2003). As we treat the tasks in pervasive computing, we should provide a new definition adapted to the pervasive environments. For elaborating a definition of a Pervasive Task, we have tried to find a compromise between the definitions of a service and a process. Indeed, the objective of this compromise is to provide a high level abstraction of the service and process notions. In other words, a Task definition shall provide a general description of the activities and services defined in a given process. We thus propose the following definition of a Pervasive Task:

A Pervasive Task is a set of related activities, which uses business or data services, performed by a domain actor and executed by a pervasive information system to provide the user with data that are relevant to his request and needs.

For example, in several telemedicine scenarios, to provide the service continuity when a patient is transported from the accident site to an hospital, then redirected from one hospital to another, and moved from an hospital to home, healthcare professionals need to perform multiple tasks, e.g. data search, data retrieval, messages exchange, etc. These tasks form one or more processes of the global patient hospitalization scenario. Additionally, these tasks shall permit to trace the used services and the exchanged data in each process. Moreover, the data and information gathered during the execution of these tasks should be stored in a final file or transferred directly to the patient’s EHR.

The input/output parameters of each task shall represent data such as, patient personal data (name, age, sex, etc), location data (address, city, country), medical data about the disease (cardiology, neurology, dermatology, etc), and resources data (data about the different required human and material resources). A process will be composed of one or more pervasive tasks that may invoke one or more services provided by different organizations, e.g., in the healthcare domain, these organizations will be hospitals, medical centers, insurance companies, etc.

In our ONOF-PAS framework, we will describe the Pervasive Tasks in an intelligent manner through ontological models providing the pervasive task specifications, as we will see later on in this chapter.

---

#### **IV.2.2 Contextual Situation**

A contextual situation is characterized by the whole context elements, i.e. the user profile, communication device, location, environment, etc, see section II.3. Each context element gathers a set of context parameters. Therefore, at a given moment, the contextual situation is defined by the values associated to these context parameters. Any modification of one of the context values provokes a transition to another contextual situation. This transition may impact the application behavior by triggering other views on its data and functions.

For example, a contextual situation might be specified by the following parameters: the user profile, i.e., all information concerning a user such as specialty, preferences, access rights; the communication device he/she has, e.g., PDA; the user location including the geographical location and logical location, i.e., IP address; the task he performs, e.g., data search, data access, data communication, etc; the resources required to perform the task, e.g., a specific equipment or device, a service, knowledge sources, etc; the organization profile which has these resources available. The previous main parameters may also be more specific, for instance, the technical characteristics of the communication device, e.g., screen size and resolution, memory size, embedded telecommunication technologies characteristics such as the bandwidth, the installed operating system, etc. Concerning the environment, other parameters might be of value to perfectly perform some specific tasks. Indeed, environmental parameters might enrich the task model with useful information. For instance,

information like ambient temperature, moisture, atmospheric pressure, wind speed, and the road traffic status, could be interesting to the user depending on the application, so that he can perform the task he wants in almost perfect conditions. The more we have the previous contextual information available, the better the application adapts to the defined context and, consequently, the better the users can exploit the application.

Context parameters can be static, i.e. do not change during a user session, or dynamic, i.e. the context parameter value may vary in the same user session. Context-aware applications must be capable to detect any context parameter changes and be adaptive to this changing. We can store the context parameters in different ways, i.e., in a Relational Database (RDB), in an Object Relation Model (ORM) XML schema or in an ontology, as we will see in this chapter.

---

### **IV.3 Knowledge Integration for Enabling Decision Making Processes**

A decision is one of the central dimensions of organizations management. The decision making steps, formalized by (Simon 1991) and known as IMC model (Intelligence, Modeling, Choice), are the following: problem intelligence, problem demarcation and some factor to be taken into consideration, i.e., modeling, identification, and evaluation of all feasible alternative solutions, choice, selection criteria development, solutions prioritizing and solution selecting.

In knowledge engineering, there are four main steps which have been adopted by researchers to model Situation-handling (Wiig 2003). These steps are: Sense-making, Decision-making, problem-solving, implementation and monitoring. We have organized and represented the aforementioned steps as shown in Figure 10.

For decision support and context-aware systems, some contextual parameters must be provided as input data to characterize the contextual situation (i.e. data about user, location, resources, etc). These contextual situation information may be provided either by users or automatically acquired from sensors installed in the systems environment. The system deduces knowledge based on these input data through additional forms of knowledge such as concepts, facts and rules. The knowledge produced by every contextual situation will be structured in new instances that may be



reused in a similar situation. Thus, our objectives in knowledge modeling are to:

- Capture and store knowledge about the main entities in a given domain, e.g.; Actor, Task, Object, Resources, Organizations, etc.
- Provide an intelligent management of the different tasks and processes by enabling a rule-based reasoning on the available knowledge.
- Produce new knowledge related to the domain entities.

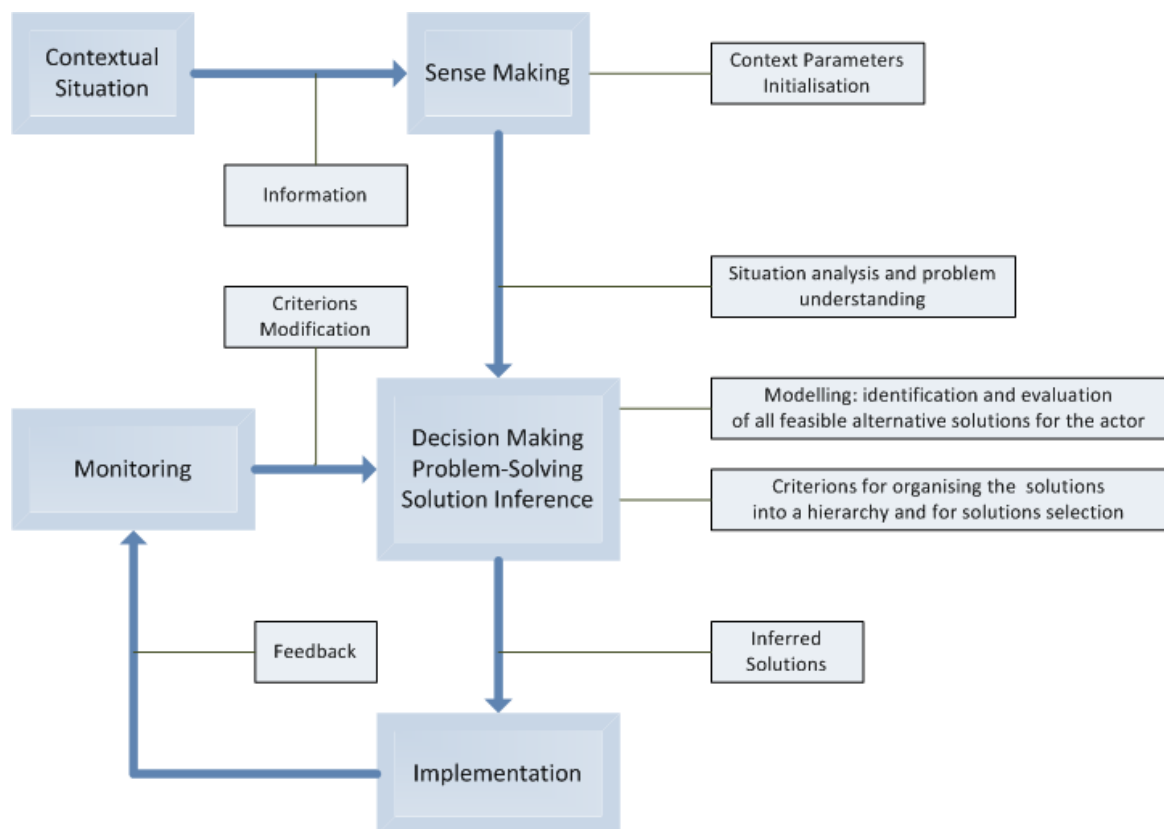


Figure 10. Main Steps of Situation Handling Modeling.

But, to integrate knowledge in pervasive systems architectures, it is essential that we represent all domain concepts and their interrelations by coherent and integrated models. In the next section, we present our methodology to build such knowledge models.

#### IV.4 ONOF-PAS Design Methodology

To realize the knowledge-based framework, we have defined several consecutive phases to construct ONOF-PAS. Figure 11 depicts these phases, together with the most general steps involved in each phase.

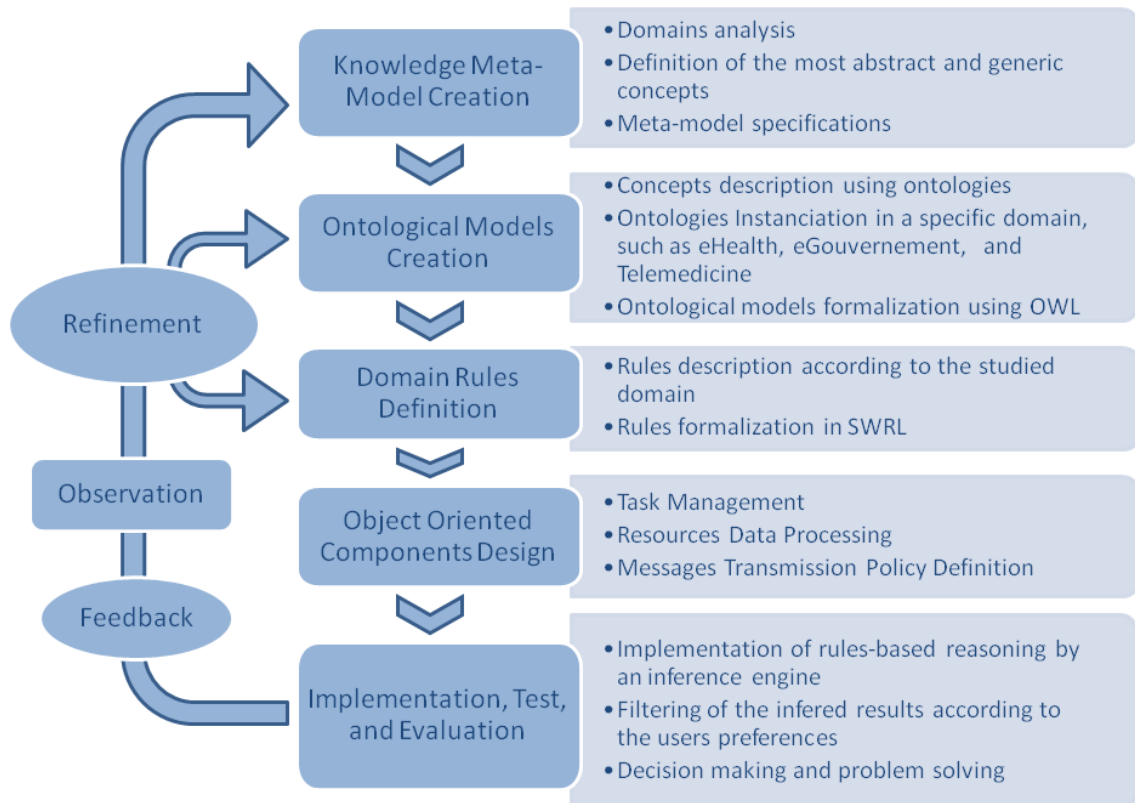


Figure 11. The Main Phases of our Methodology for ONOF-PAS Construction.

The goal of the first phase is to build a basic, generic, and domain-independent knowledge meta-model. Mainly, this phase starts by an abstraction process of the available knowledge in order to identify the most general concepts, i.e. Actor, Task, Process, Service, Resource, Organization, etc, that can be instantiated in several domains such as eHealth, eGovernment, eBusiness, etc, to take account of the specifications, the purposes, and the execution environment of the application functionalities and services.

Each instantiated model may differ from one domain to another. For example, in the eHealth domain, the actor ontology model can be extended to include the Healthcare Professional class, whereas in the eGovernment domain, it can be extended to include an Officer class.

The second phase aims at describing each concept in the knowledge meta-model by using an ontology which can be instantiated in different domains. In this phase, we also formalize the ontological models using one of the existing standard ontology description languages, i.e. OWL. This phase is an essential step in order that the knowledge represented by an ontology can be interpreted by a computer. It will indeed be used by the reasoning processes in order to infer new knowledge. As we have seen in the second chapter, there are different types of formalisms and languages which allow to represent formal knowledge by an ontology, e.g. XML, Resource Description Framework Schema (RDFS), and Ontology Web Language (OWL).

In the third phase, we define rules to manage the different tasks that should be performed to meet the user needs in different contexts of use. This is a decisive phase for designers who shall here specify the behavior of the knowledge-based system by means of rules, restrictions, and conditions. The fourth phase concerns the creation of object oriented components, e.g. tasks manager, profile manager, resources processor, etc. Additionally, interface components, e.g. messaging manager and GUI components, are also modeled in this phase.

In the last phase, we implement the whole set of the ONOF-PAS architecture elements and perform several tests to verify the proper functioning of the system. As the concepts defined in the first phase are represented by ontological models, we propose to perform rule-based reasoning to infer new facts. Various existing logic reasoning mechanisms can be employed here in order to deduce decisions that will support processes and tasks execution in pervasive applications as we will see in a next section.

The present design methodology also includes feedback loops to verify if the inference engine requires changes in the ontological models specifications. In the latter case, we should effectuate a refinement process to validate the knowledge meta-models and ontological models in order to complete the knowledge base by inserting more or less complete sets of additional knowledge instances. In the next section, we provide more

details about the design of the knowledge meta-model and of the ontological models that are the corner stones of the first two phases of the ONOF-PAS construction methodology.

---

## IV.5 Knowledge Meta-Model Definition

The ONOF-PAS knowledge meta-model is composed of a set of generic models representing the main entities of pervasive applications as well as of the associations which connect these models. According to our design methodology presented in the previous section, we will start by the main and general concepts identification step.

---

### IV.5.1 General Concepts Identification

Concepts definition is a principal step for ontological models construction. This step starts generally by the knowledge abstraction in order to identify the generic concepts that can be instantiated in several domains. This involves the specification of the reasons why the ontological model is being constructed, as well as its intended purposes. In this context, a series of refinement operations is performed to extract the concepts, their attributes, and the relations between these concepts. The concepts should be domain-independent and represent the key entities involved in the realization of different tasks and processes in pervasive systems and applications.

According to the Business Process Management (BPM) formalism, each *Process* is composed of one or more *Tasks* which invoke different *Services* during its execution. A *Task* is performed by an *Actor* who is located in a given *Location*. A *Task* treats or concerns a given *Object* which has its own *Conditions* and *Status* defined by a set of *Parameters*. Furthermore, a *Task* requires several *Resources* in order to be achieved and generates one or more *Messages* at the end of its execution. The required resources are owned or managed by different *Organizations*.

To constitute our knowledge meta-model elements, we have identified the following generic concepts: *Task*, *Service*, *Process*, *Actor*, *Organization*, *Location*, *Resource*, *Object*, *Status*, *Parameter*, and *Message*, where each of these concepts is represented by its own model according to the existing standards.

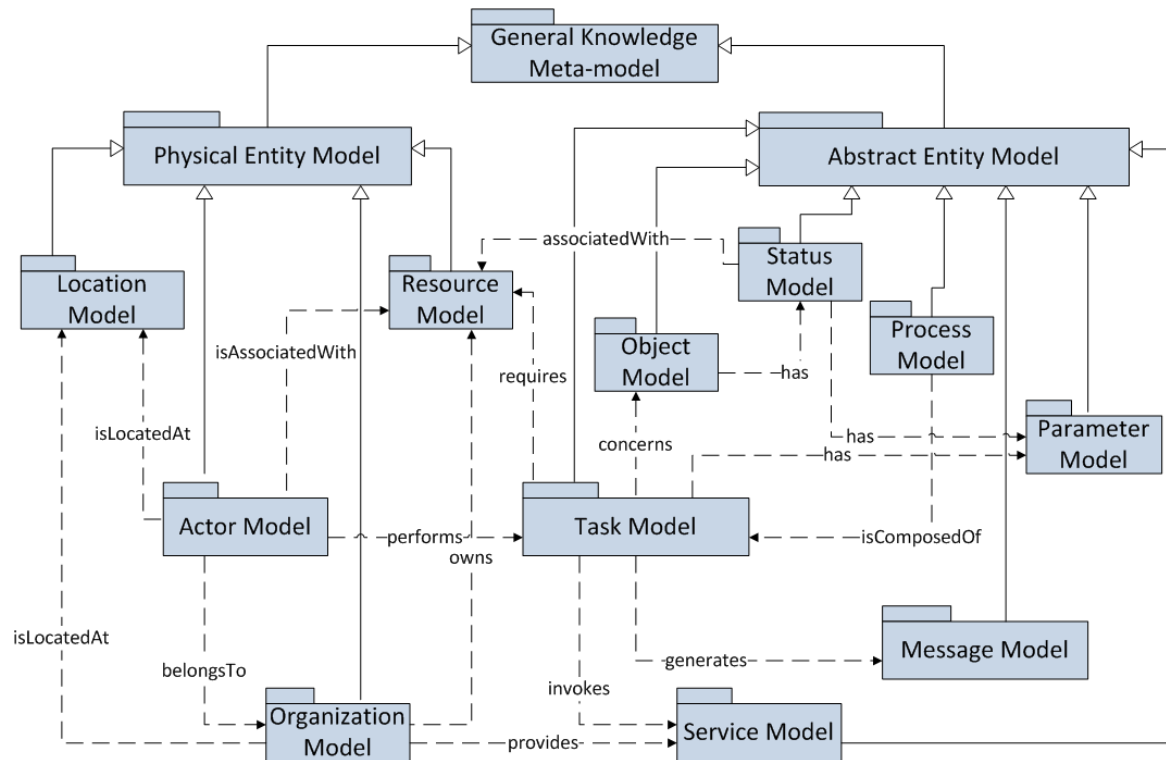


Figure 12. Graphical Representation of the Knowledge Meta-model using the UML Notation. Broken arrows represent an association between two models while straight arrows represent a generalization relation.

To make the meta-model better more organized, we have classified the entities as physical, e.g. Actor, Organization, Resource, etc, and abstract, e.g. Process, Task, Service, Messages, Parameter, etc. Figure 12 shows a global view of these entities and their associations.

We have then defined a number of interrelationships among the entities, i.e., *Actor performs Task*, *Task concerns Object*, *Task requires Resource*, *Organization owns Resource*, etc, (Figure 12).

The knowledge meta-model displayed in Figure 12 thus represents and organizes the knowledge about existing pervasive applications into three main dimensions constituting the key stones of any pervasive application or service framework. These dimensions are:

- The Actors and organizations profiles in a given domain.
- The tasks performed by the actors.
- The different human, communication, and material resources required by the tasks.

Accordingly, ONOF-PAS will be scalable, because each of the mentioned elements contains large amount of information. Furthermore, it will empower the possibility to aggregate information at different levels of abstraction, and consequently, it will constitute a key component to build knowledge-based systems. Indeed, ONOF-PAS already encompasses the different information related to existing pervasive applications and services. In addition, our knowledge meta-model provides a common and generic repository structure that will facilitate the exchange of data currently contained within individual tool models of the existing software solutions.

---

#### **IV.5.2 Meta-Model Formalization Using OWL**

Several data model standards (e.g., object model, relational model, XML model, etc) can be employed to describe data and knowledge. However, in knowledge-based and artificial intelligent systems, data models should support rule/case based reasoning. We therefore have described each model of our proposed knowledge meta-model by means of an ontology that can be extended and instantiated according to the specifications and needs of various applications in different domains. In the next section we justify our choice to use ontologies for data modeling.

---

##### **IV.5.2.1 Why Ontologies?**

Nowadays, information systems conception is increasingly based on ontologies. The ontology benefits tempted us to adopt it in our design methodology. Indeed, ontologies are flexible, scalable, personalisable and evolutionary. On the other hand, since XML became a popular data exchange protocol and the fact that the ontology description languages RDF and OWL are based on XML, this makes the ontology model transformation feasible in terms of the representation of the data stored in the ontology, in different formats. Moreover, we can exploit various existing logic reasoning mechanisms to deduce the knowledge which shall support the decision making processes.

In fact, integrating an ontology serves us in two important points. The first one is the fact that ontologies ease the knowledge model description through the concepts and interrelations. Thus, it will ease the process of analyzing knowledge by creating specific rules and profiles. The

other key point is that ontologies support the semantic interoperability problem solving based on the fact that it is possible to align two ontologies. Consequently, ontologies can be used to facilitate the implementation of data exchange standards through semantic mediation. For all the reasons mentioned above, we have decided to make our proposed framework to be ontology-oriented to enable knowledge sharing and reuse among knowledge-based systems and agents.

---

#### IV.5.2.2 Ontologies Life Cycle

It is a challenge to build a consensual ontology in a given domain. Moreover, the construction of an ontology requires the participation of domain experts who play a key role in the extraction of domain concepts and relations thanks to their knowledge. Then, these concepts and relations shall be described formally through the technologies of knowledge representation to make the data semantics interpretable by computers.

Several methods and techniques for ontology construction have been proposed (Uschold 1996). However, researchers have not reached yet a consensus on a unique and correct ontology construction methodology. On the other side, research works succeeded in determining the necessary steps which are related to the ontology characteristics, e.g., ontology application domain, information type contained in the ontology, and the ontology description language expressivity. Nevertheless, most ontology construction techniques converge on a generic methodology.

This methodology consists in the following stages:

- **Ontology domain identification:** this permits to determine the objectives and the reasons of the ontology construction.
- **Ontology construction:** it consists in two main steps. The first one is Ontology editing. At this level, the ontology designer defines the key concepts, their characteristics in the domain of interest, and the existing relations between these concepts. The second step is Ontology encoding which concerns the concepts and relations representation in a formal ontology description language to be interpretable by computers.
- **Evaluation:** this stage allows designers to verify the ontology consistency and completeness with respect to the defined objectives.
- **Documentation:** it is essential to document the various elements defined

in the ontology and to provide it to the ontology users.

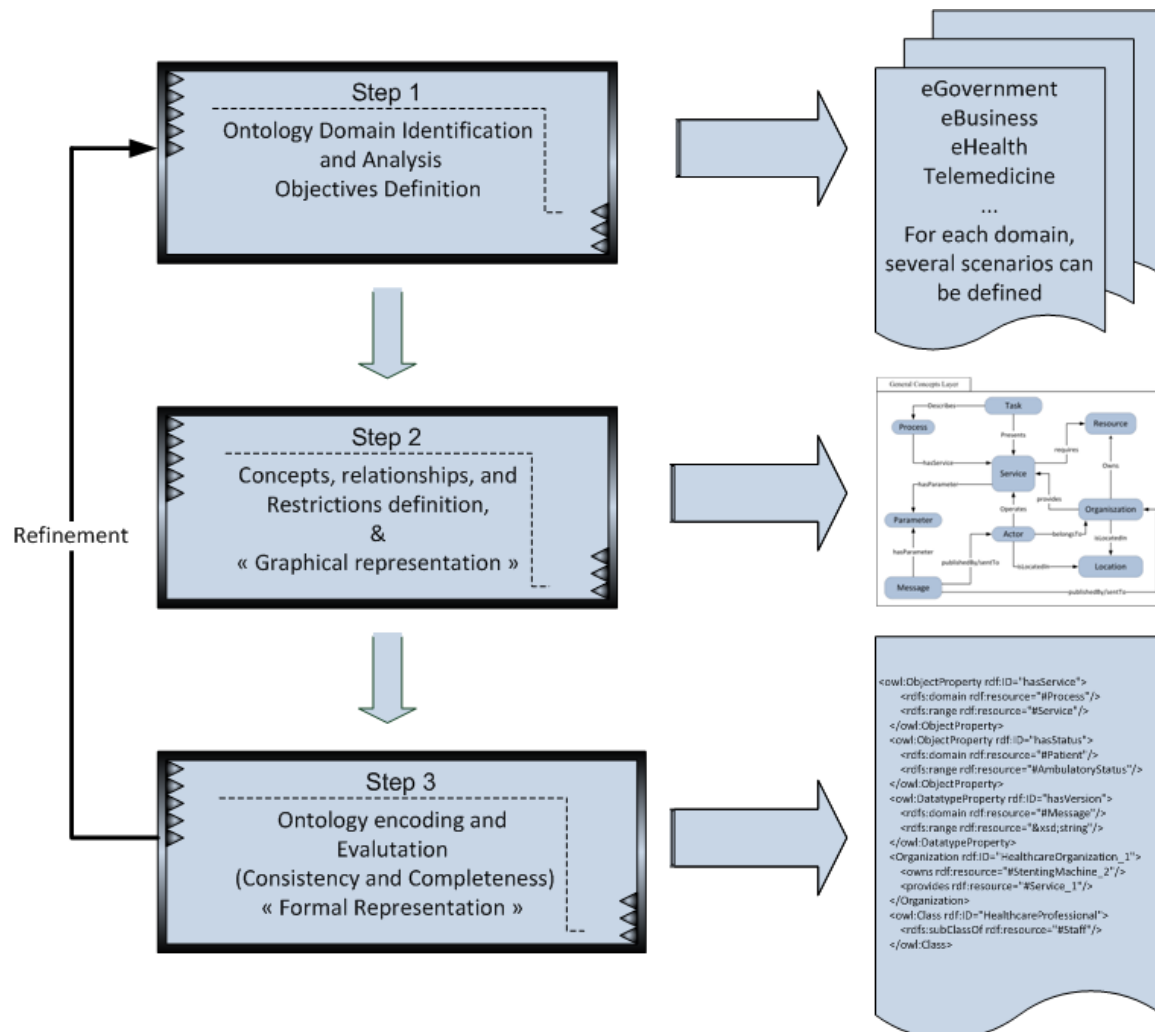


Figure 13. Ontology Construction Methodology.

In the same way, to create our ontology, we have adopted a simple methodology, based on the general methodology presented above. Figure 13 depicts the first two phases of our ontology construction methodology.

In the next sub-sections, we describe the ontological models construction and instantiation in the telemedicine domain, using the above presented methodology to construct the ontology models.



---

## IV.6 Ontological Models instantiation and Description in the Telemedicine Domain

---

### IV.6.1 Telemedicine Domain Analysis and Concepts Extraction

According to what we have explored in the bibliographical study, we noticed that the telemedicine services effectiveness is determined by many factors. The most important ones are the quality and availability of pertinent information where and when needed. However, as we have mentioned in the introduction, telemedicine application scenarios are various, some of them are well known, i.e. patient home tele-monitoring, tele-consultation, tele-radiology, tele-ECG, etc, and are compatible with pre-described services and automated knowledge, but other scenarios are contextual and more complex, for example, making a decision in order to transfer a person undergoing a healthcare problem from an accident site to the most appropriate hospital best corresponding to the person's clinical status and conditions. This type of scenario becomes more complex when it takes place in hostile environments, e.g. critical geographically and isolated areas such as high mountains resorts which lack telemedicine infrastructures and resources.

In order to master the specificity and complexity of such types of scenario, a knowledge-based telemedicine system is required. To achieve this objective, we propose to instantiate our generic knowledge meta-model presented above in order to obtain a global telemedicine domain ontology. Based on this ontology, knowledge-based telemedicine systems can then be built to grant every actor an optimal use of the available services and resources.

The telemedicine domain actors, i.e. *Healthcare Professional*, have different knowledge, aptitude, and skills. Mostly, they are geographically distributed in different locations and environments, e.g. practitioner office, medical centers, public and private hospitals, homes, accident sites, etc. They need to exchange medical or administrative data in order to provide the best healthcare to patients. Healthcare professionals also perform multiple *Tasks*, i.e. data search, data retrieval, messages exchange, etc, concerning a *Patient* medical assistance. On the other side, Telemedicine Tasks require various Resources, e.g. *Data/Knowledge*

*Sources, Services, Devices*, etc. Normally, these Resources are owned or managed by diverse *Organizations*, i.e. *Healthcare Institutions* such as *Hospitals, Clinics, Insurance Companies*, etc. The resources conditions, in terms of accessibility, availability, capabilities, and usability, are constantly changing. Given this telemedicine context and in order to provide the healthcare professionals with data and information that are the most relevant to their requests, we need to:

- Link the aforementioned entities, i.e. Healthcare Professionals, Tasks, Services, Organizations, Devices, etc, through ontological models representing all the previous telemedicine domain elements.
- Integrate all the ontological models in one ontology which we will call Global Telemedicine Domain Ontology.
- Create a rules base to perform a rule-based reasoning in order to infer the required resources related to each telemedicine task.

In the next section we instantiate the knowledge meta-model in the telemedicine domain. Then, we describe each entity of the telemedicine domain using an ontological model.

---

#### IV.6.2 Telemedicine Ontological Models Instantiation

The knowledge meta-model we have defined in section IV is a general one and it can be instantiated in several domains. In this section we personalize the knowledge meta-model according to the telemedicine domain concepts we have identified in the previous section.

Figure 14 displays a graphical representation of the telemedicine domain entities that we have instantiated from the knowledge meta-model entities. For example, the entity Telemedicine Task represents an instance of the Task model; Healthcare Professional represents an instance of the Actor Model and so on.

The global telemedicine domain ontology, displayed in Figure 14, contains sub-ontologies such as Healthcare Institution, Medical Equipment, Clinical Status, Material Actor, which is itself a generalization of Personal Health Monitor, etc. In the following, we describe in detail the main telemedicine domain ontological models.

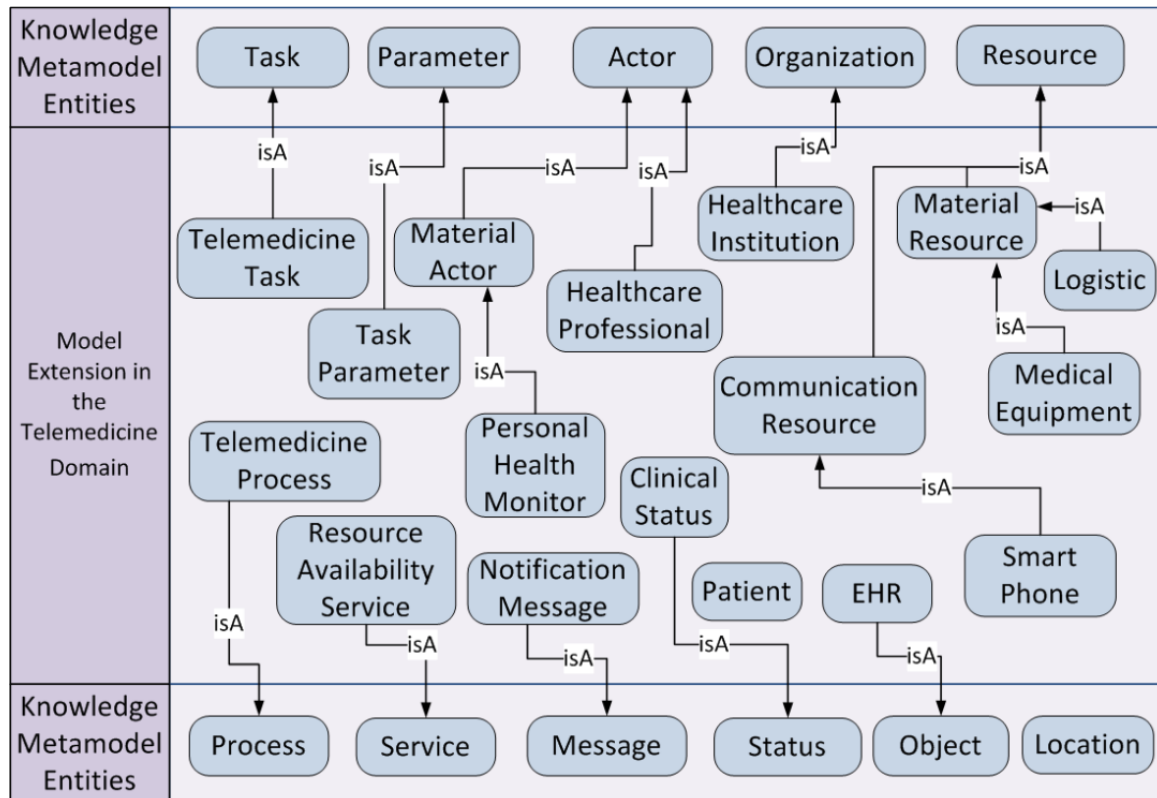


Figure 14. Generic View of the Global Telemedicine Ontology Graphical Representation.

### IV.6.3 Telemedicine Ontological Models Description

In section IV.5.1, we mentioned that the knowledge meta-model organizes the knowledge of the existing pervasive applications into three main levels, i.e. Actor, Task, and Resource. In the following sub-sections we describe in detail the three principal ontological models of Actor, Task, and Resource as well as the Object ontology.

#### IV.6.3.1 Actor Ontological Model

CEN-TC251 defined, through its norm EN 13940, the health actor as being every person, organisation, hardware or software involved in healthcare activities. The Actor ontology thus describes any entity that interacts with an information system. Generally, an Actor belongs to an organization and he can perform one or more tasks within a given process.

Figure 15 describes an actor ontology containing two main sub-classes: human actor and material actor. Generally, the human actors in the

telemedicine domain are healthcare professionals, i.e. Emergency Physician, Rescue Team Member, First Aid Person, Nurse, etc, whereas the material actors could be any information system or application integrated in a telemedicine device. The Healthcare Professional class has multiple identifying attributes, i.e. Id, first name, last name, etc, and domain related attributes, i.e. specialty, availability, organization, location, etc, and accessibility level of the performed task.

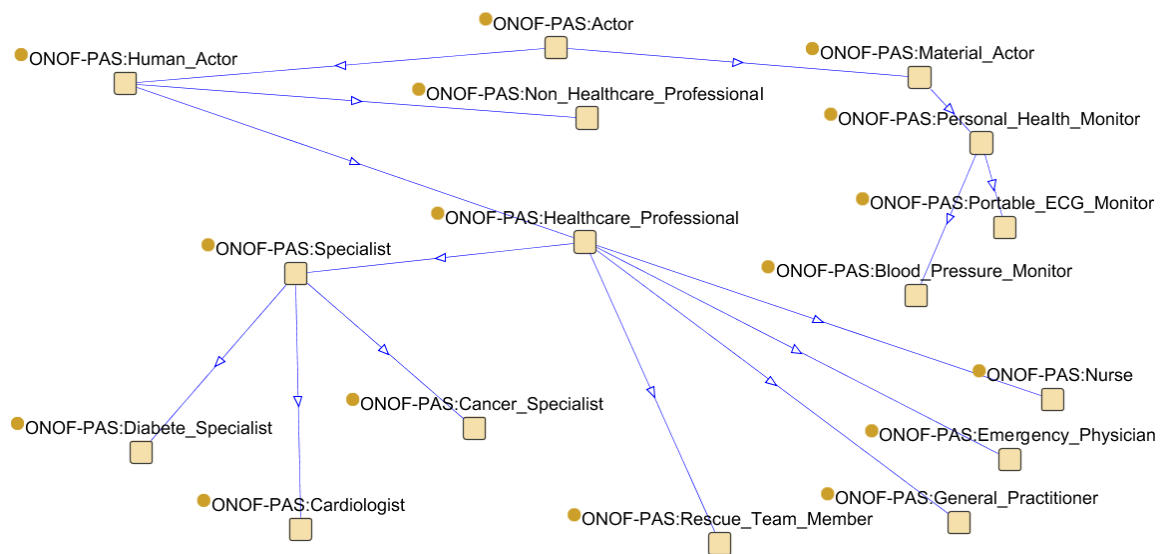


Figure 15. Actor Ontology Concepts in the Telemedicine Domain.

#### IV.6.3.2 Task Ontological Model

The task ontology describes the different functionalities of a given domain. Figure 16 shows a view of a telemedicine task ontology consisting of a collection of telemedical tasks, i.e. tele-consultation, tele-expertise, tele-ECG, etc, and non telemedical tasks, i.e. patient admission, patient transfer. This collection is not exhaustive and it can be extended to integrate different other tasks. All tasks are performed by a healthcare domain actor to provide a medical assistance to a patient. As we have seen in the previous section, the actors in the telemedicine domain can be healthcare professionals to whom we should associate a set of tasks according to their profiles. Thereby, the personalized work environment of each healthcare professional can be automatically generated.

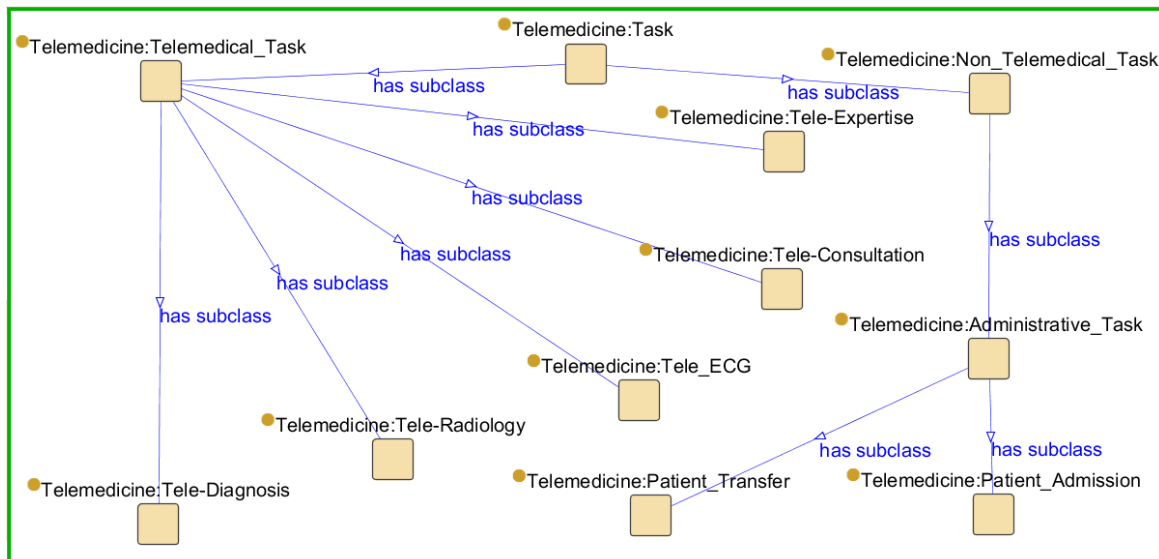


Figure 16. Example of Task Ontology Concepts in the Telemedicine Domain.

The Task class has the attribute *Accessibility Level*, which permits to manage the possibilities of the tasks performed by a healthcare professional. For example, a tele-expertise task has an accessibility level which is different from the tele-consultation one. That means, for instance, that a tele-expertise task is accessible for a specialist but not for a nurse. Additionally, each task has attributes like *id*, *description*, a *specialty domain*. As it has been previously mentioned, a task is performed by an actor and requires diverse resources, e.g. devices, data sources, services, etc. Furthermore, a task may invoke one or more services during its execution to retrieve some required data, as well as it generates one or more messages to be sent to one or more destinations, e.g. actor or organization. Task has also numerous parameters varying according to the task specialty, e.g. cardiology, radiology, biology, etc. For example, in the cardiology domain, these parameters can be heart rate, blood pressure, oxygen saturation level, Electrocardiogram (ECG), etc.

#### IV.6.3.3 Resources Ontological Model

Tasks require heterogeneous resources, e.g. devices, services, data sources, and human resources. Human resources have already been represented by the Actor ontology (Figure 15). Similarly, services may be described by a service ontology. Indeed, we have preferred to separate these two ontologies (Actor and Service) of Resources ontology, due to their direct

relationships with tasks. In the knowledge meta-model, we have defined the relationships between task and service as *Task invokes a Service*, and we have defined the relationship *Actor performs a Task*. Therefore, there is no need to include neither the service nor the actor ontology in the resources ontology. Figure 17 displays the resources ontology describing different types of resources that are used to perform a telemedicine task. We have organized the resources in two main classes.

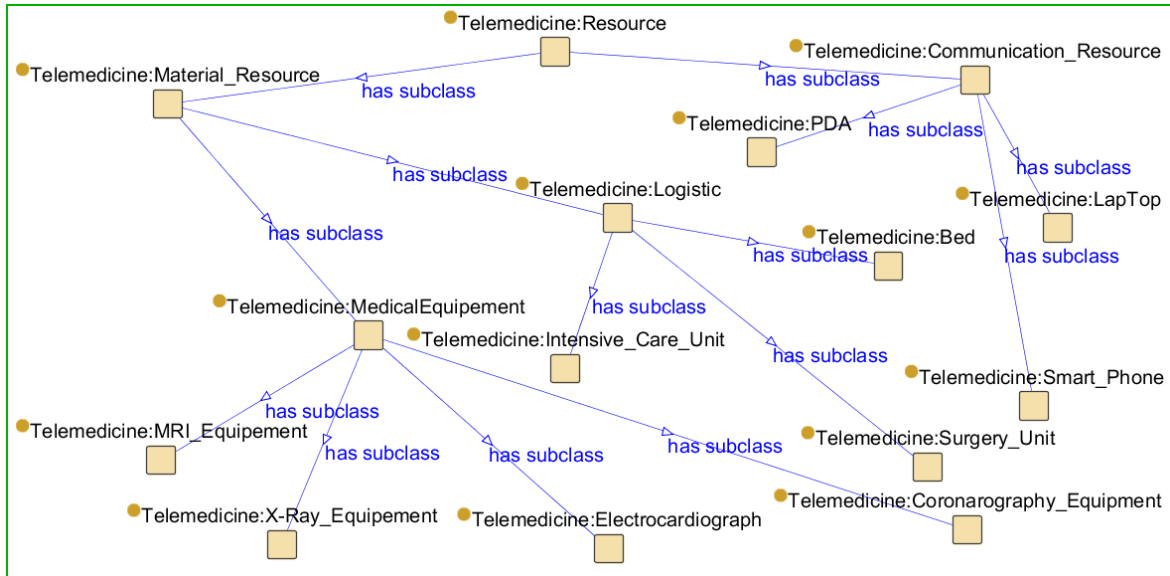


Figure 17. Example of Resources Ontology Concepts in the Telemedicine Domain.

The first one depicts the material resources including the logistic resources, such as intensive care unit or surgery unit, and medical equipments, such as electrocardiograph, Magnetic Resonance Imaging (MRI) equipment, X-ray equipment, etc. The second class is the communication resources one representing the communication tools such as Personal Digital Assistant (PDA), smartphone, laptop, etc, used by different actors to communicate and exchange data.

Principally, resources are owned or managed by organizations and have common attributes like identifier, description, while material resources have as attributes, availability and quantity. Concerning the communication resources, we have defined additional attributes such as screen size, memory size, embedded telecommunication technology type (e.g. GPRS, EDGE, UMTS, HSDPA, HSUPA, WiMax, LTE, etc).

#### IV.6.3.4 Data Objects Ontological Model

The Object ontological model represents the different types of data that are included in the electronic health record of the patient and that may be communicated to or interchanged between healthcare professionals for the patients' treatment. In telemedicine, a typical object ontology is composed of two main classes: medical data object, i.e. textual data, images, bio-signals, etc, and non medical data object including for example, bill, prescription, social security data, etc. Figure 18 shows the main concepts for each of the classes cited above.

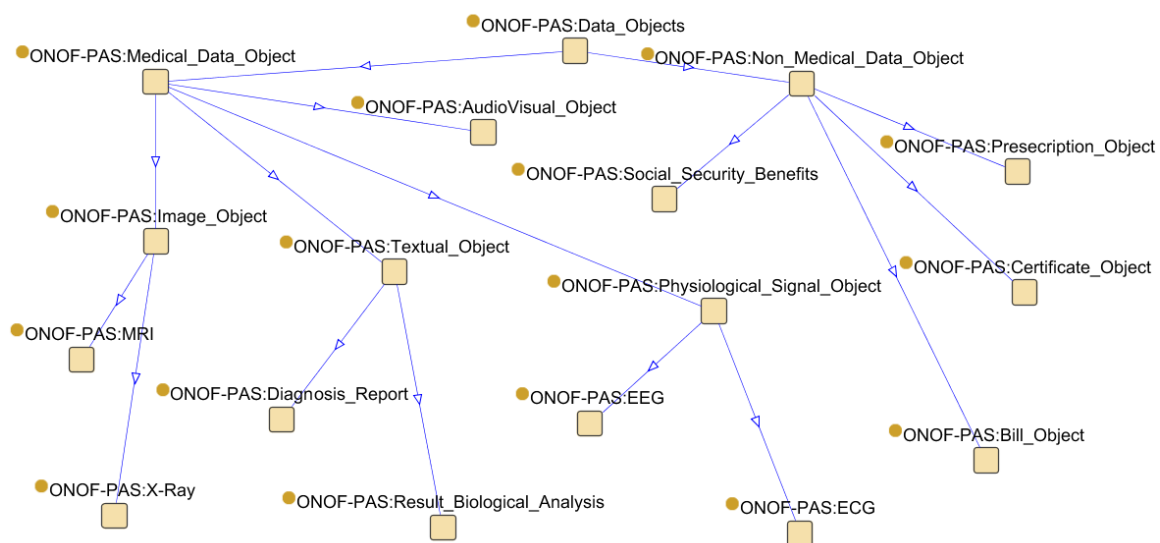


Figure 18. Concepts of a Typical Data Objects Ontology in the Telemedicine Domain.

#### IV.6.3.5 Example of Task Model Instantiation in Telemedicine

We provide hereafter an example of a task model instantiation in the telemedicine domain, including entities, relationships and attributes. This model describes the patient orientation or transfer scenario which we have adopted in this thesis.

The task “patient orientation/transfer” is a telemedicine task which concerns a patient, “Mr X”, who has as clinical condition a “heart attack” (Figure 19). According to the task accessibility level attribute, this task is accessible and can be performed by a rescue team member or an emergency physician who has a communication device like a Smartphone

“communication resource”. Furthermore, the patient orientation/transfer task requires different logistic and medical equipments, i.e. a “material resource” such as a place in an intensive care unit, a surgery unit, a coronarography device, etc. These resources are managed by hospitals, i.e. “organizations” located at different places. The Location class provides address information, i.e. building number, street name, city, country, and zip-code, about the physical location of each actor and organization. The Organization class represents any institution in the application domain. In telemedicine, healthcare organizations can be Hospitals, Medical Centers,

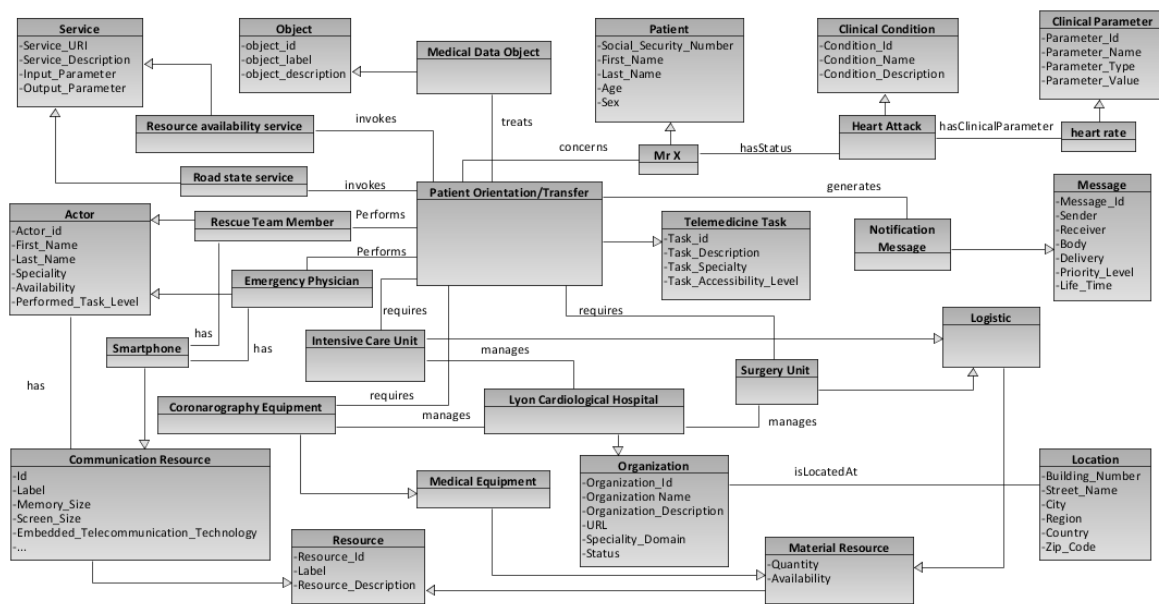


Figure 19. Example of a Telemedicine Task Model Instantiation.

Insurance Companies, etc. Furthermore, an Organization provides a set of services and owns material, communication, and human resources. Depending on the task specialty, several services can be invoked by the task. A service has input/output parameters, and other attributes such as service description, URI (Uniform Resource Identifier), etc. Services can be invoked and executed to update resources data and to retrieve data hosted by other organizations, for example, services to access the patient’s EHR or social security data.

Tasks generate multiple messages to be exchanged between the actors. The Messages structure consists of the sender, receiver, and message body. Actors and Organizations can publish and receive messages



from each other. For instance, a notification message can be generated by the patient orientation task and sent to an emergency center to inform the staff that a patient is being transported to the hospital where he/she will be hospitalized. The Parameter class represents the data included in a message. It has three main attributes: parameter name, type, and value. For example, in a patient transfer task, the generated message may include as parameters date, time, and location of the accident.

---

#### IV.6.4 Formal Representation of Ontological Models

In the previous sections, we have described the graphical representation of the ontological models which allowed us to display their basic concepts and relationships. However, we need to have the data represented by ontologies to be interpretable by computers and usable in reasoning processes. In this section, we will elaborate a formal representation of the ontological models by using a standard ontology description language like OWL-DL. Ontologies will also be semantically enriched by restrictions that we will define in order to allow a Reasonner to infer the hierarchy of classes.

---

##### IV.6.4.1 Ontology Description Language Selection

To develop the formal representation of our ontological models, we firstly must choose an appropriate language. Currently, there are different types of formalisms and languages which allow formal knowledge representation in an ontology. A study of these languages has been conducted in section II.5.3 of the second chapter of this manuscript. Among these languages, we have chosen OWL-DL.

As we have seen in chapter II, OWL includes three increasingly-expressive sublanguages which are OWL-Lite, OWL-DL (Description Logics), and OWL-Full. To formalize our ontological models, we used OWL-DL which is much more expressive than OWL-Lite and includes additional DL that are flexible enough to automate reasoning. In OWL-DL, there are two main types of properties that can be defined. The first type is Object properties, e.g., Actor *performs* Task. The second type is Data type properties, e.g., Actor *has Specialty*. The DL extension provides OWL with a rich set of primitives, i.e., intersection  $\wedge$ , union  $\vee$ , complement  $\neg$ , etc. It allows the ontology designer to define restrictions and conditions on the ontology classes. It is therefore possible to automatically determine the

classification hierarchy and check for inconsistencies in an ontology that conforms to OWL-DL.

#### IV.6.4.2 Ontological Models Encoding

After having chosen OWL-DL to formally represent the ontological models, we present hereafter some coding examples of the main models in a file with the “.owl” extension. Fragment 1 shows an example of a class and sub-class declaration in OWL. In this example, the class *Actor* is disjoint with the other classes which are *Organization*, *Resource*, *Location*, *Service*, etc. Additionally, *Human actor* and *Material actor* have been declared as sub-classes of class *Actor*. Fragment 2 provides examples of the definition of different object type properties in OWL, such as the property “*performs*” which has as domain the class *Actor* and as range the class *Task*. “*isPerformedby*” is the inverse of the “*perform*” property. The same yields for the object properties of “*requires*”, “*concerns*”, “*has clinical conditions*”, etc.

Fragments 3, 4 and 5 give examples of different data type properties definitions, such as “*message priority level*”, “*task accessibility level*”, and “*embedded telecommunication technology*”, including the class domain and the data range.

```
<owl:Class rdf:ID="Actor">
  <rdfs:subClassOf rdf:resource="#Telemedicine_Concept"/>
  <owl:disjointWith rdf:resource="#Clinical_Condition"/>
  <owl:disjointWith rdf:resource="#Location"/>
  <owl:disjointWith rdf:resource="#Message"/>
  <owl:disjointWith rdf:resource="#Organization"/>
  <owl:disjointWith rdf:resource="#Parameter"/>
  <owl:disjointWith rdf:resource="#Patient"/>
  <owl:disjointWith rdf:resource="#Process"/>
  <owl:disjointWith rdf:resource="#Resource"/>
  <owl:disjointWith rdf:resource="#Service"/>
  <owl:disjointWith rdf:resource="#Task"/>
</owl:Class>
<owl:Class rdf:ID="Human_Actor">
  <rdfs:subClassOf rdf:resource="#Actor"/>
</owl:Class>
<owl:Class rdf:ID="Healthcare_Professional">
  <rdfs:subClassOf rdf:resource="#Human_Actor"/>
</owl:Class>
```

Fragment 1. Actor class declaration. “Extract of our ONOF-PAS Ontology Source Code”.

```

<owl:ObjectProperty rdf:ID="performs">
  <rdfs:domain rdf:resource="#Actor"/>
  <owl:inverseOf rdf:resource="#isPerformedBy"/>
  <rdfs:range rdf:resource="#Task"/>
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="Concerns">
  <rdfs:type rdf:resource="#owl:ObjectProperty"/>
  <rdfs:domain rdf:resource="#Task"/>
  <rdfs:range rdf:resource="#Patient"/>
</owl:FunctionalProperty>
<owl:ObjectProperty rdf:ID="requires">
  <rdfs:domain rdf:resource="#Task"/>
  <rdfs:range rdf:resource="#Material_Resource"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasClinicalCondition">
  <rdfs:domain rdf:resource="#Patient"/>
  <rdfs:range rdf:resource="#Clinical_Condition"/>
</owl:ObjectProperty>

```

Fragment 2. Object type property declaration examples “Extract of our ONOF-PAS Ontology Source Code”.

```

<owl:DatatypeProperty rdf:ID="accessibilityLevel">
  <rdfs:domain rdf:resource="#Task"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="messageDelivered">
  <rdfs:domain rdf:resource="#Message"/>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf>
        <rdf:List>
          <rdf:first rdf:datatype="&xsd:boolean">false</rdf:first>
          <rdf:rest>
            <rdf:List>
              <rdf:first rdf:datatype="&xsd:boolean">true</rdf:first>
              <rdf:rest rdf:resource="&rdf:nil"/>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>

```

Fragment 3. Data type property declaration example. “Extract of our ONOF-PAS Ontology Source Code”.

```

<owl:DatatypeProperty rdf:ID="messagePriorityLevel">
  <rdfs:domain rdf:resource="#Message"/>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf>
        <rdf:List>
          <rdf:first rdf:datatype="&xsd:string">Low</rdf:first>
          <rdf:rest>
            <rdf:List>
              <rdf:first rdf:datatype="&xsd:string">Medium</rdf:first>
              <rdf:rest>
                <rdf:List>
                  <rdf:first rdf:datatype="&xsd:string">High</rdf:first>
                  <rdf:rest rdf:resource="&rdf:nil"/>
                </rdf:List>
              </rdf:rest>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>

```

Fragment 4. Data type property “message priority level”. “Extract of our ONOF-PAS Ontology Source Code”.

```

<owl:DatatypeProperty rdf:ID="embeddedTelecomTech">
  <rdfs:domain rdf:resource="#Communication_Resource"/>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf>
        <rdf:List>
          <rdf:first rdf:datatype="&xsd:string">GPRS</rdf:first>
          <rdf:rest>
            <rdf:List>
              <rdf:first rdf:datatype="&xsd:string">EDGE</rdf:first>
              <rdf:rest>
                <rdf:List>
                  <rdf:first rdf:datatype="&xsd:string">HSDPA</rdf:first>
                  <rdf:rest>
                    <rdf:List>
                      <rdf:first rdf:datatype="&xsd:string">HSUPA</rdf:first>
                      <rdf:rest rdf:resource="&rdf:nil"/>
                    </rdf:List>
                  </rdf:rest>
                </rdf:List>
              </rdf:rest>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>

```

Fragment 5. Data type property “embedded telecommunication technology”. “Extract of our ONOF-PAS Ontology Source Code”.

#### IV.6.4.3 Restrictions Definition

OWL-DL provides a rich set of constructors for the definition of conditions on the concepts, relationships, and instances. On the other hand, properties are used to create restrictions which may be used to restrict the individuals that belong to the classes. We will describe hereafter some restrictions (conditions) that we have introduced in our ontologies.

OWL allows three categories of restrictions: Quantifier, Cardinality, Has value. The restrictions describe the constraints on the relationships that the individuals participate in for a given property. In OWL a quantifier restriction is composed of a quantifier, a property and a filler. The two quantifiers that may be used are: the Existential Quantifier “ $\exists$ ”, which can be read as “at least one” or “some”, and the Universal Quantifier “ $\forall$ ”, which can be read as “only”. For a set of individuals, an existential restriction specifies the existence of a relationship (i.e. at least one) along a given property to an individual that is a member of a specific class. In Table 3, we have listed some classes of the Global Telemedicine Domain Ontology and the restrictions which we have defined on these classes.

Table 3. Restrictions on the Classes of the Global Telemedicine Ontology Domain.

| Class                   | Restriction  |
|-------------------------|--|
| Actor                   | $\exists$ performs Some Task                             |
| Healthcare Professional | $\forall$ belongsTo only Organization                    |
| Patient                 | $\exists$ hasClinicalConditions some Clinical Conditions |
| Patient-Admission       | $\exists$ performedBy some Rescue Team Member            |
| Tele-Expertise-Cardio   | $\exists$ performedBy some Cardiologist                  |
| Tele-Consultation       | $\exists$ performedBy some General Practitioner          |

In the first restriction, every actor should perform at least one task. More formally, for an individual to be a member of the class “*Actor*”, he/she should perform at least one “*Task*”. In other words, for somebody or something to be an actor, it is necessary for him/her or it to be in a relationship with an individual that is a member of the class “*Task*” via the property “*performs*”. Thus, the class “*Healthcare Professional*” inherits

this restriction from the class “*Actor*”. Additionally, for somebody to be a “*Healthcare Professional*”, it is necessary for him/her to be in a relationship with an individual that is a member of the class “*Organization*” via the property “*belongsTo*”. In the same way, we defined restrictions for the class “*Task*”, e.g., for a task to be a “*Tele-Expertise*” in cardiology, it is necessary that the healthcare professional who performs this task is a “*Cardiologist*” as described in the restriction on the “*Tele-Expertise-Cardio*” class.

---

## IV.7 Rules Description

OWL has some limitations which concern the definition of composite properties. Efforts have been made by the W3 community to increase the expressiveness of OWL, particularly, to develop a language for the description of rules in SWRL (Semantic Web Rule Language). This language is based on a combination of the OWL-DL and OWL-Lite sublanguages of OWL, see section II.5.3.2.

We have used the SWRL language to define the rules to be applied by an inference engine. In the following, we will show an example, how we can develop complex relationships between concepts in our global telemedicine domain ontology by creating SWRL rules. As illustrated in Figure 20, the property “*concerns*” defines a link between a “*Telemedicine Task*” (e.g. *Patient Transfer*) and a “*Patient*”. On the other hand, the property “*hasClinicalConditions*” links a “*Patient*” with a “*ClinicalCondition*”, the property “*associatedWith*” links the patient’s “*ClinicalCondition*” with “*Resources*”. Thus, a rule must be created in order to deduce a relationship between the *Telemedicine Task* and the *Resources*. In general, this rule shall indicate that: if a telemedicine task concerns a patient, if this patient has clinical conditions, and if these clinical conditions are associated with some resources, then the telemedicine task requires these resources.

The rules we have created can be organized in two main groups as depicted in Figure 21. The first group of rules encompasses the management rules which link the actors to the tasks that they perform according to their profiles. Additional management rules must however be defined to link each task to the required heterogeneous resources,

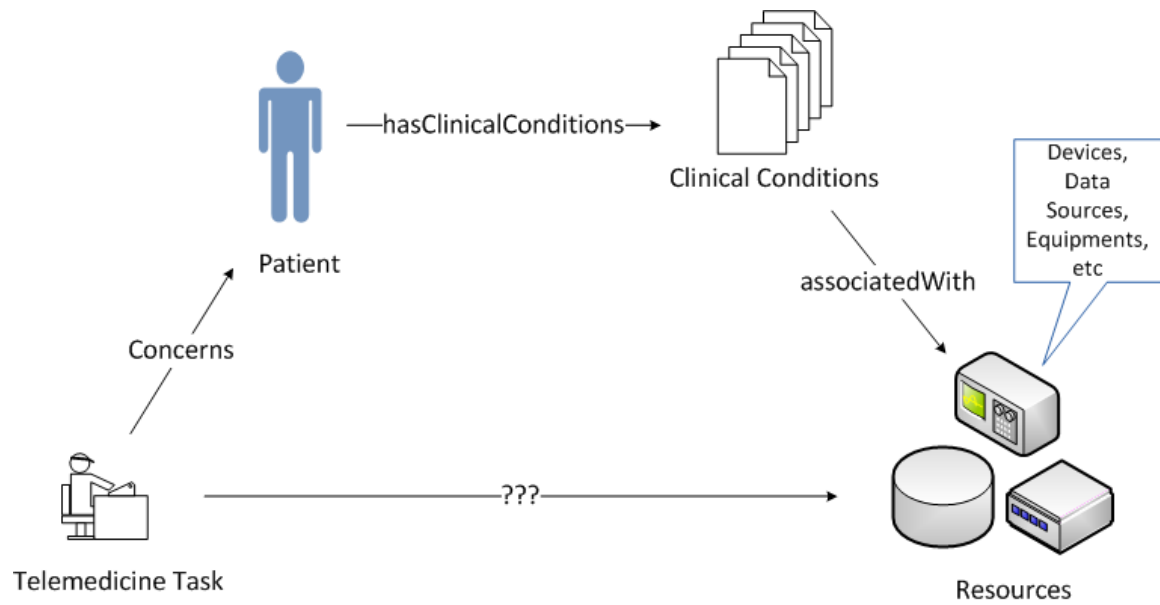


Figure 20. Example of a Complex Relationship Between Different Concepts from the Global Telemedicine Domain Ontology.

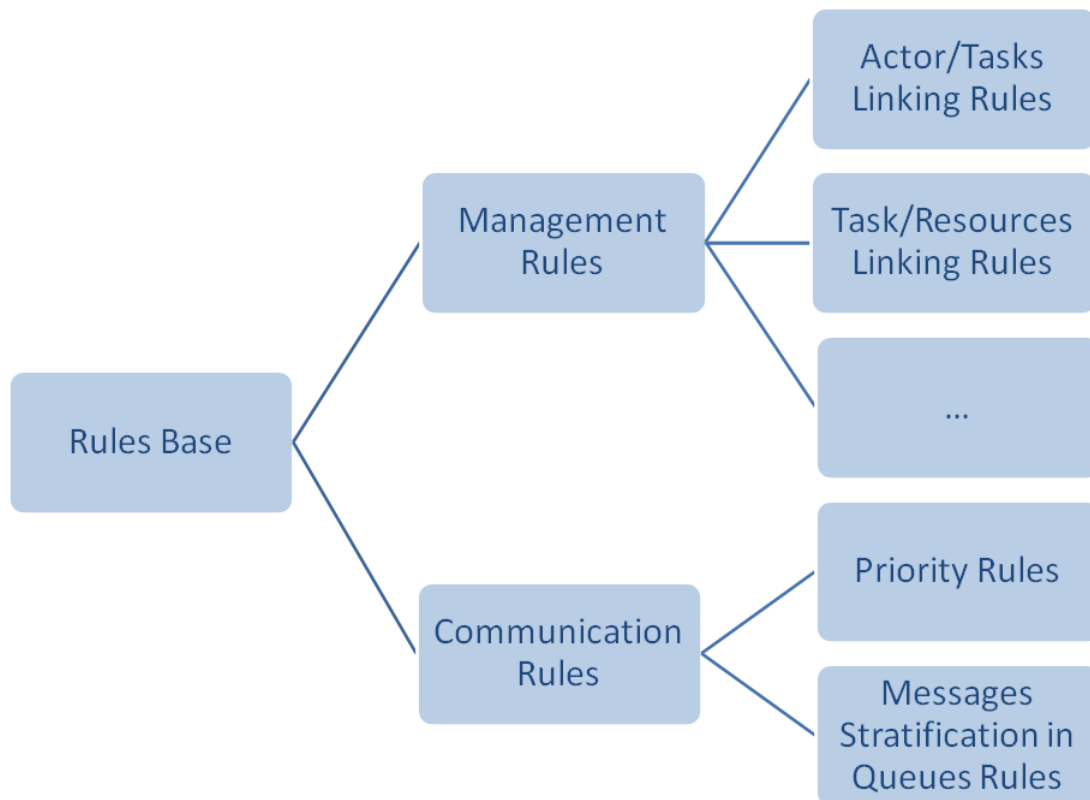


Figure 21. ONOF-PAS Rules Classification.

depending on the conditions of the objects concerned by the task, as we have seen in the previous example.

The second main group is the communication rules group. It includes the rules created in order to prioritize the messages generated by the tasks, according to the actors profiles. This rules group also contains stratification rules which we have defined to stratify messages in queues corresponding to the priority levels, i.e. Low, Normal, and High.

The rules are formulated as a form of an implication between an antecedent (body) and a consequent (head). Fragment 6 describes a formal representation of SWRL rules. The intended meaning can be read as: whenever the condition specified in the antecedent is hold, then the conditions specified in the consequent must be hold. Both the antecedent (body) and the consequent (head) consist of one or more atoms. SWRL atoms are defined as indicated in Table 4.

Table 4. SWRL Atoms Description.

| Variable           | Description  |
|--------------------|--|
| <b>C</b>           | <b>Class</b>   |
| <b>P</b>           | <b>Object Property</b>                                   |
| <b>U</b>           | <b>Data type property</b>                                |
| <b>D</b>           | <b>Data type</b>   |
| <b>i,j</b>         | <b>Object variable names or object individual name</b>   |
| <b>V1, ..., Vn</b> | <b>Data type variable names or data type value names</b> |
| <b>b</b>           | <b>Built-in names</b>                                    |

$C(i) P(i,j), U(i,v), D(v), \text{builtIn}(b, v1, v2, \dots, v3) \rightarrow \text{Atoms.}$

Fragment 6. SWRL Formal Representation.

The following management rule example, fragment 7, describes the context for a telemedicine task that concerns a patient who has as symptoms chest pain. The rule consequent indicates that the telemedicine task requires material resources such as a bed in an intensive care unit, a place in a surgery unit and a coronarography equipment to perform a specific treatment to the patient. Fragment 8 shows other examples of SWRL rules which we have defined to explicitly describe complex relationships



between the concepts of the global telemedicine domain ontology. For example, rule number 1 links two facts to infer a third one, i.e., if a healthcare professional is located at the same location as an organization is, it means that the healthcare professional belongs to this organization. Rule number 2 determines the location of a healthcare professional depending on the fact that he/she belongs to an organization located at a given location, whereas rule number 3 assigns a cardiologist as a human resource required to perform a tele-expertise task concerning a patient who has a cardiac arrhythmia as clinical status.

```

ONOF-PAS:Task(?Telemedicine:tas) ∧
ONOF-PAS:Patient(?Telemedicine:p) ∧
ONOF-PAS:Chest_Pain(?Telemedicine:cp) ∧
ONOF-PAS:Concerns(?Telemedicine:tas, ?Telemedicine:p) ∧
ONOF-PAS:hasClinicalCondition(?Telemedicine:p, ?Telemedicine:cp) ∧
ONOF-PAS:Intensive_Care_Unit(?Telemedicine:icu) ∧
ONOF-PAS:Surgery_Unit(?Telemedicine:su) ∧
ONOF-PAS:Coronagraphy_Equipment(?Telemedicine:coro)
→
ONOF-PAS:requires(?Telemedicine:tas, ?Telemedicine:icu) ∧
ONOF-PAS:requires(?Telemedicine:tas, ?Telemedicine:su) ∧
ONOF-PAS:requires(?Telemedicine:tas, ?Telemedicine:coro)

```

Fragment 7. Example of a management rule that assigns different resources to a telemedicine task.

```

Rule 1:
ONOF-PAS:HealthcareProfessional (?hcp) ∧ ONOF-PAS:Organization (?org) ∧ ONOF-
PAS:Location (?loc) ∧ ONOF-PAS:locatedAt (?org,?loc) ∧ ONOF-PAS:locatedAt (?hcp, ?loc)
→
ONOF-PAS:belongsTo (?hcp, ?org)

Rule 2:
ONOF-PAS:Healthcare Professional (?hcp) ∧ ONOF-PAS: Organization (?org) ∧ ONOF-
PAS:Location (?loc) ∧ ONOF-PAS:isLocatedAt(?org,?loc) ∧ ONOF-PAS:belogsTo(?hcp,?org)
→
ONOF-PAS:isLocatedAt(?hcp,?loc)

Rule 3:
ONOF-PAS:Tele-expertise Task (?task) ∧ ONOF-PAS:Patient(?patient) ∧
ONOF-PAS:concerns(?task, ?patient) ∧
ONOF-PAS:Cardiac_Arrhythmia ( ?condition) ∧
ONOF-PAS:hasClinicalStatus(?patient,?condition) ∧
Cardiologist (?cardiologist)
→
ONOF-PAS:requires(?task,?cardiologist)

```

Fragment 8. SWRL Rules Defined in the Telemedicine Task Ontology.

SWRL rules are saved as OWL individuals with their associated OWL file. To effectuate a rule-based reasoning using SWRL rules, we need to integrate an inference engine as we will see in chapter VI.

## **IV.8 Conclusion**

In this chapter, we have presented our methodology to construct the ONOF-PAS framework we have proposed in this thesis. We have described the different phases of our methodology including the knowledge meta-model definition and the ontological model creation. We have instantiated the different ontologies in the telemedicine domain. We then formalized the global telemedicine domain ontology using OWL-DL. Additionally, we have created a rules base related to the telemedicine domain to enrich the semantics of our global telemedicine domain ontology. We have also formalized some rules using SWRL.

In some particular application domains such as the telemedicine domain, the construction of an ontology is a long and iterative process which requires several phases of validation and refinement. In this chapter, we have instantiated our generic meta-model in the telemedicine domain. However, other application scenarios would require to add new concepts, change or delete the pre-defined ones. This iterative process shall stop when ontology is estimated to be satisfactory.

In this chapter, we have achieved the first three phases of our ONOF-PAS design methodology we have described in section IV.4. The objective is to create a telemedicine knowledge base to capture and organize knowledge about the specifications of telemedicine tasks in pervasive environments.

In the next chapter we will realize the fourth phase of our methodology. We will present the ONOF-PAS architecture and the different object oriented components for tasks management and resources computing.



# *Chapter V*

## *Task-centred ONOF-PAS Architecture*



# Chapter V      Task-centred ONOF-PAS Architecture

## Overview

In this chapter, we present the ONOF-PAS architecture, based on the knowledge base layer we have created and formalized in the previous chapter. We model the different object-oriented components used for task management and resources processing. Additionally, we define a message management model as well as a policy for messages transmission. We also address the issues related to resources ontology mapping and resources data updating using web services.

---

### V.1      Introduction

Our goal in this chapter is to build the Task-centred ONOF-PAS architecture to provide a solution enabling, from one side, an optimized operating of the available resources and services in pervasive environments, and from the other side, the support of decision making processes in non pre-defined scenarios that occur in hostile environments. The proposed architecture should support the management and execution of various tasks in the telemedicine domain.

In the previous chapter, we have achieved the first three phases of the design methodology we have proposed in section IV.4. This chapter concerns the realization of phase number 4 (Figure 11, section IV.4). Thus, in this chapter, we will first describe the user/system interactions based on the patient transfer scenario that we have adopted to investigate the usability of the proposed framework. Then, we will define the design requirements of the ONOF-PAS architecture. After that, we will present a global view of the ONOF-PAS architecture and provide a detailed description of its components.

---

### V.2      Actor/System Interactions in the Patient Transfer Scenario

Actor/System interactions in pervasive environments can be described by the messages that will be exchanged between the users and the system. A typical actor/system interaction scenario could be as follows: an actor sends

a message to the system, then, the system performs an action or changes its state, and finally a message will be sent by the system to the user.

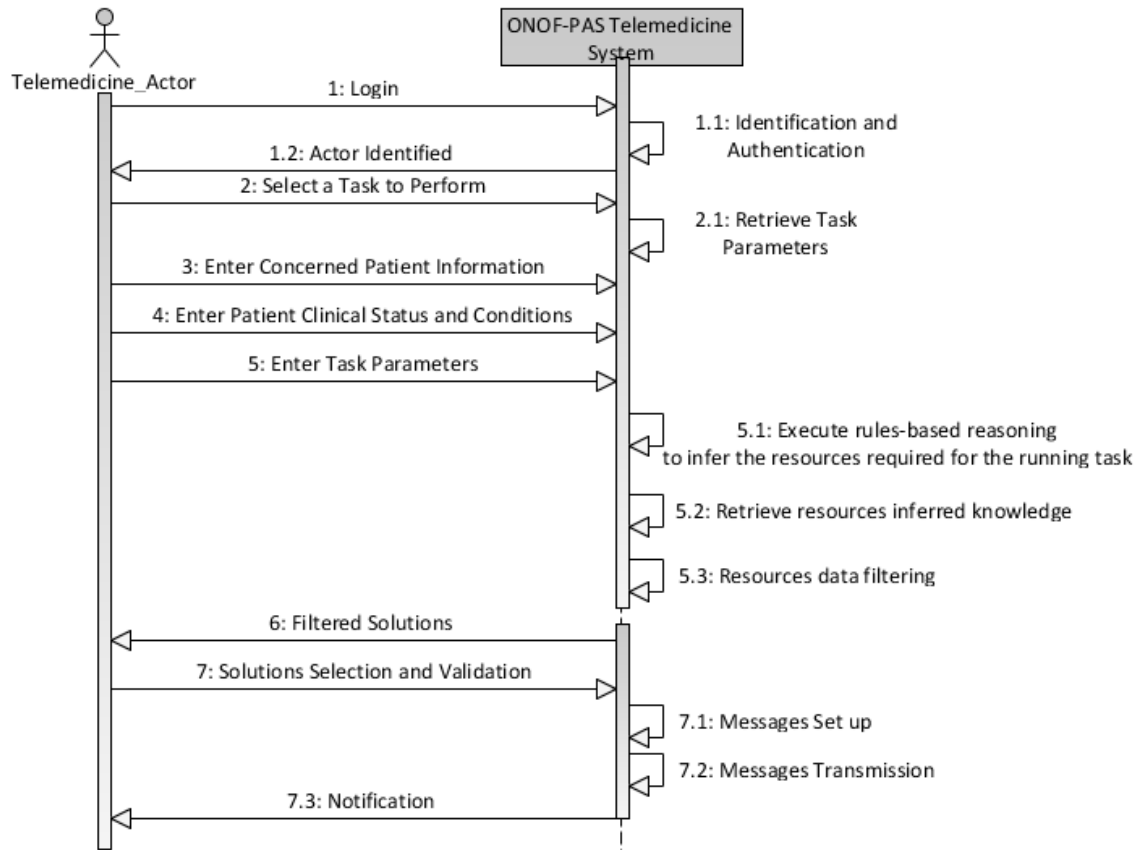


Figure 22. Sequence Diagram for a Typical Patient Transfer Scenario

We retake the scenario example concerning a patient orientation making decision, presented in section I.2.2 of the first chapter. The sequence diagram depicted in Figure 22 describes the system behavior from an external view point. We have represented the system as a black box. As we can see in Figure 22, after the identification and authentication process, a rescue team member or a first aid person selects a task to perform, for example “Patient Transfer Task”. The system links the selected task with the input parameters concerning the object treated by the task. In the patient transfer scenario, the study object is the patient and the input parameters can be, for instance: patient personal data, e.g. first name, last name, social security number, etc; the patient’s clinical conditions, e.g. an heart attack, high blood pressure, etc; and other clinical parameters

related to the patient's clinical status. For example in case of a heart attack, these parameters can be blood pressure, oxygen saturation level, heart rate, etc.

Once these parameters have been entered by the actor, the system executes a rule-based reasoning on the facts related to the actor who performs the task, to the object treated by the task and to its conditions. The reasoning result is a set of resources required to perform the task. These resources could be, for example, a place in an intensive care unit, a surgery unit, some specific equipment for the patient's treatment, or the staff that are able to perform a specific treatment, etc. The inferred resources are filtered according to their validity, in terms of availability and capabilities, and other user preferences, such as location, organization, etc. Based on the filtered resources, the system generates a list of recipients' addresses, i.e. hospitals or medical centers that possess the required resources and are ready to receive the patient. Once the actor has validated one of the proposed solutions, the messages can be generated and sent to the concerned destinations.

---

### **V.3 ONOF-PAS Architecture Design Requirements**

In this thesis, one of our main objectives is to design and to set-up a knowledge-based system architecture enabling an automated and contextual-dependant telemedicine task execution. The design of the ONOF-PAS architecture should meet a set of requirements that seem us to be extremely important to be taken into consideration throughout the system architecture design process. These requirements are:

- The possibility to instantiate the reference architecture to support different scenarios in the telemedicine and in the eHealth fields.
- The architecture should support semantic interoperability throughout the different ontological models.
- The architecture should be as compatible as possible with multiple standard computing tools, i.e. Object Oriented programming languages, Web programming standards, etc.
- The architecture should support the exchange of messages among actors and organizations. The transmitted messages must be based on XML



and able to encapsulate different types of data (i.e. textual, images, audio-video, signals, etc).

Additionally, we have determined four principal and more general requirements to design our architecture. These requirements are: genericity, interoperability, scalability, and reusability. We briefly comment, in the following subsections, the impact of these general requirements on the telemedicine and eHealth fields.

---

### **V.3.1 Genericity**

In the telemedicine and eHealth domains, different organizations (such as hospitals, medical centers, insurance companies, etc) are involved in healthcare processes and also in business management processes. On the other hand, the application scenarios are diverse and specific. For these reasons, we intentionally designed our ontological models to be as generic as possible so that they can be instantiated to deal with different application scenarios types, as we have seen in the previous chapter. For example, it shall be possible for any organization to instantiate the ontological models we have designed to adapt them to their specific profile.

---

### **V.3.2 Interoperability**

We meet the interoperability challenge at two main levels: the data exchange standards level and the semantic level. In the eHealth field, messages are exchanged in the EDI (Electronic Data Interchange) or XML format. Although the different standardization efforts (HL7, EHRcom and OpenEHR) have progressed to solve the interoperability problem, they did not succeed to convince all organizations to conform to a single standard. Moreover, interoperability problems are encountered at the level of the standard itself, for example, HL7 version 2 and HL7 version 3 are not interoperable. Thus, addressing the interoperability problem at the semantic level is essential. Furthermore, the information shared by systems must be understood at the domain concepts level so that the information can also be reprocessed by the receiving systems. Given the reasons mentioned above, the system functionalities we target to design must be based on standard protocols and on various message-oriented middleware tools such as HTML, XML, SOAP, etc, which are the most popular standards for data exchange.

---

### **V.3.3 Scalability**

To meet the fundamental problems encountered in the real healthcare domain world, every architecture solution must be scalable to a large number of users, applications, networked devices and huge amounts of data. The scalability issue will be even more challenging in the future to face the steadily increasing number of users, devices, applications and the increase in geographic areas coverage by internet. Thus, it is valuable to design a generic architecture model that will be scalable to multiple scenarios and applications. In fact, Cloud Computing should meet this requirement. However, the security question is still a risk to be considered by organizations.

---

### **V.3.4 Reusability**

In our opinion, it is less complicated to reuse a model than a source code. Thus, model reusability is a key feature for software engineers and project managers, and it is a strategic choice for software companies to develop models that can be reused in future, new projects. Thus, adopting MDA (Model Driven Architecture) and ODM (Ontology Definition Meta-Model), which are OMG standards, makes the architecture model flexible and open. Indeed, one of the most important advantages provided by MDA and ODM resides in the reusability of the designed models, and any MDA can be transformed to be reused in different platforms (e.g. J2EE, .Net ...).

---

## **V.4 Task-centred ONOF-PAS Architecture**

---

### **V.4.1 Why Task Centred ?**

The task ontological model is the cornerstone of our knowledge meta-model. As we have earlier stated, business processes are composed of a set of tasks which may invoke one or more services during their execution. In the existing pervasive systems, services are rigid and do not adapt to the context of use, and their output usually only partially meet the user needs. These drawbacks hinder data sharing and exchange among actors. To overcome these limitations and to effectively meet the user needs in terms of providing the relevant data to the user's request, we have focused on

tasks rather than on services, because the tasks can invoke the required services. For this reason, we propose a task-centered architecture model. In other words, according to the task performed by the user, the system will infer the required resources and invoke the required services. As we have seen in the previous chapter, we have defined several relations to associate the tasks to the other domain concepts, i.e. *task is performed by an actor*, *task concerns an object*, *task invokes a service*, etc (Figure 12, sectionIV.5).

---

#### V.4.2 ONOF-PAS Architecture Overview

The diverse operations, executed by the different components of the ONOF-PAS architecture, can be classified into three main types. The first type of these operations is related to the knowledge base, i.e. ontology consistency checking, rule-based reasoning, knowledge base querying, etc. The second type of operations is associated with tasks management and resources processing, i.e. task parameters linking, call of the inference engine, resources mapping, resources updating, resources filtering, etc. The third type of operations (i.e. user interface functionalities, web services discovery, etc.) concerns the interaction between the ONOF-PAS based system and an external actor (i.e. a user or any other information system). Based on this classification, we decided to organize the ONOF-PAS architecture into three main layers, i.e. Knowledge base, management layer, and interface and services layer (Figure 23).

In the previous chapter, we have presented in detail the design and realization of the core element of the ONOF-PAS architecture, the knowledge base layer, which includes the knowledge meta-model, the ontological models and the rules base.

The objective of the ontological models is to link the task performed by the user, the object treated by the task and its conditions, with the heterogeneous and available resources required to perform the task. The ontological models thus represent the key entities of pervasive environments. Each entity, i.e. *Actor*, *Task*, *Object*, *Service*, *Resource*, *Organization*, etc (Figure 12), has been represented by an ontology containing classes, sub-classes, object type property, data type property, and restrictions.

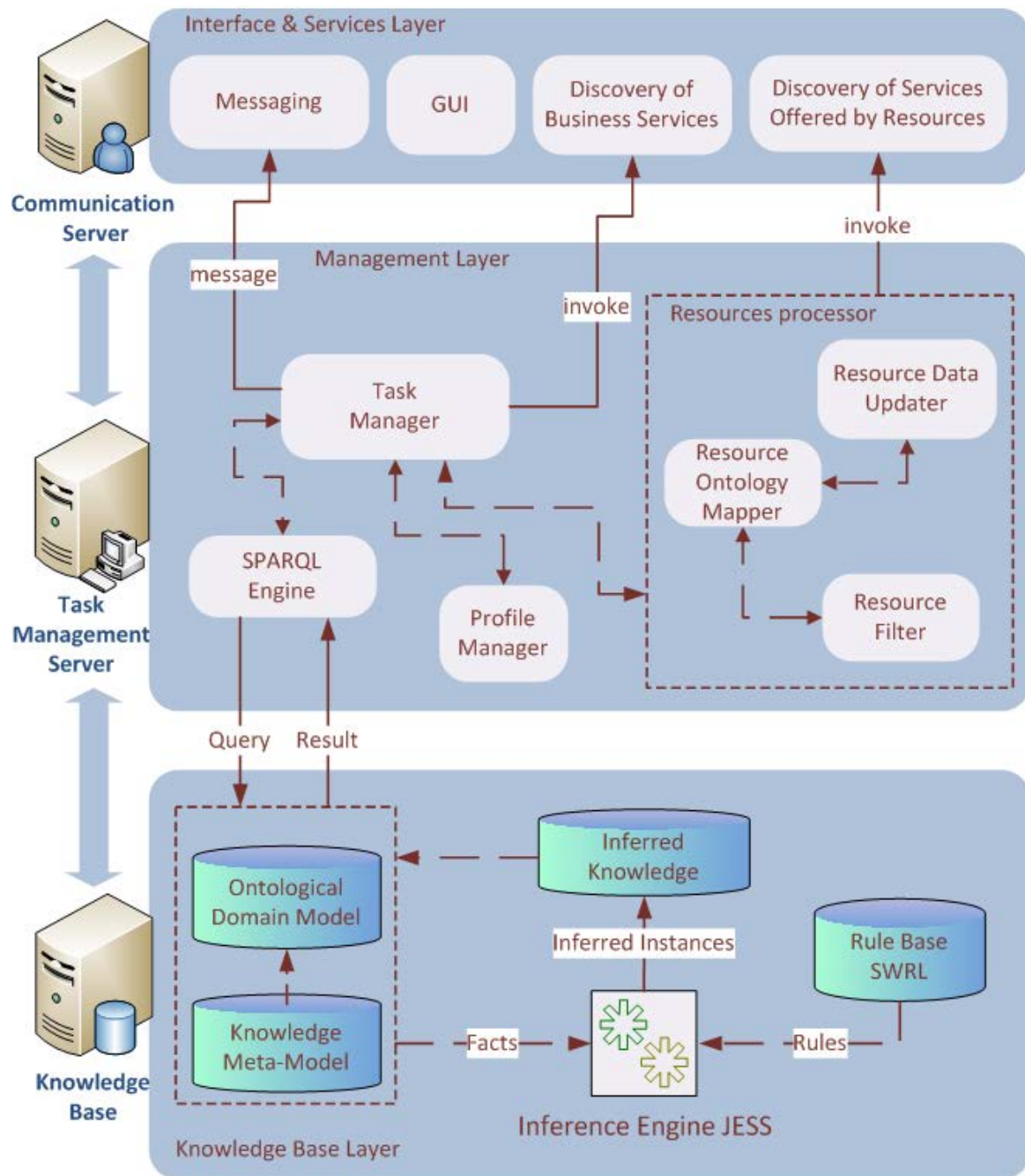


Figure 23. ONOF-PAS Architecture Overview

The rules base has been created to allocate the heterogeneous resources needed by the tasks according to the conditions of the object treated by the task. We have integrated an inference engine at the knowledge base level to perform a rule-based reasoning over the facts stored in the knowledge base. The inferred knowledge is automatically

added to the knowledge base. The main role of the inference engine is to reason on the knowledge represented by the ontologies in order to bind each task with the necessary resources, distributed in different environments and owned or managed by different organizations.

At the management level, we have modeled the different components, i.e. *Task Manager*, *Profile Manager*, *SPARQL Requestor*, and *Resources Processor*, which assure the tasks management and resources allocation.

In the next sections, we provide a full description of these components. We also focus on the Resources Processor component involving the issues of resources ontology mapping as well as resources data updating and filtering. At the interface and services level, we propose a model for the messages management and a policy for messages transmission. The GUI component will be described in chapter VI.

---

## **V.5 Tasks and Resources Management**

---

### **V.5.1 Profile Manager**

Actor identification and authentication can be viewed as major requirements for the use of telemedicine applications from a legal point of view. Therefore, all actors are obliged to accept the use of some certificates, established by an official governmental organisation, that fulfil the legal requirements. The profile manager component manages the actor registration process. It creates an instance of the actor and sets the data type properties of the actor instance, such as an identifier, a password, etc. Figure 24 shows an UML-based profile manager model.

The relation between actor and task has been defined in the knowledge meta-model as “*Actor performs a Task*”. Each task has an accessibility level defined as a data type property in the Task ontology. The accessibility level data element will be used to decide if the actor is illegible to perform such a task or not.

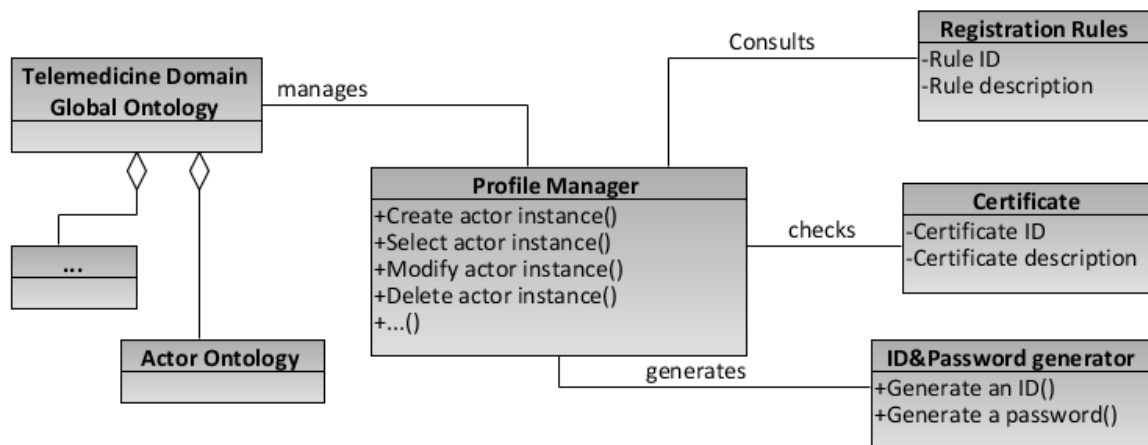


Figure 24. Profile Manager Model

### V.5.2 Tasks Manager and Execution Algorithm

The tasks manager, which is the core component of the management layer; has an orchestrator role within the task execution process. Figure 25 shows the tasks manager model. The following activities are carried out by this component:

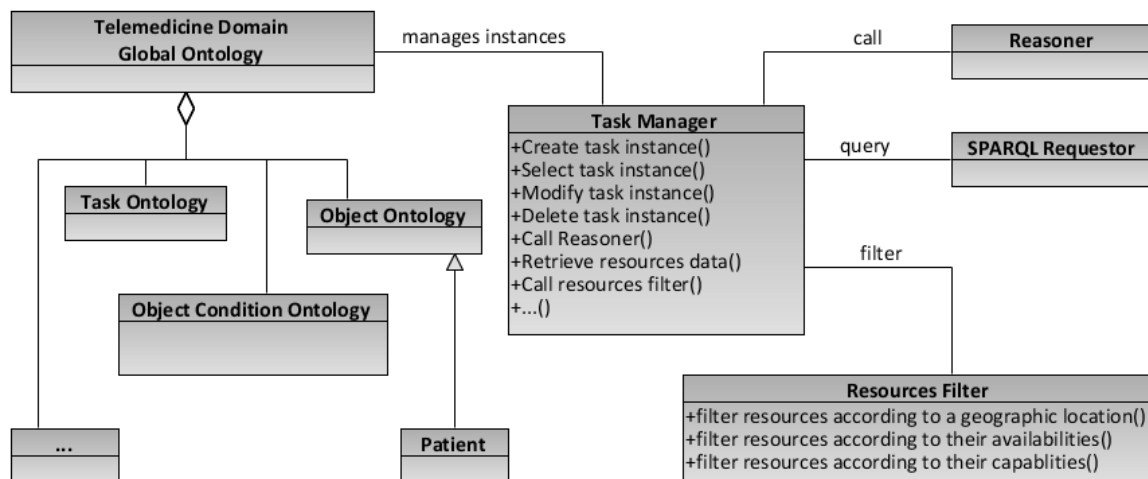


Figure 25. Task Manager Model

- Instantiate the task ontology in order to create task elements instances.
- Set the different data type properties of the task, i.e. task id, task description, task accessibility level, etc.

- Create an instance of the object treated by the task and set the object conditions, for example, create an instance of the patient and set the clinical conditions of the patient.
- Call the Reasoner to effectuate the rule-based reasoning in order to infer the required resources.
- Retrieve the resources data through the SPARQL requestor.
- Call the resources filter in order to get just the resources that meet the user needs.
- Generate one or more messages to be sent to the recipient validated by the user.

Figure 26 displays the different steps of the task execution algorithm. The algorithm inputs are the identifiers of actor, task, and object treated by the task, as well as the object's conditions, in addition to the contextual task parameters, e.g., date, time, and location. The algorithm outputs are a list of required resources to reach the task objectives and a list of organizations for which these resources will be available. The Task/Actor linking phase depends on the restrictions we have defined over the relationship "*performs*" between Actor and Task (Table 3, section IV.6.4.3).

In the reasoning phase, the inference engine applies the defined rules to infer facts related to the task being executed, i.e., actor, resource, organization, and message. More specifically, the inference engine deduces knowledge related to the resources needed by the running task. We have used the SPARQL query language to retrieve the inferred facts, i.e. to retrieve the organizations which have the resources required by the task (we provide an example of SPARQL queries in the next section). Furthermore, the inferred solutions can be classified and filtered according to the context of the user, such as location, technical characteristics of the communication devices used by the sender and receiver, or any other context element. Finally, the messages generated by the tasks are transmitted to the recipients that have been validated by the actor. Thereby, some communications rules are applied to optimize the exchanged messages and to ensure their reception by the recipient actors, as we will see in section V.6.

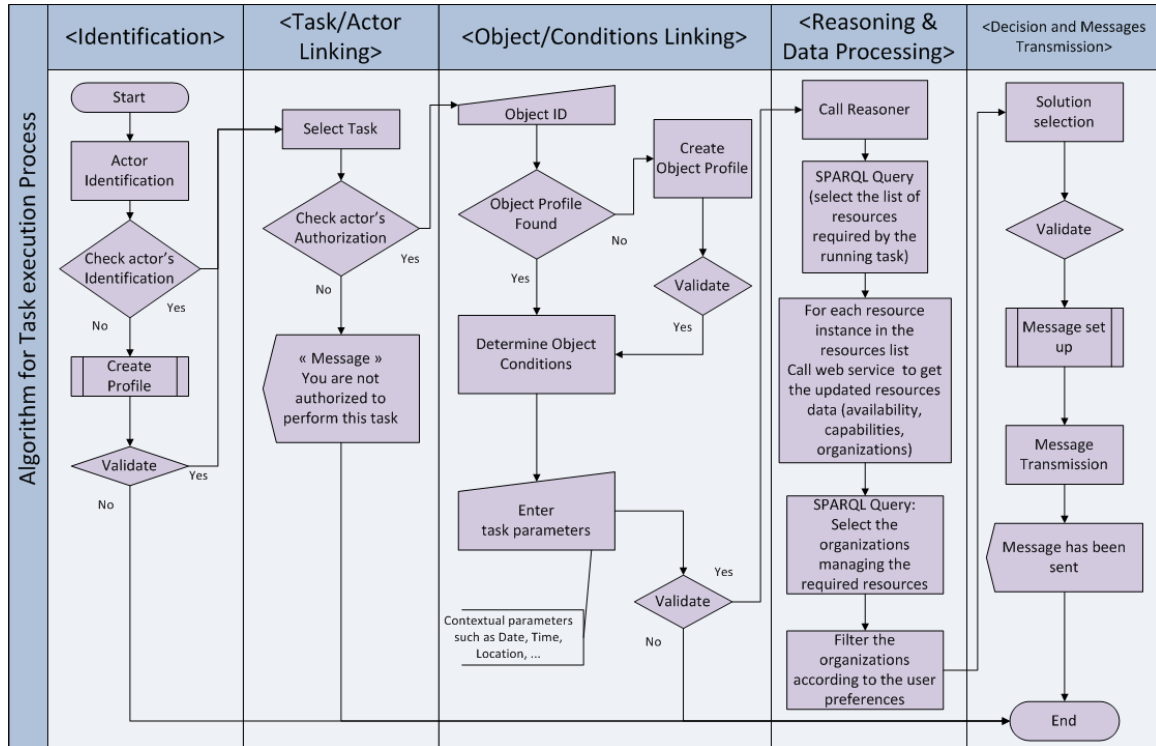


Figure 26. Task Execution Algorithm

### V.5.3 SPARQL Requestor

The SPARQL engine is based on the W3C SPARQL Query Language. SPARQL can be used to express queries across diverse data sources, where the data are stored as RDF graphs or viewed as RDF via middleware. It contains capabilities to query the RDF graphs along with their conjunctions or disjunctions. Therefore, we used this W3C standard query language to retrieve data from the knowledge base through the query forms *SELECT*, *CONSTRUCT*, *ASK* and *DESCRIBE*.

The profile manager and the task manager retrieve the data of the different entities by querying the knowledge base using the SPARQL requestor. The latter will match the graph patterns that contain triples of the type “*subject predicate object*”, where each subject, predicate or object may be a variable. The data returned by the SPARQL queries are also RDF graphs. We give below, in Fragment 9, an example of a SPARQL query over the global telemedicine domain ontology. The query has been written according to the SPARQL syntax. Figure 27 shows a graphic representation of the same SPARQL query.



```

PREFIX                      Telemedicine:      http://www.owl-
ontologies.com/OntologyTelemedicineKnowledgeModel.owl#
SELECT ?Task ?HCP ?Resource ?Message ?Service ?TaskAccessibilityLevel
WHERE {
  ?Task Telemedicine:isPerformedBy ?HCP.
  ?Task Telemedicine:requires ?Resource.
  ?Resource Telemedicine:availability "True".
  ?Resource Telemedicine:isOwnedBy Telemedicine:Lyon_CHU.
  ?Task Telemedicine:generates ?Message.
  ?Task Telemedicine:invokes ?Service.
  ?Task Telemedicine:accessibilityLevel ?TaskAccessibilityLevel.
}

```

Fragment 9. Example of SPARQL Query over the global telemedicine domain ontology.

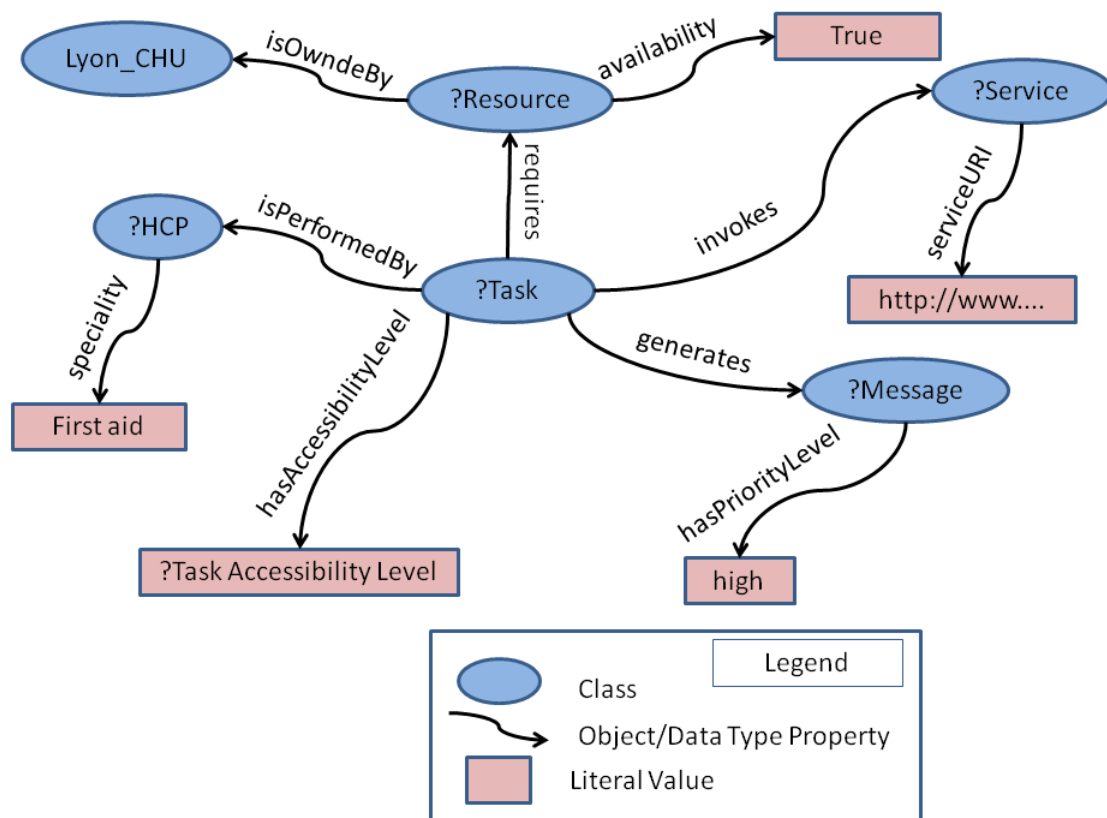


Figure 27. Graphic Representation of the SPARQL Query Described in Fragment 9.

The query example presented in fragment 9 retrieves data, in RDF format, about the task “*?Task*”, the actor “*?HCP*” who performs this task, the resources “*?Resource*” required by the task, the services “*?Service*”

invoked by the task, the messages “*?Message*” generated by the task, and the “*?Task Accessibility Level*” data type.

As we can see in Figure 27, the SPARQL query is composed of two types of nodes. The first one is the entity node which represents a class in the global telemedicine domain ontology. This type of node is linked to the other entity nodes by the object type property, such as *requires*, *generates*, etc. The second type of node is the data type node which represents the literal value related to the entity, for example, the priority level literal value “*High*”, the resource availability literal value “*True*”, etc.

---

#### V.5.4 Heterogeneous Resources Processing

This component is composed of three main elements: Resources Updater, Resources Ontology Mapper, and Resources Filter. We provide hereafter a detailed description of each element.

---

##### V.5.4.1 Mapping to ONOF-PAS Ontologies

In a particular domain, several ontologies can be created to represent knowledge models. The entities of the various ontologies, i.e. concept, property, or instance, can be mapped in order to establish relationships among these models. On the other hand, resources data stored by different organizations are not necessarily represented by ontologies. They might be represented by diverse classical database schemas.

In both cases, no matter the way how resources data are represented, organizations and institutes can exchange what they wish to communicate of their own data. However, interoperability problems might appear at the data semantic level. We need to solve this problem by mapping the concepts coming from different data repositories to our resources ontology concepts. Accordingly, ontology mapping mechanisms will be required to handle the semantic heterogeneity. The question now is how to realize the ontology mapping?

In fact, many works concerning ontology mapping have been performed. A recent state of the art of diverse approaches to semantic interoperability among ontologies has been realized by (Ferreira Da Silva 2007). The author has extracted from the literature several terms related to ontologies mapping such as alignment, merging and integration. We briefly

describe these terms hereafter as they have also been defined by several other authors of the ontology mapping research community.

- **Ontology Alignment:** establishes a collection of binary relations between the vocabularies of two ontologies (Kalfoglou 2003). Ontology alignment can therefore bring two ontologies to mutual agreement to make them compatible and consistent.
- **Ontology Merging:** it means, in general, the sharing of ontology sources, which probably overlap, to generate an unified target ontology (Pinto 2001).
- **Ontology Integration:** it is the ontology construction process by reusing (by combination, aggregation, specialization, etc) other available ontologies. These different ontologies are then part of the new built ontology (Pinto 2001).

Multiple frameworks and tools have been presented in the state of the art realized by (Kalfoglou 2003), such as IF-Map for ontology mapping, the SMART, PROMPT and PROMPT-DIFF tools for the protégé ontology development environment. One of these methods can be integrated and used in the ONOF-PAS architecture to execute ontology mapping. Therefore, creating a new methodology of ontology mapping is not the scope of this thesis.

In the healthcare domain, institutes have their own medical data represented in different standards, i.e. EDI, HL7, openEHR, EHRcom, DICOM, etc. Nowadays, XML is the most widely used standard for data integration. In other words, no matter the type of the organizations' databases, e.g. relational or object-oriented databases, data can be transformed in XML to be communicated to other information systems. This challenge has been largely addressed by the data integration research community. One of our team work members has also addressed this problem as part of his PhD thesis focusing on the mediation automation between relational databases and XML (Jumaa 2010), where an approach of data transformation into XML has been proposed. Then, the transition from XML to OWL can easily be done using XML Schema Definition (XSD). This process is called conceptual normalization which describes how a specific XSD is transformed into the corresponding OWL schema.

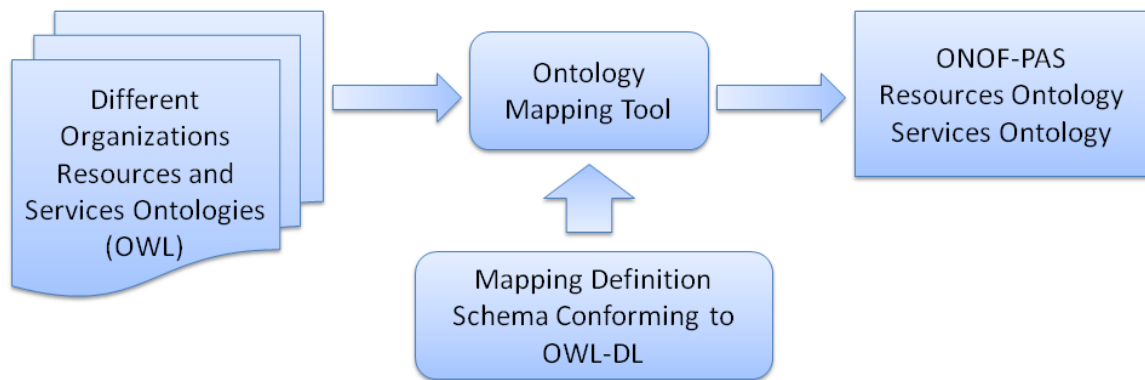


Figure 28. Mapping to ONOF-PAS Resources and Services Ontologies

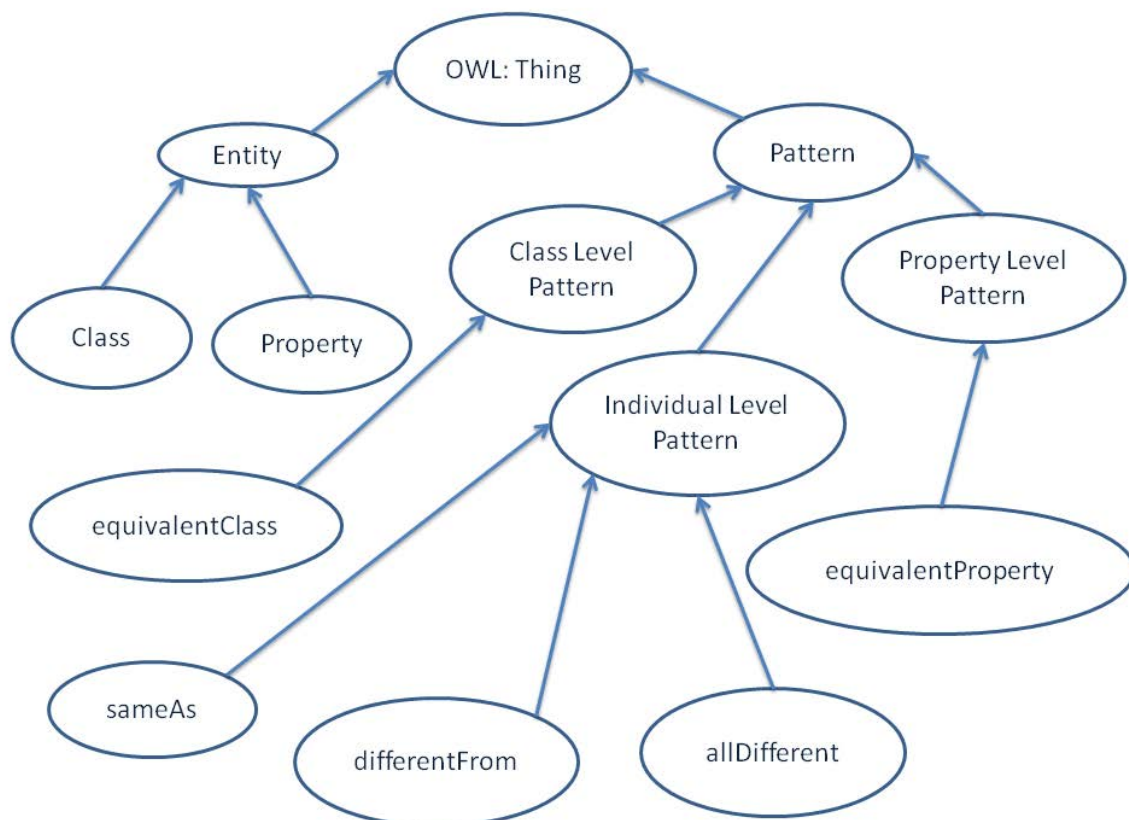


Figure 29. Mapping Definition Conforming to OWL-DL

Figure 28 describes a general mapping process to map various resources and services ontologies defined by different organizations or institutes to the ONOF-PAS resources and services ontologies, using a

mapping definition schema conforming to OWL-DL and an ontology mapping tool.

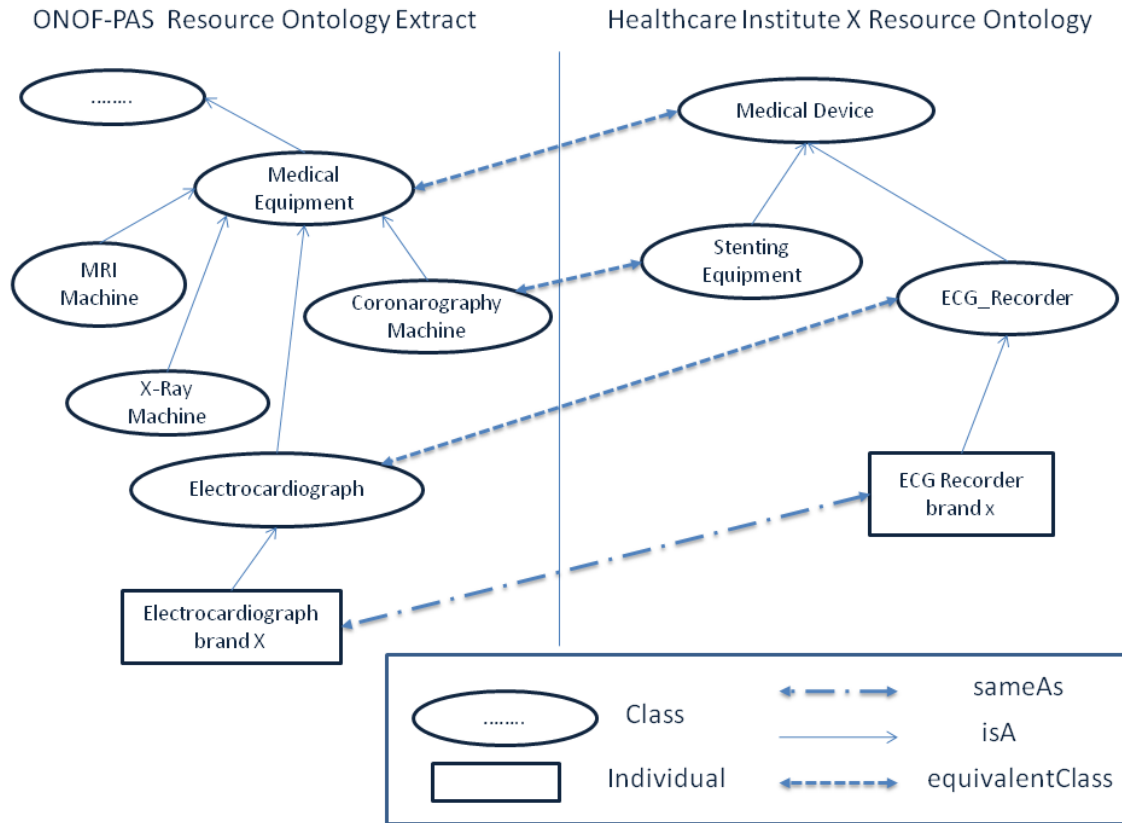


Figure 30. Entity Mapping Example

To perform the mapping process in our work, we have been inspired by the framework created and used in the ARTEMIS project to map an ontology source with an ontology target (Bicer 2005). We thus adopted the same methodology as Bicer to create a mapping schema definition. However, our mapping schema definition conforms to OWL-DL (Figure 29) and is thus easier to implement. To do this, we have included in our mapping definition schema several patterns of the OWL-DL, W3C standard, such as *equivalentClass*, *equivalentProperty*, *sameAs*, *differentFrom*, etc. This mapping definition is then used to automatically transform the source ontology classes and instances into the target ontology classes and instances.

Figure 30 shows a simplified example of a mapping between the ONOF-PAS ontology and an organization X ontology, at the resources ontology level. It is realized using the mapping patterns mentioned above. Two identical classes are mapped through the *owl: equivalentClass* pattern. *Owl: sameAs* specifies that two individuals are identical. The property “*owl: equivalentClass*” is used to indicate that two classes have precisely the same instances.

Considering the definitions given in Figure 31, we might decide that the class “*medical device*” is equivalent to the class “*medical equipment*”. We might also consider that the individual “*ecg\_recorder brand x*” is the same as “*electrocardiograph brand x*”. The object type properties and data type properties can be mapped using the “*owl: equivalentProperty*” pattern. For example, *isA (?ECG\_Recorder, Medical Device)* is a property that is equivalent to *isA(Electrocardiograph, Medical Equipment)*. In the same way, all classes, individuals, and properties of any healthcare institute’s resources ontologies can be mapped to the ONOF-PAS ontologies.

---

#### V.5.4.2 Using Web Services for Resources Data Updating

In the real world, the context of the resources is permanently changing. For this reason, resources data must be constantly updated in order that the ONOF-PAS system allocates only the available resources to the tasks. The role of the resources data updater is to maintain the resources data up to date by invoking and executing the resources data web services provided by the organizations that own or manage these resources.

In the application scenario we have presented earlier, it is assumed that the different organizations, i.e. hospitals, medical centers, insurance companies, etc, own or manage the heterogeneous resources needed by the tasks, and that they expose what they want to communicate about their resources data, using web services. Indeed, nowadays, the Web Service (WS) technology provides organizations an interesting interoperability solution in terms of data exchange (Alonso 2004).

WS has been defined by the W3C Web Service Architecture Working Group as follows: “A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically

WSDL). Other systems interact with the Web service in a manner prescribed by its description using Simple Object Access Protocol messages (SOAP), typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards.” (Booth 2003).

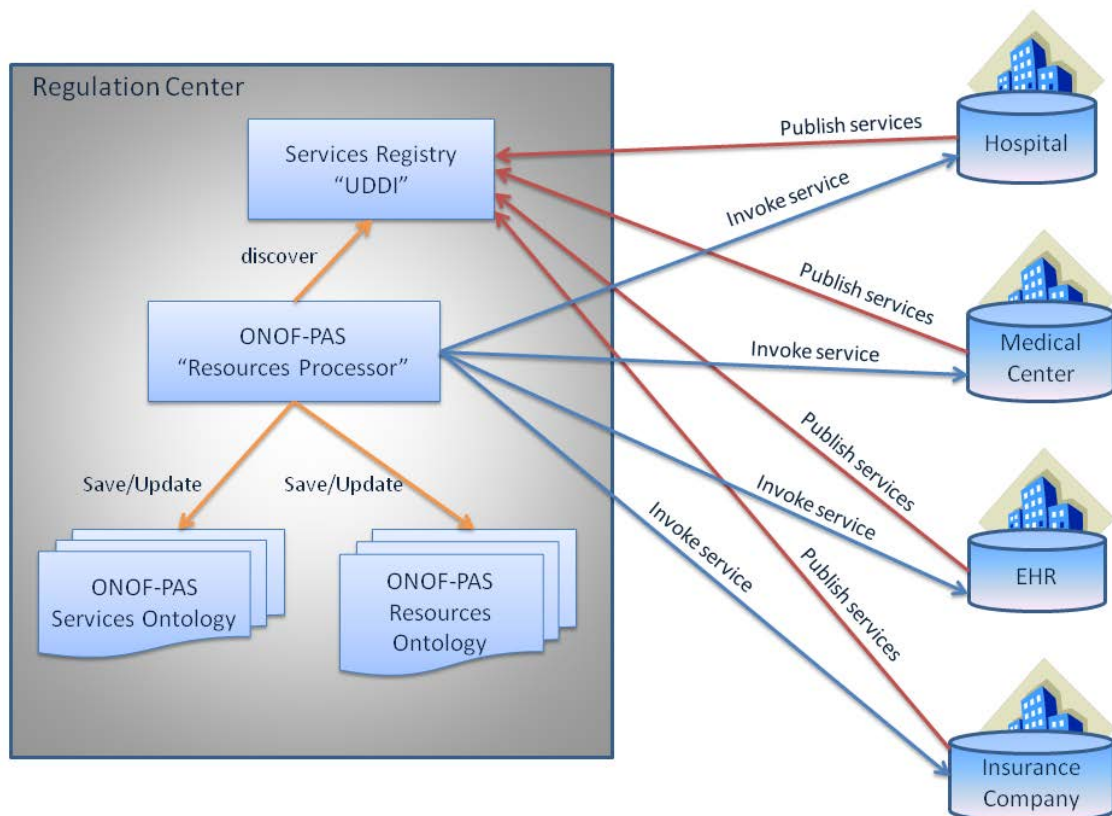


Figure 31. The ONOF-PAS Resources Processor Shall Request the Services Registry to Discover the Web Services Related to Resources Data Updating.

Since its beginning, openness and interoperability have probably been the keywords associated to the WS technology. The history of web services started back in the year 2000 when the first recommendations were published by organizations like W3C and OASIS, in conjunction with all major IT actors (IBM, SAP, Microsoft, HP...). In 2002, the three main standards (SOAP, WSDL, UDDI) of the WS technology were widely accepted by the community, leading to the first real world implementations and concrete solutions provided by software editors. SOAP defines a

communication protocol for Web services. WSDL (Web Service Description Language) enables service providers to describe their services. UDDI (Universal Discovery Description and Integration) offers a registry service that allows advertisement and discovery of Web services. From 2002 to now, the main goal of the WS actors was to turn this technology into an enterprise-ready solution. Many standards recommendations have been proposed solving specific WS issues, viz. security, reliability, etc (Le Blevec 2008).

Each healthcare institute has its own resources ontology representing heterogeneous resources, viz. services, devices, and knowledge sources. These ontologies can be imported and mapped to the ONOF-PAS ontologies using a mapping mechanism as we have seen in the previous section. Then, on the other side, the organizations resources data can be stored as instances in the ONOF-PAS resources ontology. Additionally, the healthcare institutes should declare or publish the resources data update services they provide to the services registry, as displayed in Figure 31. Consequently, healthcare institutions using the WS technology may act as a WS provider and supply data about the availability and capabilities of the different resources they own or manage. These services can be invoked and executed by the resources data updater of ONOF-PAS, which behaves as a WS consumer. Meanwhile, the invoked services will be mapped and stored as instances in the ONOF-PAS services ontology for a future use.

Table 5. Typical Resources Services Examples

| Service Signature       | Service Semantics  | Constraints on Input/Output Parameters |
|-------------------------|--|--|
| <b>S1 (!Rlab, ?Ra)</b>  | Get the availability of a given resource " <i>Ra</i> " which has the label " <i>Rlab</i> " | Ra = true                              |
| <b>S2 (!Rid, ?Rd)</b>   | Get the description of a given resource " <i>Rd</i> " which has the id " <i>Rid</i> "      |  |
| <b>S3 (!Rid, ?Rq)</b>   | Get the quantity of a given resource " <i>Rq</i> " which has the id " <i>Rid</i> "         | Rq > 10                                |
| <b>S4 (!Rlab, ?Rd)</b>  | Get the description of a given resource " <i>Rd</i> " which has the label " <i>Rlab</i> "  |  |
| <b>S5 (!Rid, ?Rlab)</b> | Get the label of a given resource " <i>Rlab</i> " which has the id " <i>Rid</i> "          |  |

In the resources ontology, we have defined multiple data type properties, i.e. *resource id*, *label*, *description*, *availability*, and *capabilities*.



Table 5 lists several typical examples of resources data web services that can be used to retrieve and update resources data.

Each service has input/output parameters, semantic description, and constraints on input/output parameters. For example, service S1 has as input resource label “!Rlab” and as output resource availability “?Ra”. Service S3 takes the resource identifier as input parameter “!Rid” and returns the resource quantity “?Rq”, and so on for the other services.

---

#### V.5.4.3 Discussion about Resources Data Update Services Discovery

Web services discovery is the process of finding web services with a set of given capabilities. In general, web services advertise their capabilities through a services registry. UDDI has been designed to be interrogated by SOAP messages and to provide access to Web Services Description Language documents describing the protocol bindings and message formats required to interact with the web services listed in its directory. UDDI registries help administrators to manage deployed services and allow client applications to dynamically access services metadata, such as WSDL documents and taxonomies.

Figure 31 shows different healthcare institutes (hospitals, medical centers, EHR hosts, insurance companies ...) that are supposed to publish resources data web services to a services registry that may be located at a regulation center. In addition, WSs can be used to retrieve patient medical data stored in different EHR hosts. Especially when the patient’s profile is not locally available, then data from the patient’s electronic health record (EHR) can be retrieved by using WSs where the patient ID or his social security number as well as the Uniform Resource Identifiers (URIs) of the services provided by the EHR hosts may be required for data access.

The registry role is both to store the advertisement of the capabilities of the WSs and to match the requests of the WSs consumers with the advertisements. Web services capabilities correspond to their provided functionalities. At this stage, we should discuss two essential issues which are capability representation and capability matching.

---

#### V.5.4.3.1 Capability Representation

There are two possible approaches to represent capabilities of a service (McIlraith 2001). The first approach is based on the service semantic, which means that the capabilities are represented by a concept defined within an extensive taxonomy/ontology enumerating all the actions/functions that may exist in a particular domain. Figure 32 displays an example of a simple ontology specifying the taxonomy of services in the cardiology domain. Using such an ontology, web services such as those offered by a cardiological hospital can be defined as instances of classes that represent their capabilities. The cardiological hospital of Lyon, for example, may advertise services that are instances of the ontology classes “cardiac implantable devices”, “cardiac device for heart failure”, “cardiac catheterization”, etc.

In the second approach, the capability is represented by the “state transformation” induced by executing the service. The state transformation is represented by IOPE, standing for Input, Output, Preconditions, and Effects. For example, a bookselling service may be represented by the following IOPE concepts: the input is an address or a credit card number, the precondition is a valid credit card, the output is “book is shipped”, the effect is “card is charged”.

Both of these approaches to represent functionalities (i.e. capabilities) have their advantages and their drawbacks. Thus, the use of one approach or the other depends on the task needs. The use of an ontology enumerating all potential capabilities within a specified application domain facilitates the discovery process since the matching process is reduced to a subsumption test between capabilities in the used ontology. On the other hand, enumerating all possible capabilities, even in restricted domains, may be difficult (if not impossible).

Despite their differences, both of these approaches necessitate the use of domain ontologies to represent the semantics of the different concepts that make up the service’s capability (i.e. its performed actions and/or its inputs, outputs, preconditions and effects). Semantic Web service languages like for example OWL-S and WSDL-S, are used to make the connection between the described web services and the ontological concepts that capture their semantics.

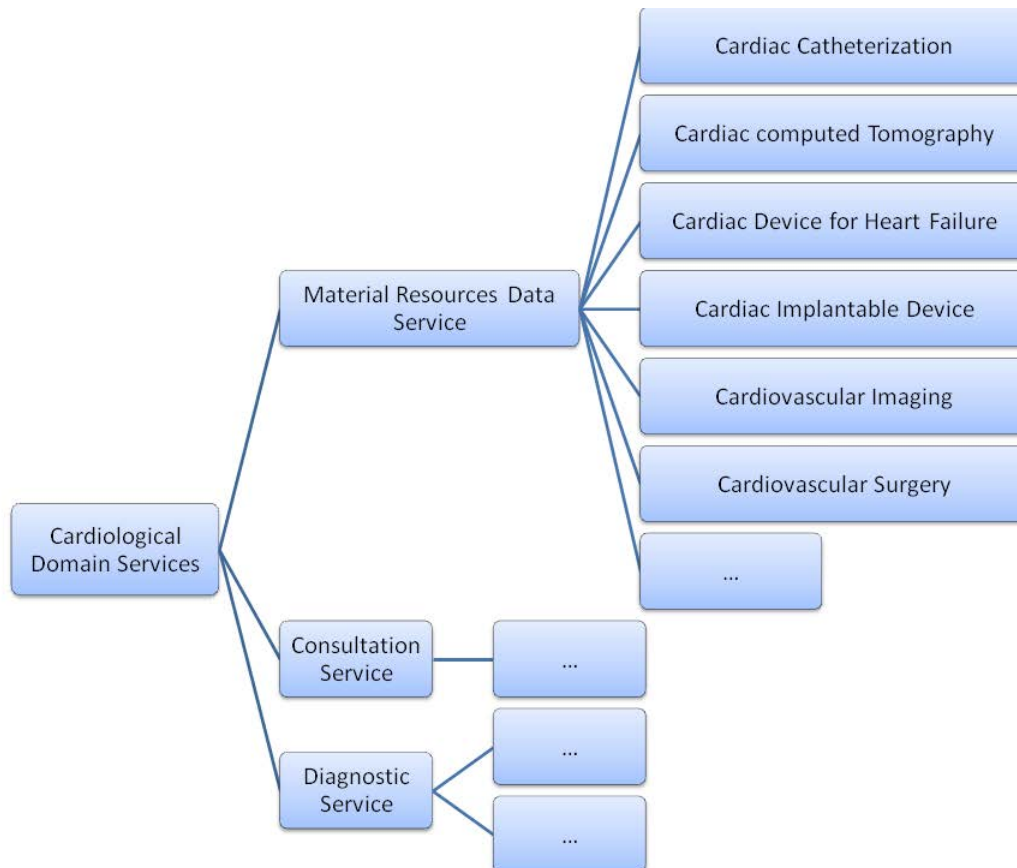


Figure 32. An Example of an Ontology Specifying the Taxonomy of Some Services in a Cardiological Hospital

#### V.5.4.3.2 Matching Services Capabilities with Requestor Needs

In the capability matching process, the capabilities of the advertised services are compared to the capabilities needed by the requestor. The goal is to find the advertiser that produces the results required for the requestor. In general, advertised services might not exactly match the requested service.

Many capability matching algorithms have been proposed for semantic web services in the literature (Li 2003). In general, they exploit one of the two aforementioned approaches to capability representation.

Several interesting approaches, which assume that capabilities are described by the state transformation produced by the web service, have been proposed by (Colucci 2004), (Paolucci 2002). Services are described here using OWL-S. Indeed, OWL-S allows describing services as state

transformations using the Service Profile which provides a view of the web service as a process that requires inputs and some preconditions to be valid, and results in outputs and has some effects that become true. The matchmakers in these works compare the state transformation described in each advertisement to the one described in the request. They perform two matches, one comparing the outputs and one comparing the inputs. If the output required by the requestor is of a kind covered (subsumed) by the advertisement, then the inputs are checked. If the inputs specified in the request are subsumed by the input types acceptable to the service, then the service is a candidate to accomplish the requestor's requirement. In reality, there is an asymmetry between the matching of the inputs and the outputs of a web service. Ultimately, the requestor needs a web service that produces the desired outputs. Once the web service that provides the desired outputs has been found, the requestor can either attempt to satisfy all the inputs, or use its own composition capabilities to find other web services that can provide the desired inputs (Barhamgi 2010).

Both capabilities representation approaches can be implemented in ONOF-PAS. However, for resources data services updating, the service semantic based approach is more interesting than the state transformation one, because by a simple concept-based research in the services registry, several services related to resources data can be discovered and then stored in the ONOF-PAS service ontology.

---

#### V.5.4.4 SPARQL Queries Based Resources Filtering

In the resource ontology, we have defined several data type properties, i.e. resource *id*, *label*, *description*, *domain of use*, *quantity*, *availability*, and *capability*. The rules link the tasks with the required resources according to the conditions of the object treated by the tasks. However, resources conditions are permanently changing. For instance, some resources may be not available or are not accessible at the time of the task execution. For example, a device may have broken down, a data source has been blocked, a service does not respond or is not available, etc. Consequently, the task execution will be disrupted. On the other hand, the inference engine will infer all the resources required by the task, no matter if these resources are available or not. To overcome this problem, we decided to integrate a resources filter in the ONOF-PAS architecture (see Figure 23 and Figure

25). We may thus filter the inferred resources according to certain requirements which can be related to the following factors:

- Resource availability, no matter if this resource is logistical, a medical equipment or even a human resource.
- Resource domain of use, i.e. for which objectives a given resource can be used.
- Technical characteristics of a given material or communication resource.
- Different parameters of quality of services (i.e. response time, processing time, availability, accessibility, etc) provided by different organizations.
- Geographical location of the organization which owns or manages the resource.

The main functionality of the resources filter is to perform a filtering process over the inferred resources in order to obtain just those resources which meet the requirements mentioned above.

In fact, we can carry out the resources filtering process by two methods. The first one is a simple one and can be realized by integrating one or more constraints in a SPARQL query. A constraint, expressed by the keyword `FILTER`, is a restriction on solutions over the whole group in which the filter appears. SPARQL `FILTER`s can set up restrictions by implementing arithmetic expressions (i.e. `>`, `<`, `<=`, `>=`, etc). The following examples clarify the use of SPARQL `FILTER`s to filter solutions. In the first example, Fragment 10, we use a filter in the SPARQL query to obtain just the resources instances, required by the task, whose data type property *“availability”* have the literal value *“true”* and the data type property *“quantity”* have the literal value more than 5. In the second example, Fragment 10, we use both the `FILTER` and the `OPTIONAL` patterns to obtain the instances of the resources used in the domain of cardiology, where the keyword `OPTIONAL` is used to restrict the selection in order to get just the resources used in the cardiology domain and owned by the Lyon University Hospital Center *“CHU\_Lyon”*.

The second method to filter solutions can be implemented using SWRL rules. In this approach, we will add filtering rules into the rules base. The filtering rules will be applied by the inference engine. We provide hereafter, in Fragment 11, an example of filtering rules described

in SWRL. In this example, we integrated, at line 7, the clause “*if the intensive care unit availability is true*”, to associate the resource “intensive care unit” to the task. In the same way, we can add multiple rules to take into account the requirements mentioned above concerning resources filtering.

```

PREFIX                                Telemedicine:                        http://www.owl-
ontologies.com/OntologyTelemedicineKnowledgeModel.owl#

Example 1:
SELECT ?resource ?availability
WHERE {
  ?task Telemedicine:requires ?resource.
  ?resource Telemedicine:availability ?res_ava.
  ?resource Telemedicine:quantity ?res_quantity.
  FILTER regex (?res_ava, "true")
  FILTER regex(?res_quantity "> 5").
}

Example 2:
SELECT ?resource ?organization ?res_domain
WHERE{
  ?task Telemedicine:requires ?resource.
  ?resource Task Telemedicine:domainOfUse ?res_domain.
  ?resource Task Telemedicine:isOwnedBy ?organization.
  FILTER regex (?res_domain, "Cardiology")
  OPTIONAL (?organization, "CHU_Lyon")
}

```

Fragment 10. An Example of a SPARQL Query implementing Resources Filtering

```

Telemedicine:Task(?Telemedicine:task) ∧
Telemedicine:Patient(?Telemedicine:p) ∧
Telemedicine:Chest_Pain(?Telemedicine:cp) ∧
Telemedicine:Concerns(?Telemedicine:tas, ?Telemedicine:p) ∧
Telemedicine:hasClinicalCondition(?Telemedicine:p, ?Telemedicine:cp) ∧
Telemedicine:Intensive_Care_Unit(?Telemedicine:icu) ∧
Telemedicine:availability(?Telemedicine:icu, "true")
→
Telemedicine:requires(?Telemedicine:tas, ?Telemedicine:icu)

```

Fragment 11. Example of a SWRL Rule implementing Resources Filtering

Table 6. Comparison of Resources Filtering Methods

|   | Rapidity           | Flexibility        | Expressiveness |
|---|--------------------|--------------------|----------------|
| Resource Filtering Using SPARQL Filters | +                  | +                  | +              |
| Resource Filtering Using SWRL rules     | –                  | –                  | +              |
| Notation                                | + more appropriate | - less appropriate |                |

However, adding more rules will negatively affect the reasoning performance, in terms of rapidity and executing time. In addition, as SWRL rules are static and not very flexible, it means that every time we need to add or remove some filtering criterions we should add or remove the corresponding rules which may also require a preliminary reasoning test to verify that the inference engine does not lead to unexpected results. At the expressiveness level, we think however that there is no significant difference between the two formalisms. For the previous reasons, we thus opted to adopt the first method of resources filtering using SPARQL Filters in the query form. Table 6 shows a simple comparison of both methods of resources filtering.

---

## **V.6      Messaging**

As displayed in Figure 23, actors interact with the system through the interface layer, which consists in three main components (i.e. messaging, resources data services discovery, and Graphic User Interface GUI) providing several functionalities, such as identification, authentication, profile management, and task execution according to the actors' profiles. We have already discussed the resources data services discovery in the previous section, considering that it is related to resources processing. In the following sub-sections, we provide a description of the messaging components whereas we will describe the GUI component in the next chapter.

---

### **V.6.1    Messages Management Model**

Once the resources have been inferred by the inference engine and filtered by the resources filter, the ONOF-PAS system proposes to the user a list of solutions related to the objectives of the task he/she performs. For example, in our patient transfer scenario, the solutions can be a list of hospitals that own the resources required to perform a specific treatment for the patient (e.g. a coronarography procedure). Once the user selects and validates a solution, the task manager generates one or more messages to be sent to the corresponding recipients, e.g. a notification message or an information message. The messages generated by the task manager can be sent in the

XML format and may encapsulate different types of data (i.e. textual, image, audio, video, signals, etc).

XML is playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. It is now considered as a common and independent data format that provides standard data types and structures independent of any programming language, development environment, or software system. This independent data format also provides ubiquitous software for handling operations in XML, including parsers, queries, and transformations. In Fragment 12, we give a typical example of an XML message that respects the XML syntax.

```
<?xml version=" 1 . 0 " encoding="UTF-8"?>
<message>
  <date>
    <day>14</day>
    <month>07</month>
    <year>2011</ year>
  </date>
  <to>Dr. Smeth</to>
  <from>First-aid-worker</from>
  <heading>Notification_Patient_Arrival</heading>
  <body>the patient Mr. X, being transporting to CHU_Lyon hospital</body>
</message>
```

Fragment 12. XML Message Typical Example

In our ONOF-PAS framework, we have defined a messages ontology that is composed of two main classes, the notifications message class and the information message class (Figure 33). The messages class has object type properties such as *generatedBy*, *isSentBy*, and *isSentTo*. Additionally, the messages class has the following data type properties: *messagePrioritylevel*, *messageDelivered*, and *messageLifeCycle*.

The principal object in the messages management model is the message manager, which provides the following functionalities: *send message*, *save message*, *get message*, and *stratify message*. In the next sub-sections, we propose a policy for messages transmission including messages prioritizing and messages stratification.



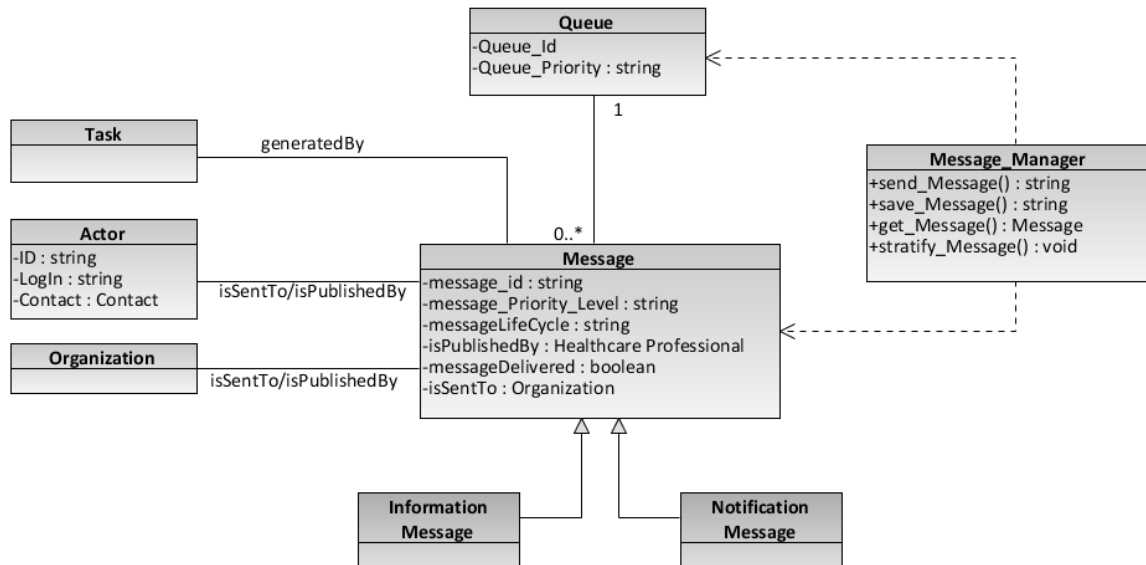


Figure 33. Messages Management Model

## V.6.2 Messages Transmission Policy

The communication model integrates a messages transmission policy which is based on two principal techniques: messages prioritization and messages stratification. In the following, we clarify what we mean by these techniques.

### V.6.2.1 Messages Prioritization

The process of messages prioritisation is based on two factors which are the task domain and the profile of the actor who performs this task. The priority attribute refers to the emergency degree of the message being transmitted. Additionally, it implies the need for a response in a determinate time limit. Moreover, as suggested in section IV.7, the whole process can be automatically performed through predefined SWRL rules in the knowledge base.

In Fragment 13, we give two examples of such SWRL rules, specifying a priority level to each message generated by a task as follows:

- If an emergency physician performs a task, then assign the messages generated by the task, with a “high” priority level.
- If a specialist performs a task, then assign the messages generated by the

task, with a “normal” priority level.

**Rule 1:**

EmergencyPhysician (?ep)  $\wedge$  Telemedicine Task (?task)  $\wedge$  Message (?m)  $\wedge$  performs (?ep, ?task)  $\wedge$  generate(?task,?m)  
 $\rightarrow$  messagePriorityLevel (?m, “high”)

**Rule 2:**

Cardiologist (?c)  $\wedge$  Telemedicine Task (?task)  $\wedge$  Message (?m)  $\wedge$  performs (?c, ?task)  $\wedge$  generate(?task,?m)  
 $\rightarrow$  messagePriorityLevel (?m, “normal”)

**Rule 3:**

...

Fragment 13. Messages prioritizing rules examples described in SWRL

Table 7 provides examples of different healthcare professionals types and the priority levels of the messages exchanged between them. We suppose that: P1= high priority level, P2= normal priority level and P3= low priority level.

It is possible to restrict the messages destinations selection according to other technical factors adding other rules concerning the technical characteristics of the actor’s terminals such as the memory size and the telecommunication technologies embedded in these communication devices (i.e. GPRS, UMTS, EDGE, HSDPA, HSUPA, etc). In this case, the messages will be sent to the recipients who have a terminal embedding the largest size of memory and the telecommunication technology which provides the highest data transmission rate.

Table 7. Examples of Messages Priority Level

| Messages exchanged among healthcare professionals | Emergency physician | Specialist | General Practitioner | Nurse |
|---|---------------------|------------|----------------------|-------|
| Emergency physician                               | P1                  | P1         | P1                   | P1    |
| Specialist  | P1                  | P2         | P2                   | P2    |
| General Practitioner                              | P1                  | P2         | P2                   | P3    |
| Nurse   | P1                  | P2         | P2                   | P3    |

#### V.6.2.2 Messages Stratification

To take into consideration the time factor, which is crucial in some specific applications scenarios, we have integrated a technique of messages stratification into the messages transmission policy. Actually, this

technique is mainly based on the messages prioritisation explained above. Additionally, within the messages stratification process, messages generated by the different tasks are classified in queues according to their priorities.

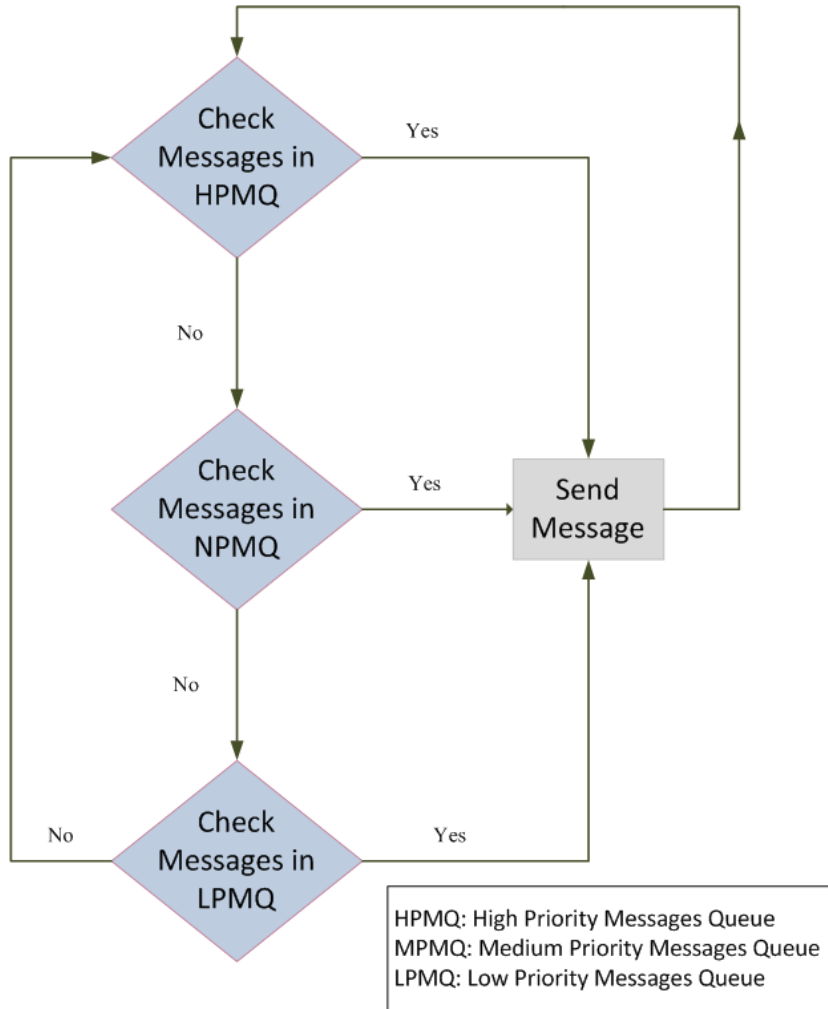


Figure 34. Messages Transmission Algorithm

To achieve this objective, we have defined three types of messages queues which are:

- High Priority Messages Queue HPMQ.
- Normal Priority Message Queue NPMQ.
- Low Priority Messages Queue LPMQ.

The HPMQ contains the messages assigned with a high priority level, and so on for NPMQ and LPMQ.

We have defined a simple algorithm for messages transmission as depicted in Figure 34.

Therefore, the principal messages management model functionalities are: (1) Identify the priorities of the messages coming from the task management layer according to the sender and receiver actor. (2) Stratify the messages in queues according to their priorities. (3) Execute the messages transmission algorithm as follows:

- Send the first message in the HPMQ.
- If there is no other message in the HPMQ, send the first message in the NPMQ.
- If there is no other message in the NPMQ, send the first message in the LPMQ.

The messages management model should take the time factor into consideration to avoid that, in some cases, the messages with a low priority level stay waiting for a long time. For that, we have defined the data type property “messageLifeCycle”. Thus, when the time of a message life cycle expires and the message has not been sent yet, then its priority will change to a higher level and simultaneously the “messageLifeCycle” time will be reinitialized. In this way the messages queues congestion will be avoided.

---

## V.7 Conclusion

In this chapter, we have described the phase number 4 of the ONOF-PAS construction methodology concerning the building of a task-centred architecture model which permits an intelligent operation of the heterogeneous resources required by a given task. We have provided a full description of the different ONOF-PAS architecture components including the task management and the resources processing components.

We have addressed the different issues related to resources ontologies mapping and resources data services discovery. The use of web services is restricted to resources data updating. However, web services may also be used to retrieve general information, e.g. forecast, road circulation status, etc. Finally, we have proposed a policy for messages

transmission based on messages prioritizing and stratification techniques as well as a simple algorithm for messages transmission.

Based on the proposed architecture model, we have succeeded to link four crucial factors which are: the actor profile, the task type, the status of the object treated by the task, and the resources conditions in terms of availability, capabilities and accessibility.

In the next chapter, we will present the last phase of our ONOF-PAS construction methodology. We will describe a prototype implementing the proposed ONOF-PAS system to handle the earlier presented scenario of patient orientation decision making support in inimical environments, and, we will demonstrate how the required resources can be automatically inferred and processed to meet the user needs in such telemedicine scenarios and environments.

## *Chapter VI*

# *ONOF-PAS System Implementation*



# Chapter VI ONOF-PAS System Implementation

## Overview

Research prototypes are extremely valuable for performing the proof of concepts and ideas. In this chapter, we realize the last phase of our ONOF-PAS construction methodology (Figure 11, section IV.4). We present the implementation of the main ONOF-PAS architecture components using different technologies and development environments. Therefore, we present, in the first step, the necessary tools used during the development process, particularly the ontology development tools as well as the tools needed for the implementation of the telemedicine tasks, with illustrations about the different features available. Finally, we present an example of a patient orientation task including the generation of the required resources.

We have performed the implementation process using two Integration Developing Environments (IDE). The first one is the Protégé IDE which has been used to edit the different ontologies of ONOF-PAS, to describe the SWRL rules, and to realize rule-based reasoning to infer the resources required by the task. The second IDE is Netbeans to develop the TTROIE prototype using the J2EE technology.

---

### VI.1 Development Tools

In this section, we introduce the tools used during the implementation process. These tools include: ontology development tools, i.e. the Protégé environment and its plug-ins SWRLTab which permits SWRL rules editing and management, as well as the inference engine JESS (Java Expert System Shell), J2EE, and the Application Programming Interface (API) Jena.

---

#### VI.1.1 Ontology Development Tools

---

##### VI.1.1.1 Protégé Integrated Development Environment

The core element in our ONOF-PAS architecture is the knowledge base which is composed of several ontological models, i.e. *Actor Ontology*, *Task Ontology*, *Resource Ontology*, etc (Figure 35).



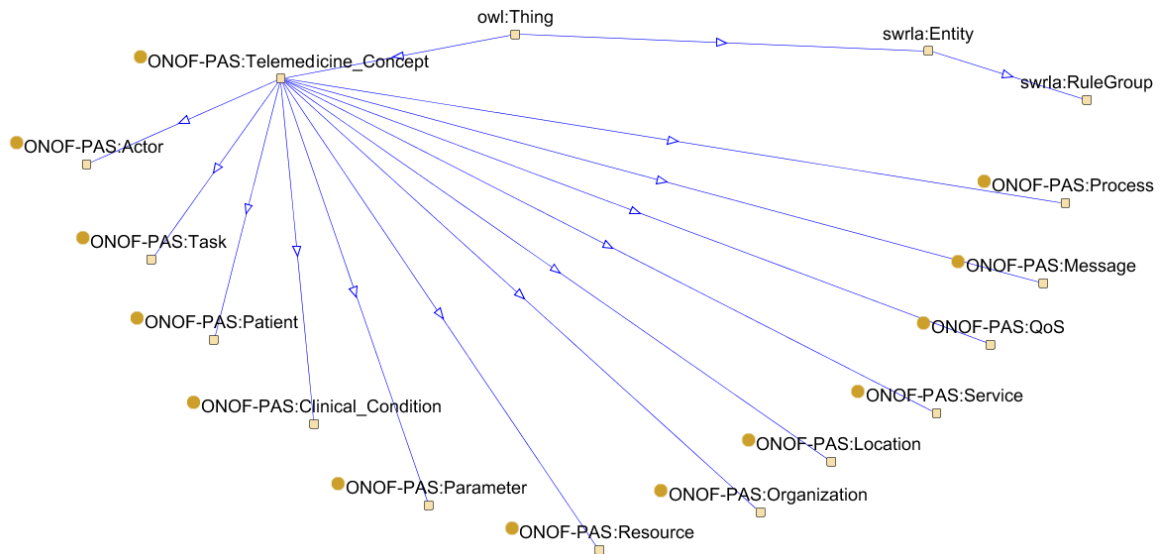


Figure 35. Graphical Representation of the Main ONOF-PAS Ontologies, issued by the Protégé Jambalaya Plug-in

The development of these ontologies is a fundamental step. Thus, we need a tool that allows us to create SWRL rules permitting the inference of the heterogeneous resources required by the different tasks. Consequently, for the ontologies implementation and the rules creation, we have decided to use the Protégé version 3.4 development environment which allows to build an ontology for a given domain, to define data entry forms, and using these forms to acquire data as instances of this ontology. Our decision to choose Protégé is based on several reasons. Indeed, Protégé has been created by the Stanford University and it is widely used, by both the academic and the industrial community, in the field of semantic web and knowledge representation. Moreover, Protégé is a Java library, whose code is open source. It has been extended to create real knowledge-based applications using inference engines to reason and infer new facts by applying inference rules on the ontology instances. Additionally, Protégé has a simple, easy, and friendly user interface. Finally, Protégé does not consist in a simple modelling application, but rather an open architecture which supports a wide variety of plug-ins for knowledge modelling.

#### VI.1.1.2 SWRLTab

As we have previously described, the rules are formalized using the standard language SWRL. Additionally, the creation of rules is an essential

step in our methodology. It makes the ontologies semantically richer and facilitates the automation performed by the inference engine.

To implement SWRL rules, we have used the SWRLTab plug-in, which is an extension to the Protégé-OWL plug-in that permits creating and editing SWRL rules.

---

#### VI.1.1.3 Inference Engines

One of the great advantages of using ontologies is the reasoning capabilities on the knowledge base. To effectively represent the semantic, an ontology must allow automatic detection of the semantic connections between different concepts. Thus, new knowledge may be inferred from any explicitly defined knowledge. Another advantage of the reasoning lies in the ability to maintain the ontology consistency during its evolution.

In fact, in the last version of Protégé (version 3.4) it is possible to directly call a Reasoner, such as Pellet. The Pellet Reasoner (version 1.5.2) can also be used for inferring supplementary facts from the knowledge defined in an ontology. Pellet has been developed and is commercially supported by Clark and Parsia LLC (Pellet 2007). Its source code is implemented in Java and is available as open source. Moreover, Pellet supports the total expressiveness of OWL DL and allows inferences on the ontologies represented in OWL DL. However, Pellet does not effectuate inferences on rules defined in SWRL. Therefore, we will only use Pellet to check the ontology consistency and to infer the asserted class hierarchy.

On the other hand, there are a large number of rule engines which work well with Java, and many are available as open source software. One of the most popular rule engines is JESS (Java Expert System Shell) (JESS 2008). Several projects have successfully used JESS within the Protégé environment, e.g. JessTab and SWRLJessTab. JESS proposes an environment for creating rule-based systems. It includes a rule engine which permits to construct systems able to reason using the knowledge provided in declarative rules forms. Jess is available for research communities under free academic licences. It provides both an interactive interface command line and a Java-based API to its rule engine. The JESS system consists of a rule base, a fact base, and an execution engine. The execution engine matches facts in the facts base with the rules in the rule base (see a comparison of existing inference engines in appendix A).

When a SWRL rule is triggered, it can create new classifications for existing individuals. For example, if a rule consequent asserts that an individual is to be classified as a member of a particular class, that individual must be made a member of that class within OWL when the rule antecedent is satisfied. Similarly, if a SWRL rule asserts that two individuals are related via a particular property, then that property must be associated with each individual that satisfies the rule. Thus, four main tasks must be performed to allow Jess to interoperate with the SWRL Editor. These tasks are:

- Represent any relevant knowledge about OWL individuals as Jess facts.
- Represent SWRL rules as Jess rules.
- Perform inferences using these rules and reflect the results of these inferences in an OWL knowledge base.
- Control these interactions by means of a graphical interface.

---

### **VI.1.2 Prototype Application Implementation Tools**

---

#### **VI.1.2.1 Java**

For the implementation of our patient orientation telemedicine task prototype, we have used Java Object Oriented Programming language. Indeed, thanks to its portability and independence of platform, Java enables applications to be executed independently of the hardware and the operating systems on which these applications are installed. In addition, many libraries and ontology development APIs are based on Java which makes it suitable for applications using ontologies.

Regarding the development environment, we used the NetBeans IDE which can potentially create development projects implementing different web technologies such as HTML, JSP, Servlet, Javascript, etc. Additionally, web servers can be easily integrated in the NetBeans IDE, such as the Apache TomCat server or the Glassfish server.

---

#### **VI.1.2.2 Jena API**

The APIs of Semantic Web Languages constitute libraries which permit to create and manage ontologies represented in these languages. Several APIs

have been developed with respect to the OWL language such as Jena (Jena 2005) and OWL API (OWL API 2007). Using these APIs enables developers to overcome the programming difficulties related to syntactic analysis (parsing) and writing specific syntaxes (Serialisation).

Jena, developed by Hewlett-Packard (HP), is an API based on RDF. In other words, its internal representation model conforms to the RDF graph representation. Thus, the formalisms of the ontologies that can be handled by this API are those built on top of RDF, such as OWL and its three sublanguages. Jena offers statement-centric (based on the subject-predicate-object structure) support for manipulating RDF and OWL data including typed literals, RDFS and OWL specific constructs. It comes with a built-in RDF query language, SPARQL. Jena provides a programmatic environment for RDF, RDFS, OWL, and SPARQL. Additionally, it includes a rule-based inference engine and has object classes to represent graphs, resources, properties and literals. The interfaces representing resources, properties and literals are called Resource, Property and Literal respectively. In Jena, a graph is called a model and represented by the Model interface.

---

## VI.2 ONOF-PAS Ontologies Editing

Figure 36 is a snapshot of the ONOF-PAS global ontology classes edited by Protégé. Additionally, Figure 37 shows an extract of the list of object type properties which represent the interrelationships among the ONOF-PAS concepts. Figure 38 shows an extract of the list of data type properties representing the attributes of each concept. Table 8 clarifies the domain and range of each object type property while Table 9 details the data type properties related to the main concepts in ONOF-PAS.

These ontologies have been instantiated in the telemedicine domain as we have seen in chapter 4 where we have described the instantiation of different ONOF-PAS ontologies.



Figure 36. Extract of the ONOF-PAS Ontologies Classes, edited by Protégé.

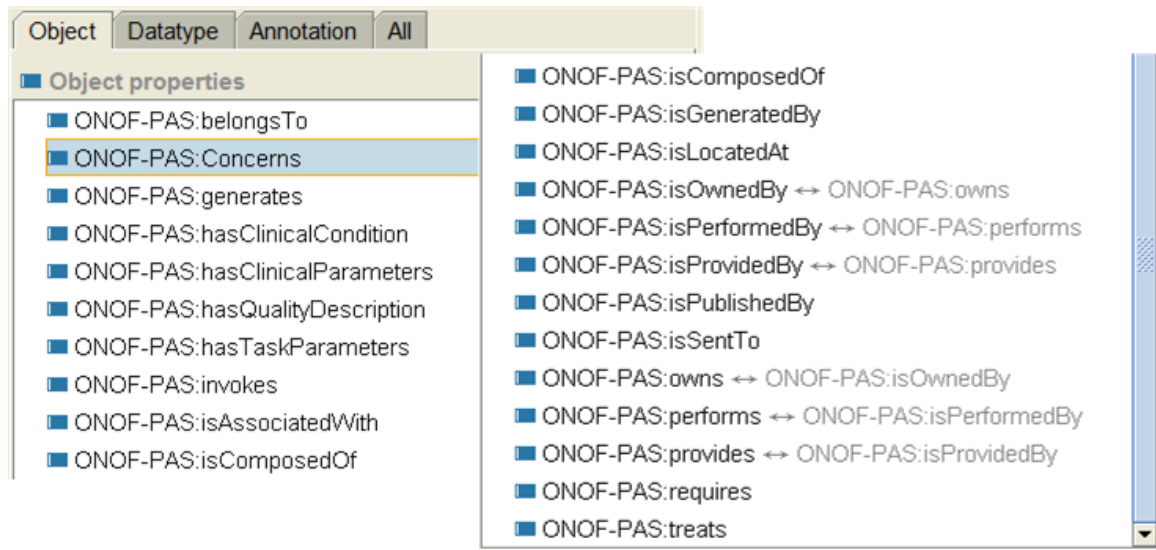


Figure 37. Extract of the Object Type Properties of the ONOF-PAS Global Ontology

Table 8. Object Type Properties Domain and Range

| Property                     | Domain             | Range                  |
|------------------------------|--------------------|------------------------|
| <b>belongsTo</b>             | Actor              | Organization           |
| <b>concerns</b>              | Task               | Patient                |
| <b>generates</b>             | Task               | Message                |
| <b>hasClinicalConditions</b> | Patient            | Clinical_Condition     |
| <b>hasClinicalParameters</b> | Clinical_Condition | Clinical_Parameter     |
| <b>hasQualityDescription</b> | Service            | QoS                    |
| <b>hasTaskParameters</b>     | Task               | Task_Parameter         |
| <b>Invokes</b>               | Task               | Service                |
| <b>isAssociatedWith</b>      | Actor              | Communication_Resource |
| <b>isComposedOf</b>          | Process            | Task                   |
| <b>generatedBy</b>           | Message            | Task                   |
| <b>isLocatedAt</b>           | Actor              | Location               |
| <b>isLocatedAt</b>           | Organization       | Location               |
| <b>isOwnedBy</b>             | Resource           | Organization           |
| <b>isPerformedBy</b>         | Task               | Actor                  |
| <b>isProvidedBy</b>          | Service            | Organization           |
| <b>isPublishedBy</b>         | Message            | Actor                  |
| <b>isSentTo</b>              | Message            | Actor                  |
| <b>isSentTo</b>              | Message            | Organization           |
| <b>owns</b>                  | Organization       | Resource               |
| <b>performs</b>              | Actor              | Task                   |
| <b>provides</b>              | Organization       | Service                |
| <b>requires</b>              | Task               | Material_Resource      |
| <b>Treats</b>                | Task               | Object                 |

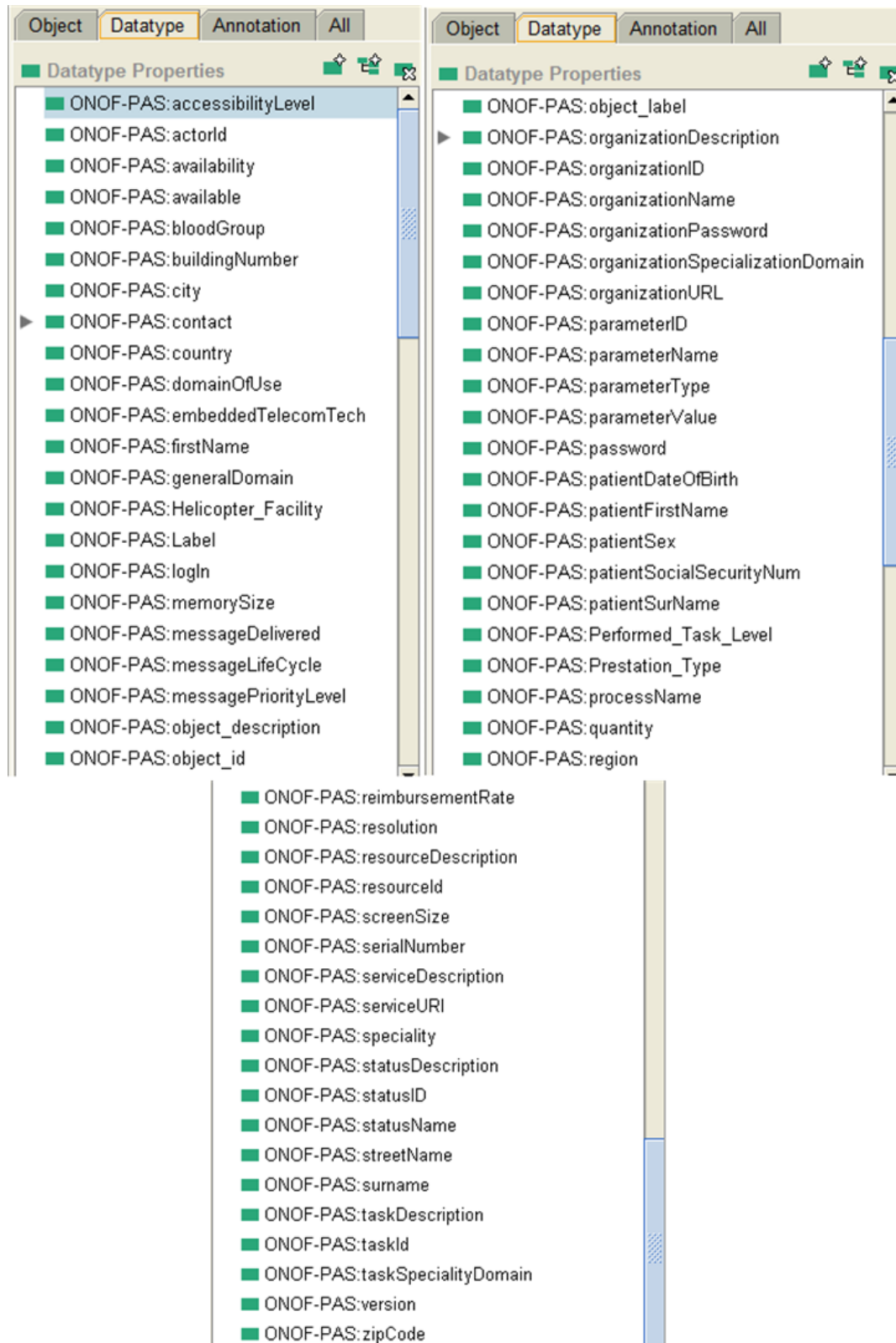


Figure 38. Extract of the Data Type Properties of the ONOF-PAS Global Ontology

Table 9. Data Type Properties, Domain and “Range”

| Domain Class                   | Data Type Property               | “Range”                           |
|--------------------------------|----------------------------------|-----------------------------------|
| <b>Telemedicine Task</b>       | taskId                           | int                               |
|                                | taskDescription                  | string                            |
|                                | accessibilityLevel               | Level_A, Level_B, Level_C         |
|                                | taskSpecialityDomain             | Cardiology, Cancer, ...           |
| <b>Healthcare Professional</b> | actorId                          | int                               |
|                                | firstName                        | string                            |
|                                | surName                          | string                            |
|                                | login                            | string                            |
|                                | password                         | string                            |
|                                | Performed_Task_Level             | Level_A, Level_B, Level_C         |
|                                | available                        | boolean                           |
|                                | generalDomain                    | Cardiology, Cancer, Diabetes, ... |
|                                | speciality                       | Cardiologist, Surgeon, ...        |
| <b>Material Resource</b>       | resourceId                       | int                               |
|                                | resourceDescription              | string                            |
|                                | label                            | string                            |
|                                | quantity                         | int                               |
|                                | availability                     | boolean                           |
|                                | domainOfUse                      | Cardiology, Cancer, ...           |
| <b>Communicaton Resource</b>   | resourceId                       | int                               |
|                                | resourceDescription              | string                            |
|                                | label                            | string                            |
|                                | memorySize                       | int                               |
|                                | embeddedTelecommTech             | GPRS, EDGE, HSDPA, ...            |
| <b>Service</b>                 | serviceURI                       | string                            |
|                                | serviceDescription               | string                            |
| <b>Patient</b>                 | firstName                        | string                            |
|                                | surName                          | string                            |
|                                | dateOfBirth                      | date                              |
|                                | sex                              | Male, Female                      |
|                                | socialSecurityNumber             | int                               |
| <b>Hospital</b>                | organizationId                   | Int                               |
|                                | organizationURL                  | string                            |
|                                | organizationName                 | string                            |
|                                | organizationSpecializationDomain | string                            |
|                                | status                           | Public, Private                   |
|                                | helicopterFacility               | boolean                           |
| <b>Location</b>                | buildingNumber                   | Int                               |
|                                | streetName                       | string                            |
|                                | zipCode                          | int                               |
|                                | city                             | Lyon, Grenoble, ...               |
|                                | region                           | RhôneAlpes, IleDeFrance, ...      |
|                                | country                          | France, ...                       |
| <b>Message</b>                 | messagePriorityLevel             | High, Medium, Low                 |
|                                | messageDelivered                 | boolean                           |
|                                | messageLifeCycle                 | T1, T2, T3                        |
| <b>Clinical Condition</b>      | statusId                         | Int                               |
|                                | statusName                       | string                            |
|                                | statusDescription                | string                            |



---

### VI.3 Task Execution Example Using Protégé

In this section, we describe an example of a telemedicine task execution using the Protégé environment. We demonstrate how the heterogeneous resources required for performing the task can be automatically inferred based on a set of SWRL rules that link the following elements: 1) the telemedicine task; 2) the patient concerned by the task; 3) his clinical conditions. We also demonstrate the rule-based reasoning process using the inference engine JESS to infer the required resources and we will illustrate how organizations that own or manage the required resources can be retrieved using a simple SPARQL query.

---

#### VI.3.1 Patient Orientation Task Scenario

Let us retake the scenario of a patient transfer decision making, presented in the introduction of this thesis. We suppose that Mr. John Smith has a cardiac problem while skiing or staying in high mountains resorts. The first aid person or the rescue team member should take a decision to orient the patient to a hospital that has all logistical and material resources needed for the patient's treatment. We assume that the hospital which will receive Mr. John Smith must have an available place in the intensive care unit, coronarography equipments in order to perform a Stenting procedure, and a surgery unit equipped with devices needed for a cardiac surgery as well as the appropriate medical staff, e.g. cardiologist, anaesthesiologist, surgeon, etc, who are available and capable to perform an eventual intervention.

The objective of ONOF-PAS in such a scenario is to infer the aforementioned heterogeneous resources and services required for performing the patient orientation task. In addition to the different material resources, ONOF-PAS should also infer other location-based services to be invoked by the running task. These services could be, for instance, location-based services to calculate the distance and the time to reach the selected destination (hospital), road traffic status services, forecast services, etc. Additionally, the patient's EHR access service could be also inferred to provide the rescue team members with additional information related to the patient's medical history, risk factors, etc.

In this way, the first aid person providing the medical assistance to the patient, who may be not fairly skilled, can take the appropriate decision for patient orientation or transfer in such an hostile environment.

### VI.3.2 Task Description

Firstly, we create an instance of the Task ontology, “Patient Orientation”. Figure 39 displays the different object type properties and data type properties related to the Patient Orientation instance. As we can see, this task has an identifier and it is classified under the speciality domain “Emergency”. The task is performed by a rescue team member, whose instance is described by Figure 40. In addition, the patient orientation task concerns Patient Mr. John Smith which has a clinical status “chest pain” (Figure 41). The clinical status “Chest Pain” has, for example, as clinical parameters: “Systolic Blood Pressure”, “Diastolic Blood Pressure”, “heart rate”, and “oxygen saturation level”. On the other side, the “Patient Orientation” task has parameters such as accident date and time.

As we can notice in Figure 39, the different resources required by the patient orientation task have not yet been associated to the task instance. These resources will be inferred automatically by the JESS inference engine based on the rules described in SWRL.

We have edited the SWRL rules using SWRLTab integrated in the Protégé environment as depicted in Figure 42. Figure 43 displays three rules that concern the patient orientation task, i.e. rules 10, 13, and 14. These rules will be applied by the JESS engine to infer the different resources, i.e. devices and services, needed by the patient orientation task, and additionally, to assign a priority level to the messages generated by the task.

The execution of rule 13 will have the effect of setting the property “*Requires*” to the patient orientation task. More specifically, if we have a Telemedicine Task “*?ONOF-PAS:tas*” which has a concerned patient “*?ONPF-PAS:p*”, and if this patient has as clinical status chest pain “*?ONOF-PAS:cp*”, then the inference engine will infer the decision that the running task needs as resources an available place in an intensive care unit, a coronarography equipment, and a surgery unit.

For Individual: [http://www.owl-ontologies.com/OntologyTelemedicineKnowledgeModel.owl#Patient\\_Orientation](http://www.owl-ontologies.com/OntologyTelemedicineKnowledgeModel.owl#Patient_Orientation)

| Property     | Value  | Lang |
|--------------|--|------|
| rdfs:comment | This telemedicine task can be performed by a healthcare professional in order to orient patient in case of emergency |      |

ONOF-PAS:taskId

| Value  | Type |
|--------|------|
| 200066 | int  |

ONOF-PAS:taskSpecialityDomain

| Value     | Type   |
|-----------|--------|
| Emergency | string |

ONOF-PAS:isPerformedBy

- ONOF-PAS:Rescue\_Team\_Member\_5

ONOF-PAS:generates

- ONOF-PAS:Message\_4

ONOF-PAS:invokes

ONOF-PAS:taskDescription

| Value  | Lang |
|--|------|
| Patient Orientation Task can be performed by a healthcare a professional in order to transfer the patient to the most appropriate hospital |      |

ONOF-PAS:hasTaskParameters

- ONOF-PAS:Accident\_Date\_8
- ONOF-PAS:Accident\_Time\_9

ONOF-PAS:Concerns

- ONOF-PAS:Mr\_John\_Smith

ONOF-PAS:accessibilityLevel

| Value   | Type   |
|---------|--------|
| Level_C | string |

ONOF-PAS:requires

ONOF-PAS:treats

Figure 39. Patient Orientation Task Instance

ONOF-PAS:Rescue\_Team\_Member\_5 (instance of ONOF-PAS:Rescue\_Team\_Member, internal name is http://...)

INDIVIDUAL EDITOR for ONOF-PAS:Rescue\_Team\_Member\_5 (instance of ONOF-PAS:Rescue\_Team\_Member)

For Individual: [http://www.owl-ontologies.com/OntologyTelemedicineKnowledgeModel.owl#Rescue\\_Team\\_Member\\_5](http://www.owl-ontologies.com/OntologyTelemedicineKnowledgeModel.owl#Rescue_Team_Member_5)

| Property     | Value | Lang |
|--------------|-------|------|
| rdfs:comment |       |      |

ONOF-PAS:actorId

| Value | Type |
|-------|------|
| 10089 | int  |

ONOF-PAS:firstName

| Value              | Lang |
|--------------------|------|
| actor's first name |      |

ONOF-PAS:surname

| Value             | Lang |
|-------------------|------|
| actor's last name |      |

ONOF-PAS:login

| Value         | Lang |
|---------------|------|
| actor's login |      |

ONOF-PAS:password

| Value            | Lang |
|------------------|------|
| actor's password |      |

ONOF-PAS:generalDomain

| Value      | Type   |
|------------|--------|
| Cardiology | string |

ONOF-PAS:available

true

ONOF-PAS:speciality

| Value      | Type   |
|------------|--------|
| Cardiology | string |

ONOF-PAS:belongsTo

- ONOF-PAS:Grenoble\_CHU

ONOF-PAS:isAssociatedV

- ONOF-PAS:SmartPhone\_30

ONOF-PAS:isLocatedAt

- ONOF-PAS:Location\_10

ONOF-PAS:performs

- ONOF-PAS:Patient\_Orientation
- ONOF-PAS:Patient\_Transfer\_1

ONOF-PAS:Performed\_Task\_Level

| Value   | Type   |
|---------|--------|
| Level_C | string |

Figure 40. Rescue Team Member Instance

Figure 41. Extract of a Patient Instance

| Enabled                             | Name                 | Expression  |
|-------------------------------------|----------------------|---|
| <input type="checkbox"/>            | ONOF-PAS:Rule-1      | ONOF-PAS:Emergency_Physician(?ONOF-PAS:ep) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:perform(?ONOF-PAS:per)   |
| <input checked="" type="checkbox"/> | ONOF-PAS:Rule-10     | ONOF-PAS:Rescue_Team_Member(?ONOF-PAS:rtm) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:perform(?ONOF-PAS:rtm, ?ONOF-PAS:tas)  |
| <input type="checkbox"/>            | ONOF-PAS:Rule-12     | ONOF-PAS:Nurse(?ONOF-PAS:n) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:perform(?ONOF-PAS:n, ?ONOF-PAS:tas)   |
| <input type="checkbox"/>            | ONOF-PAS:Rule-2      | ONOF-PAS:Specialist(?ONOF-PAS:car) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:perform(?ONOF-PAS:car, ?ONOF-PAS:tas)  |
| <input type="checkbox"/>            | ONOF-PAS:Rule-3      | ONOF-PAS:General_Practitioner(?ONOF-PAS:gp) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:perform(?ONOF-PAS:gp, ?ONOF-PAS:tas)  |
| <input type="checkbox"/>            | ONOF-PAS:Rule-4      | ONOF-PAS:Actor(?ONOF-PAS:act) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:perform(?ONOF-PAS:act, ?ONOF-PAS:tas)   |
| <input type="checkbox"/>            | ONOF-PAS:Rule-7      | ONOF-PAS:Healthcare_Professional(?ONOF-PAS:hcp) ^ ONOF-PAS:Organization(?ONOF-PAS:org) ^ ONOF-PAS:Location(?ONOF-PAS:loc) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:perform(?ONOF-PAS:hcp, ?ONOF-PAS:tas) |
| <input type="checkbox"/>            | ONOF-PAS:Rule-8      | ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:Organization(?ONOF-PAS:org) ^ ONOF-PAS:Location(?ONOF-PAS:loc) ^ ONOF-PAS:perform(?ONOF-PAS:tas, ?ONOF-PAS:mes)   |
| <input type="checkbox"/>            | ONOF-PAS:Rule-9      | ONOF-PAS:Healthcare_Professional(?ONOF-PAS:hcp) ^ ONOF-PAS:Organization(?ONOF-PAS:org) ^ ONOF-PAS:Location(?ONOF-PAS:loc) ^ ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Message(?ONOF-PAS:mes) ^ ONOF-PAS:perform(?ONOF-PAS:hcp, ?ONOF-PAS:tas) |
| <input checked="" type="checkbox"/> | Telemedicine:Rule-13 | ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Patient(?ONOF-PAS:p) ^ ONOF-PAS:Chest_Pain(?ONOF-PAS:cp) ^ ONOF-PAS:Concurrence(?ONOF-PAS:con)  |
| <input checked="" type="checkbox"/> | Telemedicine:Rule-14 | ONOF-PAS:Task(?ONOF-PAS:tas) ^ ONOF-PAS:Patient(?ONOF-PAS:p) ^ ONOF-PAS:Chest_Pain(?ONOF-PAS:cp) ^ ONOF-PAS:Concurrence(?ONOF-PAS:con)  |

Figure 42. Extract of the List of SWRL Rules edited using SWRLTab in Protégé

The execution of rule 14, Figure 43, will have the effect of setting the property “*invoke*” to the patient orientation task. In other words, by applying this rule, the inference engine determines the services that should be invoked by the running task.

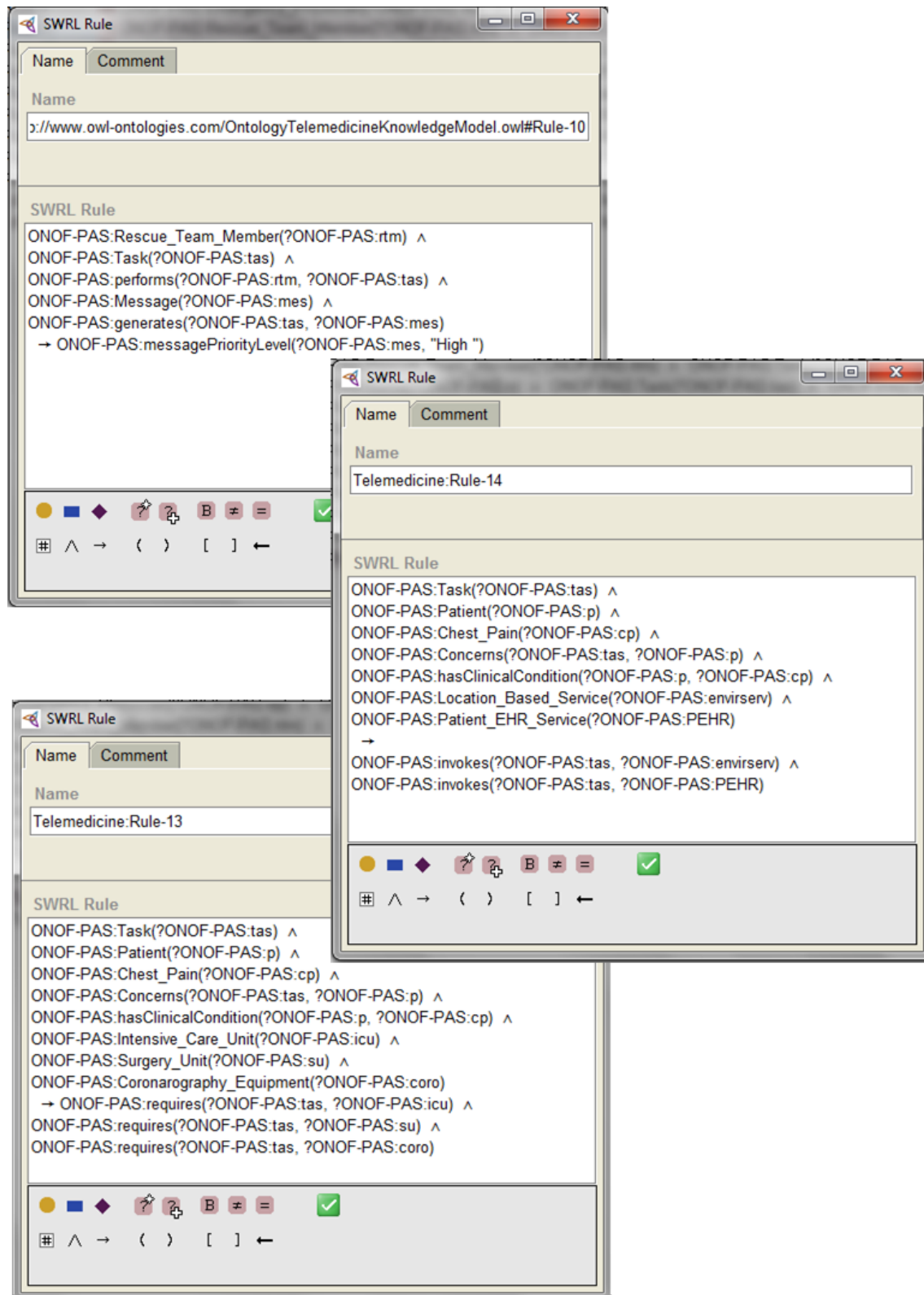


Figure 43. Extract of SWRL Rules concerning the Patient Orientation Task

In order to assign a priority level to the message generated by the task, the inference engine applies rule 10 describing the following context: if a rescue team member performs a task, and this task generates a message, then the message priority level will be “*high*”.

As we have stated earlier, JESS interprets and executes rules through three principal steps as depicted by Figure 44 showing the JESS Plug-in in Protégé.

In the first step, JESS converts OWL classes to JESS facts and transforms SWRL rules to JESS rules, as it can be seen in Figure 45. The summary of this step includes the number of SWRL rules exported to JESS, as well as the number of OWL classes, OWL individuals and of OWL axioms exported to JESS. This summary is detailed in the bottom panel Figure 44. As it can be seen, the elements which have been exported to JESS are as follows: 3 SWRL rules, 89 OWL classes, 31 OWL individuals, and 29 OWL axioms.

In the next step, JESS verifies the consistency of all the ontologies and then executes the reasoning based on the exported elements, i.e. JESS facts and JESS rules. Figure 45 displays the number of the inferred axioms, which is 41 in this particular example. These axioms constitute the relations between the running patient orientation task and the heterogeneous resources required for performing the task, as well as the relations between the running task and the services that should be invoked by it.

In the final step, JESS converts the JESS inferred knowledge to knowledge described in OWL.

Figure 46 shows the patient orientation task instance after the inference of different resources and services required for performing the task. As it can be seen in the inferred results of Figure 46, the inferred axioms, surrounded by blue lines, represent the multiple heterogeneous resources needed by the task. The requested material resources such as a place in the intensive care unit, a surgery unit, and a coronarography equipment are available at almost one of the hospitals where the patient might be transported. The left panel of Figure 46 displays the instances of location-based services such as *distance to arrive*, *time to arrive*, *road traffic status*, and *forecast* service, that might be invoked by the running task. An additional service, the access to the patient’s EHR, might also be

inferred to provide the healthcare professional with supplementary information related to the patient's medical history, risk factor, etc.

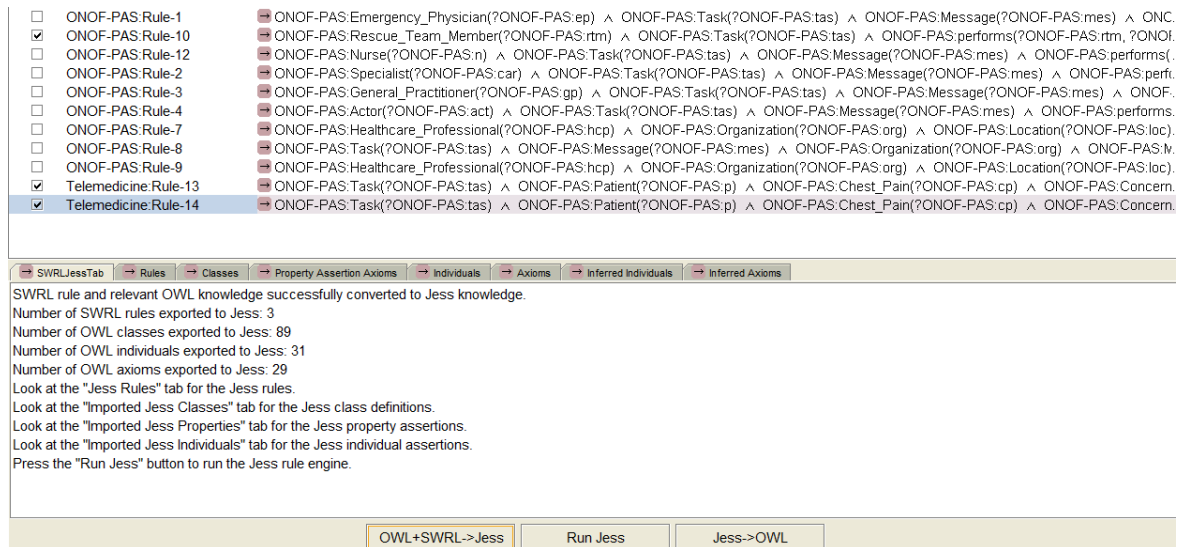


Figure 44. Reasoning with the SWRLJESSTab Protégé Plug-in

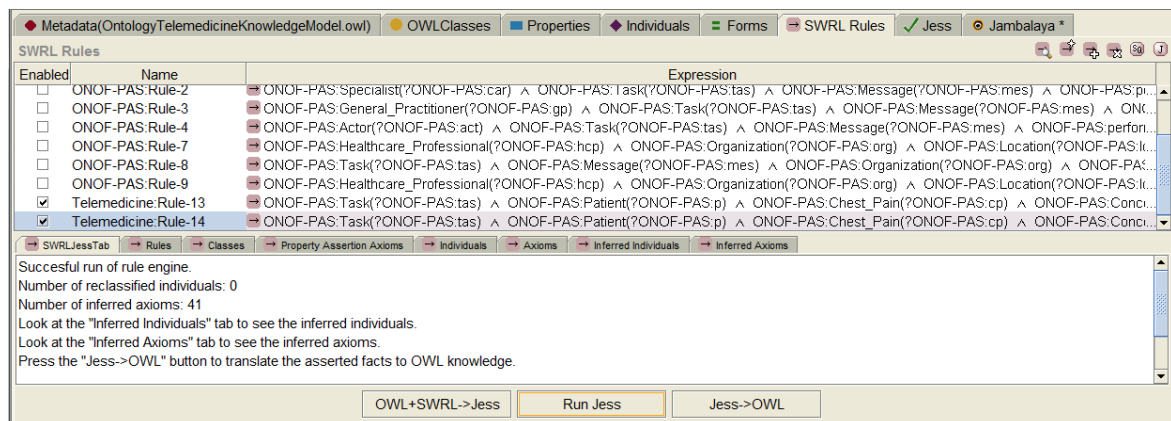


Figure 45. Number of Axioms inferred by JESS

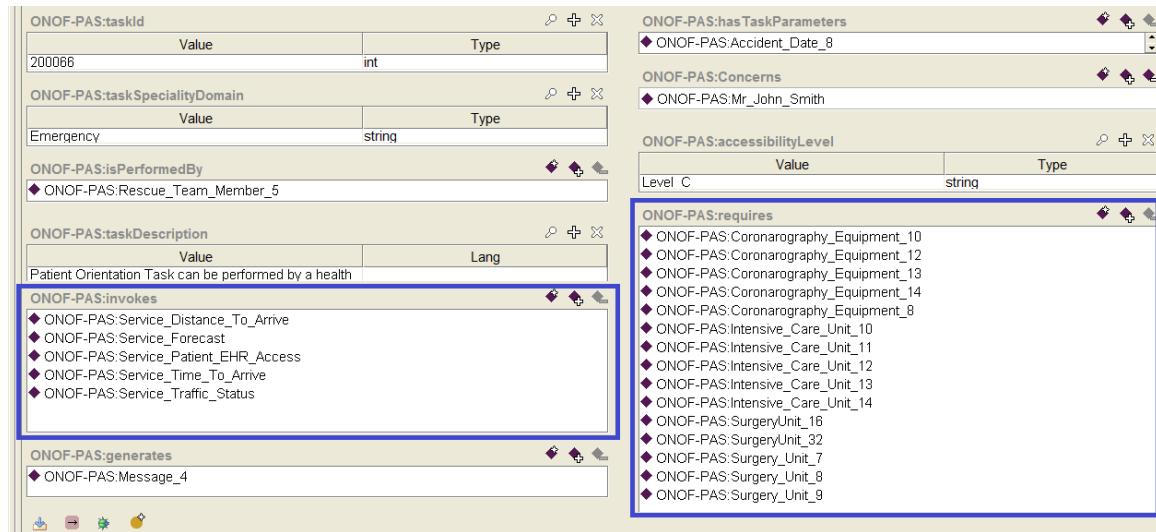


Figure 46. Snapshot of the Patient Orientation Task Instance after Resources Inferring

So far, the different axioms that describe the relations between the patient orientation task instance and the required resources, have been inferred. In the next sections, we provide SPARQL queries examples to retrieve data of the inferred resources.

### VI.3.3 Resources Data Filtering Using the SPARQL Language

In this section we demonstrate how the inferred resources data can be retrieved and filtered using the standard query language SPARQL and the SPARQL query panel integrated in the Protégé environment. In the following, we provide four queries examples:

- Query 1: Select the material resources “*?MAT\_RES*” required by the task patient orientation “*?TASK*”. Figure 47 shows query 1 and its results.
- Query 2: we can retrieve additional data such as the organizations “*?ORG*” who own or manage the required resources, as well as the organizations locations “*?LOC*” (Figure 48). Here, “*?LOC*” represents a variable of the class “*Location*” which has as data type properties “*region*”, “*city*”, “*zip-code*”, etc, see Table 9.
- Query 3: in this example, we use « *FILTER* » in the SPARQL query in order to get just the resources managed by organizations, and/or Hospitals located in the *Rhone-Alpes* region and, particularly, in the *Lyon*



city (Figure 49).

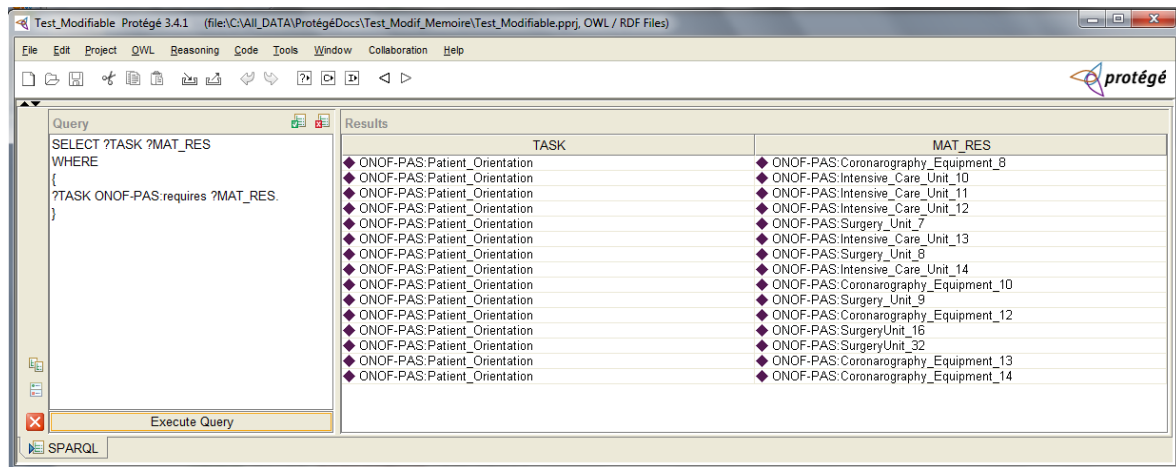


Figure 47. SPARQL Query Example 1

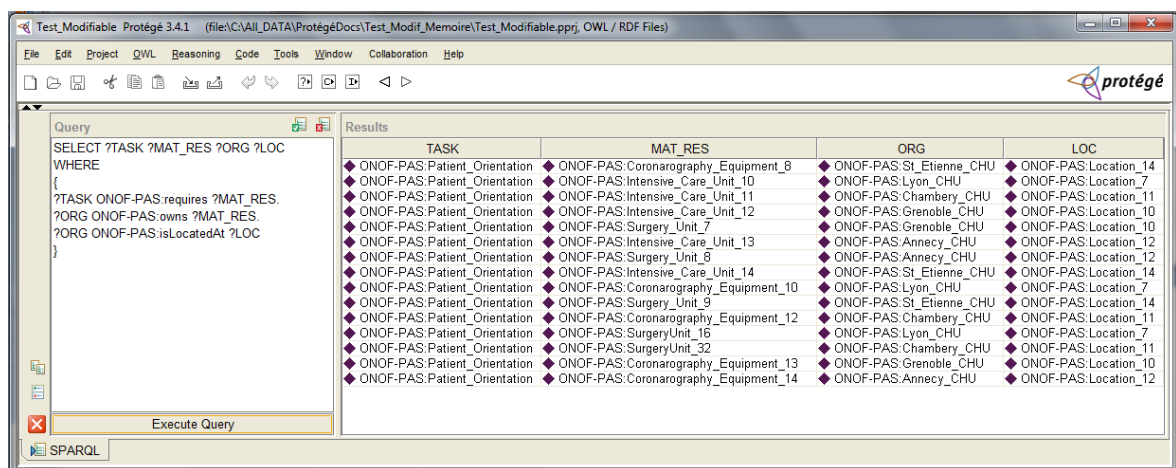


Figure 48. SPARQL Query Example 2

- Query 4: in query 4, we retrieve the patient orientation task, the resources required by the task, and the hospitals which own these resources. We have used the modifier FILTER in the SPARQL query to retrieve just the available resources owned or managed by hospitals located in the *Rhone Alpes* French region. Additionally, we have also added the modifier OPTIONAL to retrieve additional information about some logistic aspects related to the available hospitals, e.g. the possibility of providing a helicopter to transport the patient (Figure 50).

According to the task type, different SPARQL queries can also be formulated to retrieve resources data that meet user specific preferences in terms of technical characteristics or any other specifications of a given resource.

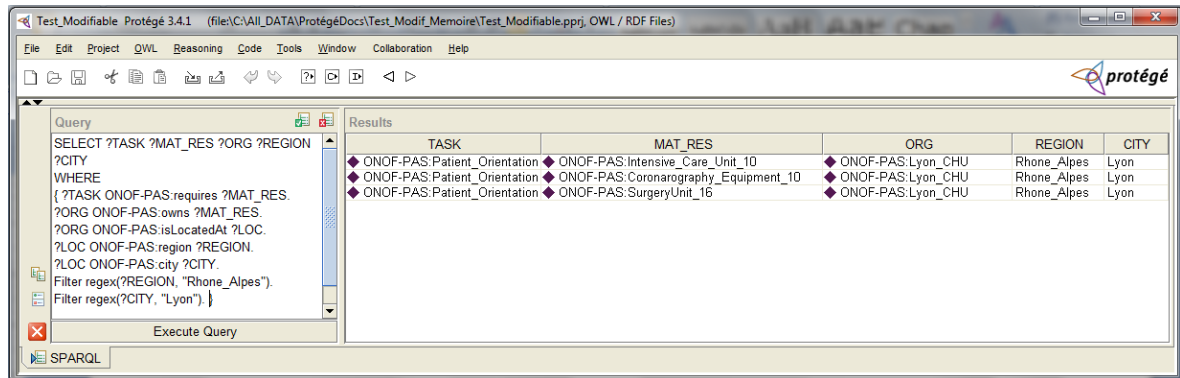


Figure 49. SPARQL Query Example 3

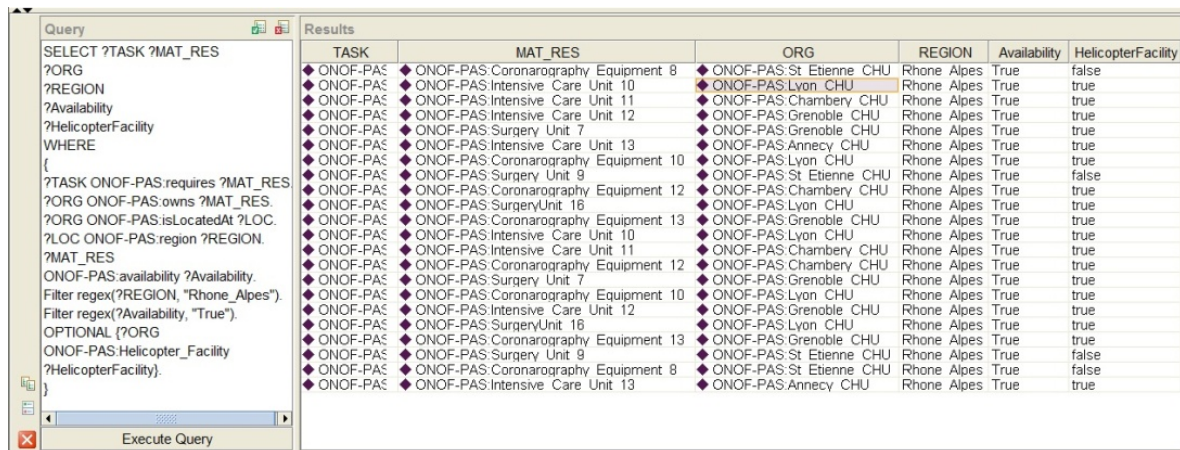


Figure 50. SPARQL Query Example 4

## VI.4 TTROIE Prototype

In this section, we present our prototype called TTROIE, standing for “Telemedicine Tasks and Resources Ontologies based system for Inimical Environment”. The main functionality of TTROIE is to generate the heterogeneous resources needed for the tasks’ execution, taking into account their conditions in terms of availability and capabilities. The

objective is to support the decision making processes in which healthcare professionals are involved.

TTROIE presently implements the task *Patient Transfer* concerning the scenario we have earlier presented. But, additional telemedicine tasks such as *Ask for Second Opinion*, and *Patient EHR Access* can be realized based on the TTROIE architecture. In the following subsections, we present, using the UML formalism, the different possible use cases, the main class diagram, the software architecture, and the main GUI of TTROIE.

---

#### VI.4.1 Different TTROIE Use Cases

Telemedicine information system requirements vary according to the healthcare professional and/or healthcare institute profiles. Actually, each type of healthcare worker, or healthcare institute, has specific needs depending on domain activities. These heterogeneous needs can be classified into three main categories, i.e. administrative management needs, medical and technical management needs, communication needs. Starting from the user needs and the telemedicine system requirements, several use cases can be described according to the application scenarios.

Figure 51 shows the main use cases that can be identified in difficult telemedicine environments such as the high mountain resorts. These use cases include telemedicine task performing, profile management and messages management. In the TTROIE prototype we have implemented the use case patient orientation which is an instance of the Tele-assistance use case, i.e. a Telemedicine Task. We have identified two types of actors. A healthcare professional actor type, i.e. Nurse, Physician, General Practitioner, Specialist, etc, and an organization actor type, i.e. hospital, private clinic, medical center, healthcare institute, etc.

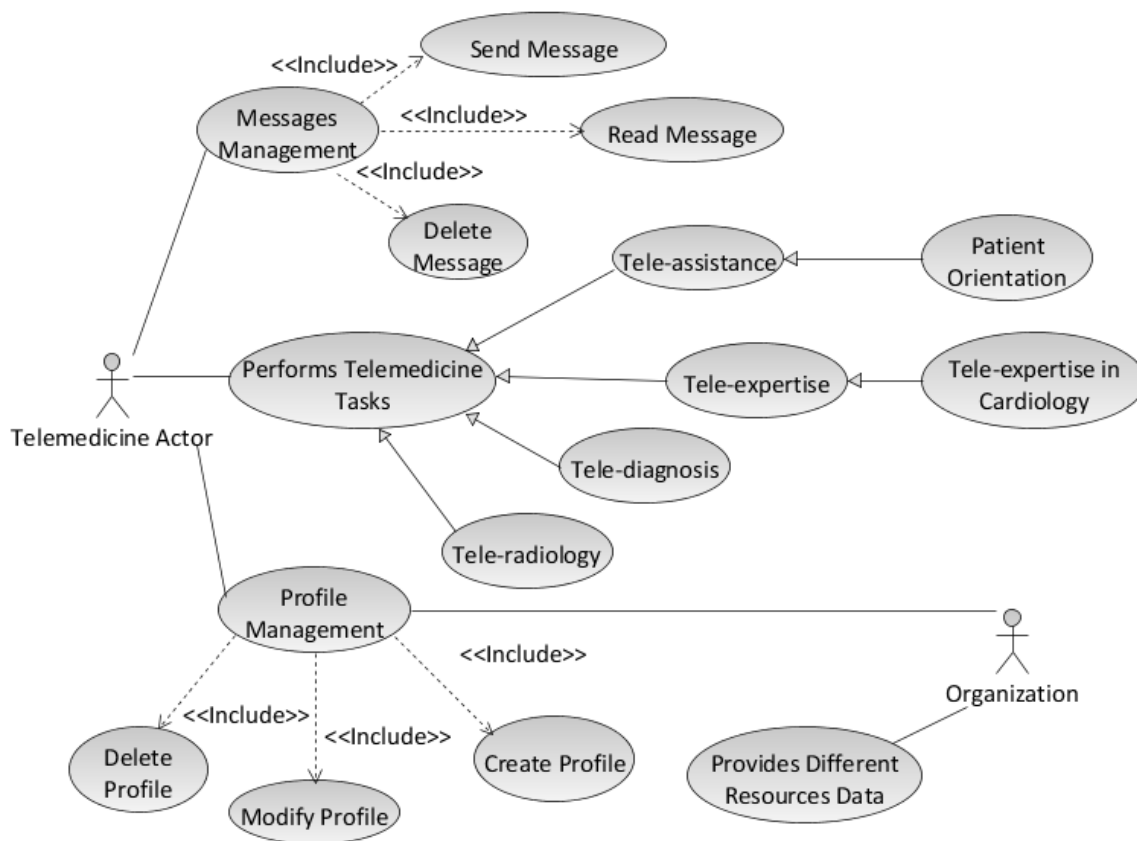


Figure 51. Different use cases of TTROIE

#### VI.4.2 TTROIE Software Architecture

We have organized the TTROIE components in three modules: the user interface module, the task management module, and the knowledge base module. The latter includes ontology classes and the Jena API. Figure 53 displays the overall TTROIE software architecture. The task management layer corresponds to the business logic layer in a three-tier architecture.

In chapters IV and V, we have already provided a full description of the knowledge base, of the task manager and of different other components, i.e. the resources updater, the resources filter, etc.

To develop the TTROIE prototype, the first step was to create an object oriented classes model to represent each concept of the ONOF-PAS ontologies. We have thus created numerous classes that have been designed to represent the main entities participating in the patient orientation task realization (Figure 52). More specifically, these object oriented classes

represent concepts such as task, actor, organization, resource, patient, location, etc. This step also comprises the definition of different variables as well as the “set” and “get” methods, *setters* and *getters*.

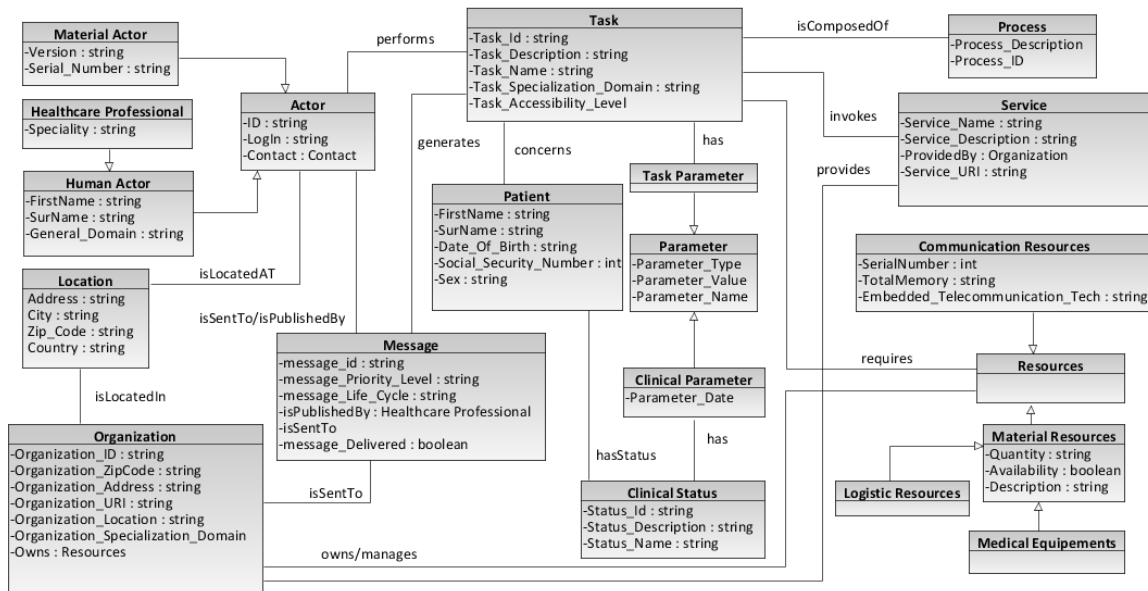


Figure 52. TTROIE main classes

The different objects instantiated from these classes will be inserted and stored as instances in the different ontologies of ONOF-PAS. For this, we have used the Jena framework which provides several packages to handle an ontology. For example, we have used the package “*com.hp.hpl.jena.ontology*” to define concepts, object type properties and data type properties. Additionally, we have used the packages “*com.hp.hpl.jena.reasoner*” and “*com.hp.hpl.jena.query*” to transform SWRL into Jena rules and to perform rule-based reasoning.

To simply use the different functionalities that we wanted to implement, it was also necessary to create a general class executing all the transactions concerning the handling of the OWL files. The creation of such a class is an essential step because it represents a cornerstone of the task management module. We have called this class “*OwlManager*”. It contains all the access and insertion methods as well as other specific methods concerning the launching of the inference engine. An extract of the *OwlManager* class Java source code is provided in appendix B.

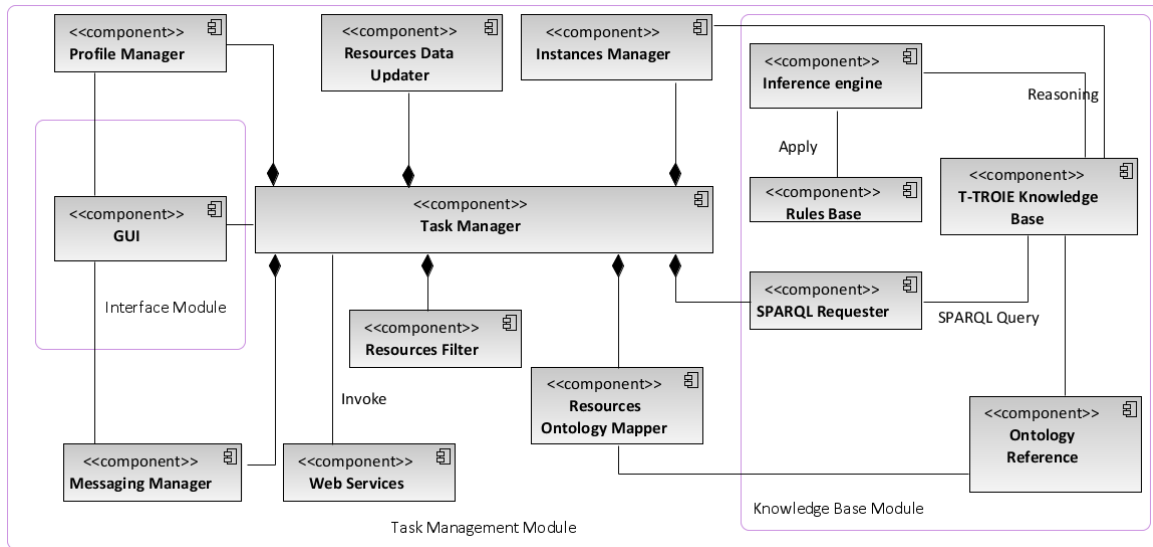


Figure 53. TTROIE Components Diagram

One of the most important functionalities that we had to implement is the inference of the resources data required by the task. Indeed, although the inference engine is integrated as standard in the Jena API, its use requires a series of operations in order to correctly generate the inferred axioms. To use the inference engine, we need first to define a model which contains all the ontology information axioms modeled as a graph. Then we shall instantiate the inference model, which includes a rules file, and after that launch the inference engine integrated in Jena by invoking the method “*createInfModel()*”.

The rules have already been written in an OWL document according to the SWRL format. However, as the Jena API does not understand this language, it is necessary to transform the SWRL rules into Jena rules. We had thus to redefine, in an annex file, the SWRL rules in a language that is interpretable by Jena in order to run the inference engine. This phase is rather complicated because of differences in data representation.

The last step is to create the interface module which contains several classes that allow user/system interaction through a friendly graphical interface as we will see in section VI.4.5.

### VI.4.3 Task Activities Diagram

Figure 54 shows the activities diagram concerning the creation of the user profiles and the execution of the tasks. The schema is a graphical representation of the workflows of the stepwise activities and actions executed in order to reach the final objective of the task.

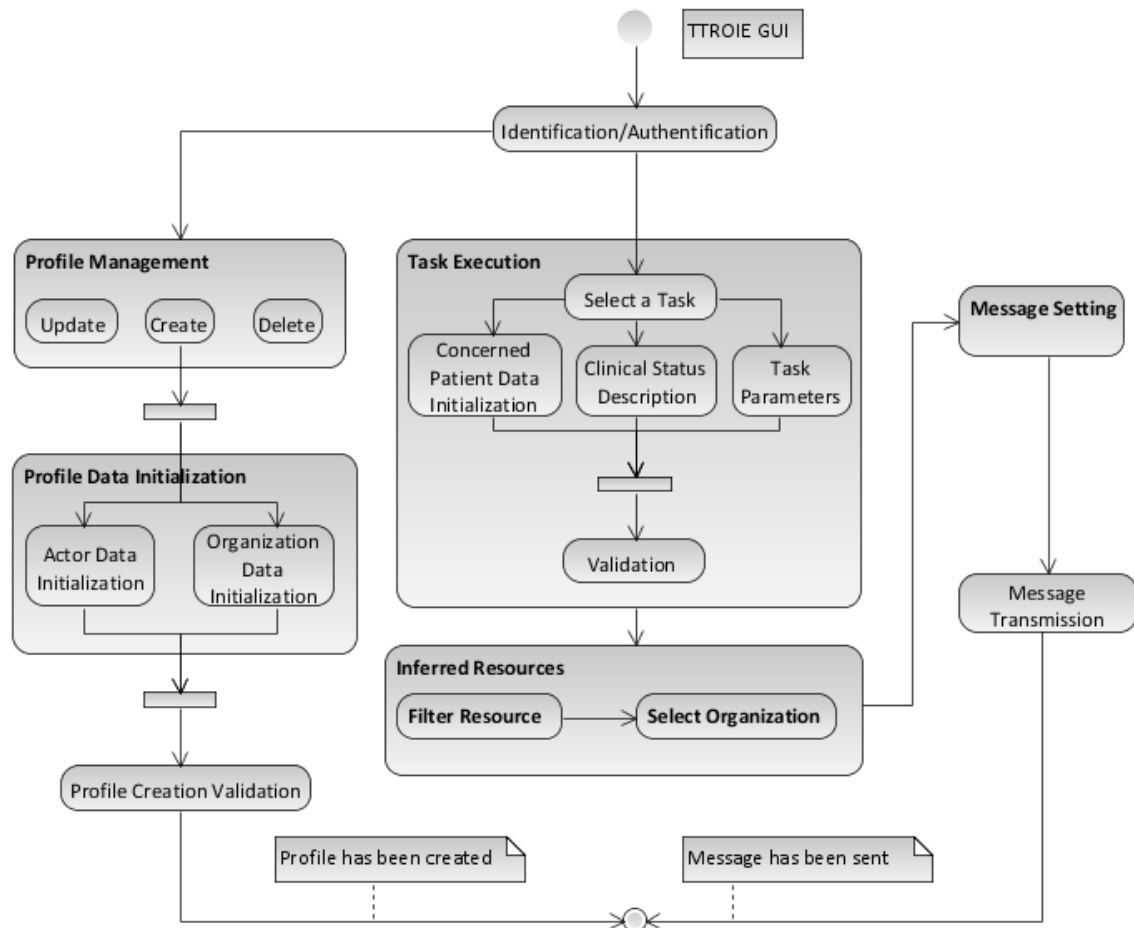


Figure 54. Telemedicine Task Activities Diagram

The identification and authentication are a common activity in all scenarios. It is indeed necessary to identify the user profile in order to check the user eligibility to perform the selected task.

Concerning the patient orientation task execution, several activities such as the initialization of the patient data, the description of the clinical status and of the task parameters, have been defined. Once the user

has validated the data input, other actions can be executed such as the resources inferring. In the next section we describe the task execution algorithm and then we detail the interaction between the user and the TTROIE system through the GUI.

#### VI.4.4 Telemedicine Task Execution Algorithm

In this section, we present the telemedicine task execution algorithm composed of multiple instructions to be executed (see Table 10).

Table 10. Task Execution Instructions

| Algorithm Inputs : instances of actor, task, patient, clinical conditions, ...  |
|---|
| 01: Begin<br>02: For each actor, task, object, and object's status<br>03: Create instance<br>04: Insert instances into the relevant ontologies<br>05: End for<br>06: Get actor location<br>07: Get the context of the objects concerned by the task (patient clinical conditions)<br>08: Call reasoning<br>09: SPARQL Query (select the list of resources required by the running task)<br>10: For each resource instance in the resources list<br>11: get resource availability<br>12: get resource description<br>13: get resource organization manager<br>14: End for<br>15: SPARQL Query (select the list of organizations managing the required resources)<br>16: For each organization<br>17: If organization location = location selected by actor<br>18: Select organization<br>19: Return a list of organizations<br>20: End for<br>21: Generate the task messages<br>22: Send messages to the concerned actors and organizations<br>23: End |
| Algorithm Output : list of instances of the resources required to achieve the task objectives and list of the organizations who manage these resources, and the resources availability and capabilities   |

The algorithm inputs are an instance of the Actor, i.e. a rescue team member, an instance of Task, i.e. the patient orientation task, an instance of the object treated by the task which is in our scenario example patient Mr. John Smith, and instances of the patient's clinical conditions, such as heart attack, chest pain, high blood pressure, etc. Additional contextual data concerning the running task and the user profile may be entered as algorithm input data, e.g. data about the technical characteristics of the communication device used by the actor, the actor's location, and other environmental data, etc. The algorithm's outputs are a list of heterogeneous resources which are necessary for achieving the task



objective, as well as the list of organizations which have these resources available.

#### VI.4.5 TTROIE User Interface

Figure 55 displays the main interface of the TTROIE prototype. The upper right panel highlights the layout of the patient transfer tab. The latter will allow the users to enter the data required for the patient transfer task execution, i.e. patient data, such as the first name, last name, sex, social security number, etc, and the clinical conditions. The interface also includes some buttons, located in the left panel, providing access to the main functionalities of TTROIE, i.e. start a new task, call reasoning, filter the solutions, and access data services.

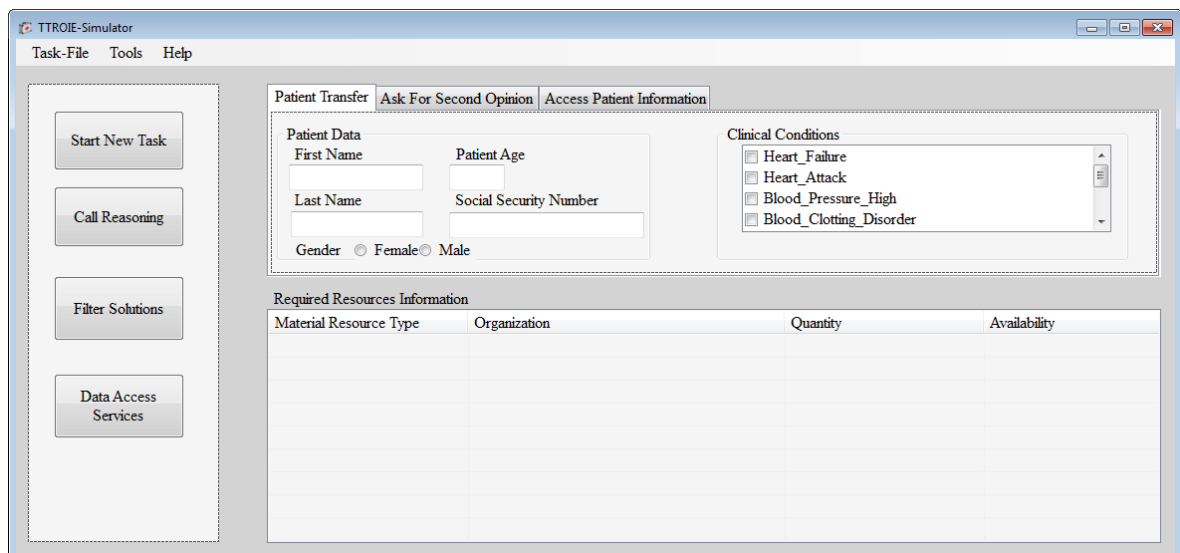


Figure 55. Main TTROIE Graphical User Interface

Once the user starts a new task, new instances of the different material resources, organizations, locations are initialized in the relevant ontology. A confirmation message appears to tell the user that these instances have been initialized in the corresponding ontologies (Figure 56). Then the user is invited to fill the task form with the required data, i.e. patient data and its clinical status, and the instances of task, patient, and clinical conditions are created and inserted in the ontology.

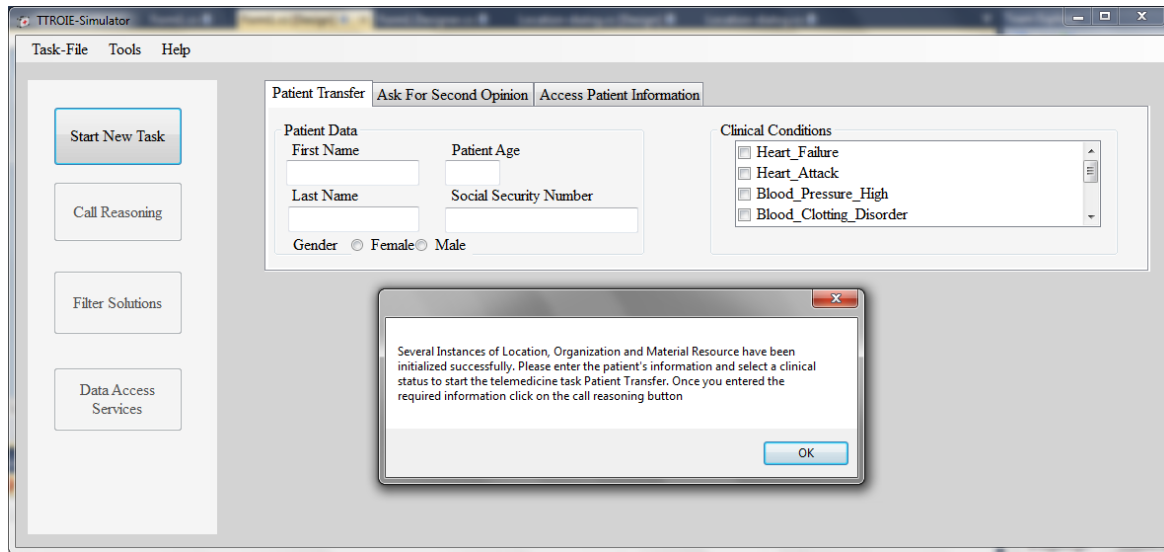


Figure 56. Confirmation Message of the Initialization of New Instances

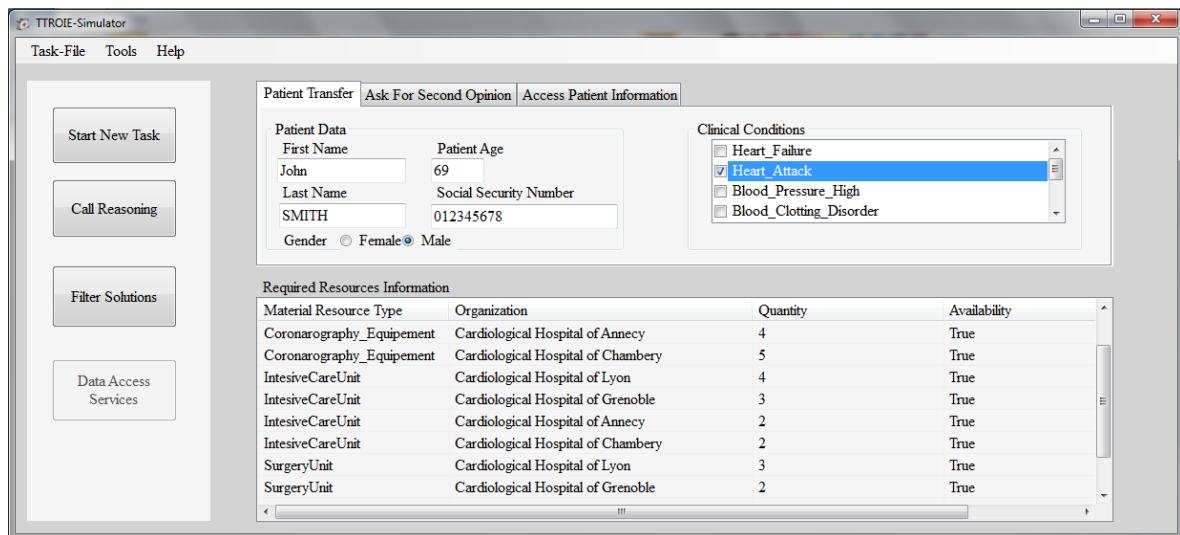


Figure 57. Heterogeneous Resources Data generated by TTROIE

As TTROIE has been realized with Protégé, several axioms can be inferred by the inference engine by applying the rules related to the patient transfer task. The inferred axioms define the heterogeneous resources required by the patient transfer task. The resources data are retrieved by a simple SPARQL query. Additionally, resources data filtering processes can be performed to take account of other contextual data, e.g. the location selected by the user, see the bottom right panel of Figure 57. As we have seen in section VI.3.3, this can easily be performed by integrating the

modifiers `FILTER` and `OPTIONAL` in the SPARQL queries in order to retrieve just the resources data corresponding to the filter expression in the SPARQL queries.

Providing information about the different available resources needed by the patient transfer task enables the first aid persons to make the right decision in order to transfer the patient to the most appropriate hospital.

---

## VI.5 Conclusion

In this chapter, we have presented the final phase of our ONOF-PAS construction methodology concerning the implementation of the ONOF-PAS system. Using the Protégé development environment, we have experimented and validated our ontological models and we have successfully realized a rule-based reasoning to infer the set of heterogeneous resources and services required by a given task running in pervasive environments. Additionally, we have developed the TTROIE prototype to demonstrate the feasibility and the utility of our proposal in terms of optimizing the access to different resources and services in the telemedicine domain.

TTROIE enables users with different attitudes and skills, to perform various eHealth or Telemedicine tasks, particularly, when the tasks are carried out in hostile environments, such as the high mountain ski resorts, where the resources conditions in terms of availability and accessibility are extremely vital.

# *Chapter VII*

## *Conclusion & Perspectives*



## Chapter VII Conclusion and Perspectives

This chapter presents the conclusion of this thesis and identifies some topics that we believe are relevant for future research. Firstly, we provide general remarks concerning the proposed ONOF-PAS framework. Then, we summarize our main research contributions. Finally, we present directions for future research.

---

### VII.1 General Remarks

Nowadays, we witness an important and increased development of smart phones and tablets equipped with high definition and touch screens that run high performance mobile operating systems e.g. iOS of Apple, Android of Google and Windows Mobile Phone of Microsoft. These new devices enable user mobility and open the door to rich and very diverse mobile applications. Based on this reality, the ONOF-PAS framework offers a promising tool that will allow mobile application designers and developers to integrate knowledge engineering in mobile applications. The ONOF-PAS framework will facilitate the design of complex telemedicine assistance processes as we have seen through the scenario of a patient transfer in inimical environments.

Performing rule-based reasoning is still a technical challenge due to the incompatibility of standards and languages. For instance, SWRL rules can be applied by the JESS inference engine using SWRLJess-Tab. However, these rules are not applicable when using the Pellet Reasoner or the Jena API. To integrate the Jena API in the TTROIE software architecture, we had to rewrite rules in the Jena format. On the other side, for developers who prefer to use the OWL API, they have to write rules in the OWL API format. Thus, we still need an integration environment that permits all semantic web technologies to perfectly run together.

Given the services multiplicity and the problems related to web services in terms of discovering, invoking, composition and execution, we have noticed that it would be more efficient to make available for users multiple tasks that perform according to their profiles. In the knowledge meta-model we have defined the relation “Task invokes Service”, where we suppose that the invoked services are already available in the services registry. Instead to perform services composition, we opted to define tasks

that can invoke the available services. That is why, in this thesis, we have mainly focused our work on task performing and on how to link the task with the required resources (service, data sources, and devices), depending on the context of the objects treated by the task.

During the ONOF-PAS framework construction, we were lead to tackle multiple research topics related to ontology mapping, services management, as well as security and privacy issues. We however did not deeply address all these topics because, from one side, they are out of the scope of this thesis, and from the other side, many research works have deeply dealt with these themes and very good results have already been obtained. Nevertheless, for the sake of clarity, we have provided a general background of the aforementioned issues, including the main methods used in each subject.

The use of web services is restricted to resources data updating. However, web services can be used to access certain parts of the patient's EHR. In this case, the issues related to data privacy must be discussed. Also, web services can be used to retrieve general information, e.g. forecast, road circulation status, distance and time calculation, etc.

We have not implemented all the aspects studied in this thesis, but only the patient transfer task to validate our approach and to perform the proof of the concepts developed in our research work.

---

## VII.2 Conclusion

Given the variety of scenarios, the need of a generic, scalable, and interoperable architecture, as well as the complexity of the tasks that systems should accomplish, we have proposed a framework that we called ONOF-PAS, standing for Ontology Oriented Framework for Pervasive Applications and Services. ONOF-PAS consists in:

a) A Knowledge meta-model to link the different entities of a pervasive environment, such as Actor, Task, Object, Object Conditions, Service, Resource, Organization, etc. The knowledge meta-model is extensible to different domains, e.g. e-Health, e-Government. We have represented each model using an ontology including concepts, object type properties, data type properties, and restrictions. The ontological models capture the knowledge about the actors, the task they perform, the distributed

heterogeneous resources required by these tasks and the organizations that own these resources.

- b) A rule base to link the different tasks with the required heterogeneous resources as well as to prioritize the messages generated by these tasks.
- c) An inference engine to realize a rule-based reasoning in order to infer the various resources required by the tasks.
- d) Object oriented components for tasks management and resources data processing.

Our contributions in this dissertation can be summarized as follows:

- *A Generic, Task-driven, and Ontology-based Telemedicine System Architecture*, (Nageba 2007), (Nageba 2008). Our target, through the design of such an architecture, is to enable telemedicine actors to perform various telemedicine tasks which concern patients or any other object, e.g. the EHR. According to the patient's clinical conditions, certain resources required by the tasks can be inferred. The main feature of the proposed architecture is the ability of handling different use contexts depending on the availability and the capability of the required material, communication, and human resources.
- *Provision of Solutions Supporting the Quality of Telemedicine Services in Pervasive Applications*, (Nageba 2009a), (Nageba 2009b). As we have early stated, the first aid persons have different skills and aptitudes in terms of medical knowledge and experiences, and they perform numerous telemedicine tasks in inimical environments which lack of adequate telemedicine infrastructures. Our proposed solution provides first aid persons, anywhere and anytime, with pertinent information concerning patient orientation or transfer. In other words, instead of letting the first aid persons wasting their time doing marginal tasks, e.g. making communication calls, waiting for responses, verifying resources availability, etc, we make available for them a system capable to infer the different available resources required for patient treatment, as well as to propose a list of recipients, i.e., hospitals, who have or manage the available resources. Doing this, we have succeeded in assuring more rapid and reliable medical tele-assistance and consequently to enhance the quality level of telemedicine services.



- *Decision Making Process Adaptation to the Dynamic Context of Resources in Pervasive Computing*, (Nageba 2011a). The decision making process in scenarios like the one we have presented is very complex and requires many algorithms, i.e. “one algorithm per case”. Thus, we have proposed an intelligent compromise solution by combining knowledge about the objects treated by the tasks, i.e. the patient in our application scenario, and their context, i.e. the patients’ clinical conditions. In the proposed knowledge-meta-model we have defined the following relationships between the ontological models: “*Task concerns Object*”, “*Object has Conditions*”, “*Task requires Resource*”. On the other side, we have constructed a set of rules to link each task with its required resources according to the conditions of the objects treated by the task. Thus, to help a first aid person to take a decision related to a patient’s orientation, he/she should just precise the task that he/she wants to perform and the clinical status of the patient concerned by the selected task. Then, the task related resources will be automatically inferred, taking into consideration the conditions of the resources in terms of availability, capability, accessibility. Consequently, a list of organizations, e.g. hospitals, which meet the first aid persons and/or patients needs can be provided. In this way, we can handle several contextual situations and support decisions making to solve problems.
- *Telemedicine System for Hostile Environment Prototype*, (Nageba 2011b). We have edited and tested all the ontological models of ONOF-PAS using the Protégé IDE. Additionally, using an inference engine, we have effectuated a rule-base reasoning based on a set of rules we have defined using a rule description language such as SWRL. We have developed a prototype we called TTROIE, Telemedicine Tasks Resources Ontologies for Inimical Environments, to implement the patient orientation task.

---

### VII.3 Perspectives

During the realization of this thesis, we have identified several elements that constitute guidelines for future research activities. We expose below such of the main future research directions we propose.

- *Object-related Process Inference and Reusable Knowledge Management*: We considered that tasks are independent, viz we did not define a relationship to link a task with the other ones. Users can perform several tasks according to their profiles and needs. However, relationships should be defined to link the different tasks that concern the same object. Thus, knowledge-based systems may trace the user's activities to deduce what we could call "Inferred Processes". In other words, the consecutive tasks performed by the users to treat the same object may be stored in the knowledge base as a process. A mechanism should be defined to enable knowledge-based systems to infer and manage object-related processes.
- *Reasoning Optimization Mechanisms*: From a technical point of view, each time we call the rule-based reasoning process, the inference engine reasons over all the instances stored in the knowledge base. As a consequence, the inference engine performance is penalized by the large number of instances, especially in huge ontologies. Also, this requires additional time for computing and more computational performance in terms of processor rapidity and memory size. For this reason, a mechanism should be defined and realized to increase the inference engines' performance. One of the possibilities to optimize reasoning is to extract just the relevant OWL knowledge, instead to reason over all the stored knowledge, as not all instances are concerned by the rule-based reasoning. The required instances can be determined from each rule.
- *Large Scale Evaluation of the TTROIE system*: In fact, the TTROIE system is still a prototype, and it would be relevant to work on a reference implementation. Once the reference implementation process will have been completed, it is probable that feedbacks on the architecture will lead us to effectuate some modifications to avoid mistakes or to improve the system functionalities. Eventually, an adequate evaluation process for testing the proposed telemedicine system has to be designed. It should provide results about the impact of using ontologies and inference engines on the system performance.
- *Data Privacy Preservation in Telemedicine*: Unfortunately, most of the existing health systems lack privacy modules, some systems provide only basic security mechanisms and only a few systems offer advanced

and sufficient security options. In eHealth, the privacy issue is being treated by the ongoing PAIRSE project (Preserving Privacy in Peer to Peer Environments) in which we are involved (Nageba 2011c). The PAIRSE project aims at designing an original framework for preserving the privacy and confidentiality of shared or exchanged medical data. The proposed framework includes a new mechanism based on query modification to take into account the privacy issues. This mechanism assumes the existence of a privacy policy specifying for each data item: (i) who may access it, (ii) for what purposes and (iii) under what conditions. Data privacy policies are defined on domain ontologies and manage data disclosure at a very fine-grained level, at the datatype property level. Privacy conditions involved with datatype properties have the form of SPARQL queries over domain ontologies. The proposed mechanism modifies the users' queries to accommodate privacy conditions before their resolution by the core composition algorithm. The proposed privacy preservation model takes the user's privacy preferences into account as well. We intend to continue to collaborate with our research group partners of the PAIRSE project to realize an efficient privacy solution.

# *Bibliography*



# Bibliography

- (Alonso 2004) Alonso, G. et al. (2004). Web Services, Concept Architecture and Applications. Data-Centric Systems and Applications, Springer, 354 p.
- (Ammon 2008) Ammon, D. et al. (2008). Developing an architecture of a knowledge-based electronic patient record. In Proceedings of the 30th international conference on Software engineering, pp 653-660.
- (Anya 2010) Anya, O. et al. (2010). Context-aware knowledge modelling for decision support in e-health. In The 2010 International Joint Conference on Neural Networks (IJCNN), pp 1-7.
- (Aoki 2003) AOKI, N., DUNN, K. & JOHNSON-THROOP, K.A. (2003). Outcomes and Methods in Telemedicine Evaluation. Telemedicine and eHealth, Vol. 9, n° 4, pp 393-402.
- (ARTEMIS 2004) ARTEMIS. (2004). Artemis Project Home Page. A Semantic Web Service-based P2P Infrastructure for the Interoperability of Medical Information Systems. <http://www.srdc.metu.edu.tr/webpage/projects/artemis/>.
- (Aura 2002) Aura. (2002). Project Aura. Project Aura Distraction-free Ubiquitous Computing. <http://www.cs.cmu.edu/~aura/>.
- (Baader 2003) Baader, F. & Nutt, W. (2003). Basic description logics. In The description logic handbook: theory, implementation, and applications, pp 43-95.
- (Bangemann 1994) Bangemann, M. et al. (1994). Europe and Global Information Society. Bangemann Report Recommendations to the European Council. [http://www.unic.pt/images/stories/publicacoes200801/raport\\_Bangemanna\\_1994.pdf](http://www.unic.pt/images/stories/publicacoes200801/raport_Bangemanna_1994.pdf).
- (Bardram 2004) Bardram, J.E. & Christensen, H.B. (2004). Open Issues in Activity-Based and TaskLevel Computing. In First International Workshop On Computer Support for Human Tasks and Activities, pp 1-7.
- (Bardram 2005) Bardram, E. (2005). Activity-based computing: support for mobility and collaboration in ubiquitous computing. Personal Ubiquitous Comput. Vol. 9, n° 5, pp 312-322.
- (Bardram 2009) Bardram, J. (2009). Activity-based computing for medical work in hospitals. ACM Transactions on Computer-Human Interaction Vol. 16, n° 2, pp 1-36.
- (Barhamgi 2010) Barhamgi, M., Benslimane, D. & Medjahed, B. (2010). A Query Rewriting Approach for Web Service Composition. IEEE Transactions on Services Computing, Vol. 3, n° 3, pp 206-222.
- (Bechhofer 2004) Bechhofer, S. et al. (2004). OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features/>.
- (Beolchi 2003) Beolchi, L. (2003). European telemedicine glossary of concepts, standards, technologies and users. (Brussels): European Commission, Information Society.
- (Berners-Lee 2001) Berners-Lee, T., Hendler, James & Lassila, O. (2001). The Semantic Web: Scientific American. <http://www.scientificamerican.com/article.cfm?id=the-semantic-web>.

- (Beuscart 2004) Beuscart, R. et al. (2004). Homecare : The need for Cooperative Information Systems. MEDINFO 2004 IMIA, pp 1343-1347.
- (BFO 2011) BFO. (2011). BFO, Basic Formal Ontology. <http://www.ifomis.org/bfo/>.
- (Bicer 2005) Bicer, V. et al. (2005). Artemis message exchange framework: semantic interoperability of exchanged messages in the healthcare domain. SIGMOD Rec, Vol. 34, n° 3, pp 71-76.
- (Blobel 2007) Blobel, B. & Pharow, P. (2007). A model driven approach for the German health telematics architectural framework and security infrastructure. International Journal of Medical Informatics, no° 76, pp 169-175.
- (Blobel 2009a) Blobel, B. & Oemig, F. (2009). Ontology-Driven Health Information Systems Architectures. Stud Health Technol Inform, Vol. 150, pp 195-199.
- (Blobel 2009b) Blobel, B. et al. (2009). The Role of Ontologies for Sustainable, Semantically Interoperable and Trustworthy EHR Solutions. Stud Health Technol Inform, Vol. 150, pp 953-957.
- (Bonabeau 2002) Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences of the United States of America, Vol. 99, n° 3, pp 7280-7287.
- (Booth 2003) Booth, D. et al. (2003). Web Services Architecture. Web Services Architecture. <http://www.w3.org/TR/ws-arch/>.
- (Bradley 2005) Bradley, N.A. & Dunlop, M.D. (2005). Toward a multidisciplinary model of context to support context-aware computing. Hum.-Comput. Interact. Vol. 20, n° 4, pp 403-446.
- (Bricon-Souf 2007) Bricon-Souf, N. & Newman, C.R. (2007). Context awareness in health care: a review. International Journal of Medical Informatics, Vol. 76, n° 1, pp 2-12.
- (Broekstra 2001) Broekstra, J. et al. (2001). Enabling knowledge representation on the Web by extending RDF schema. In Proceedings of the 10th international conference on World Wide Web, ACM, pp 467-478.
- (Brown 1997) Brown, P., Bovey, J. & Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace. Personal Communications, IEEE, Vol. 4, n° 5, pp 58-64.
- (Chaari 2007) Chaari, T., Ejigu, D., Laforest, F. & Scuturici, V.-M. (2007). A comprehensive approach to model and use context for adapting applications in pervasive environments. J. Syst. Softw. Vol. 80, n° 12, pp 1973-1992.
- (Chen 2004) Chen, Harry, Finin, Tim & Joshi, Anupam. (2004). Semantic Web in the Context Broker Architecture. In Proceedings of PerCom 2004, pp 277--286.
- (Chen 2004) Chen, H., Perich, F., et al. (2004). SOUPA: standard ontology for ubiquitous and pervasive applications. In The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp 258-267.
- (Chibani 2008) Chibani, A., Djouani, K. & Amirat, Y. (2008). Semantic middleware for context services composition in ubiquitous computing. In 1st international conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications, Vol. 278, pp 1-6.
- (Christopoulou) Christopoulou, E. & Kameas, A. (2005). GAS ontology: an ontology for col-

- 2005) laboration among ubiquitous computing devices. *Int. J. Hum.-Comput. Stud.*, Vol. 62, n° 5, pp 664-685.
- (Colucci 20004) Colucci, S. et al. (2004). An agency for semantic-based automatic discovery of web-services. In *Artificial Intelligence Applications and Innovation, AIAI*, pp 1-14.
- (Daconta 2003) Daconta, M.C., Obrst, L.J. & Smith, K.T. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Wiley, 312 p.
- (DAML 2000) DAML. (2000). DAML.org. The DARPA Agent Markup Language Homepage. <http://www.daml.org/>.
- (Dedefa 2007) Dedefa, D., Scuturici, M. & Brunie, L. (2007). CoCA: A Collaborative Context-Aware Service Platform for Pervasive Computing. In *Fourth International Conference on Information Technology, ITNG'07*, pp 297-302.
- (Dedefa 2008) Dedefa, D., Scuturici, M. & Brunie, L. (2008). Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing. *Journal of Computers*, Vol. 3, n° 1, pp 40-50.
- (Dey 2001) Dey, A.K., Abowd, G.D. & Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human Computer Interaction Journal*, Vol 16, n° 2, pp 97-166.
- (Din 2010) Din, I.U. (2010). Remote Patient Disease Diagnosing and Treatment Prototype for Third World/Remote Areas Using Real Time Protocols. In *Computer Modeling and Simulation, IEEE Computer Society International Conference*, pp 659-664.
- (Dombroviak 2007) Dombroviak, K.M. & Ramnath, R. (2007). A taxonomy of mobile and pervasive applications. In *Proceedings of the 2007 ACM symposium on Applied computing, SAC*, pp 1609-1615.
- (Doran 1996) Doran, C., Ferguson, E. & Nicogossian, A. (1996). Telemedicine and Telescience in the US space program. In *20th International Symposium on Space Technology and Science*, p 1-7.
- (Fayn 2010) Fayn, J. & Rubel, P. (2010). Towards a Personal Health Society in Cardiology. *IEEE Transactions on Information Technology in Biomedicine*, Vol. 14, n° 2, pp 401-409.
- (Fensel 2008) Fensel, D., Harmelen, F. van & Andersson, B. (2008). Towards LarKC: A Platform for Web-Scale Reasoning. In *IEEE International Conference on Semantic Computing*, pp 524-529.
- (Ferreira 2007) Ferreira Da Silva, C. (2007). *Découverte de correspondances sémantiques entre ressources hétérogènes dans un environnement coopératif*. Thèse de doctorat, Lyon, France: Université Claude Bernard (Lyon).
- (Fox 2003) Fox, J. & Beveridge, M. (2003). Understanding Intelligent Agents: analysis and synthesis. *Aicommunications* Vol. 16, n° 3, pp 139-152.
- (Fukazawa 2006) Fukazawa, Y. et al. (2006). Role-Task Model for Advanced Task-Based Service Navigation System. In *Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science*, Vol. 4251, pp 1021-1028.



- (Gagneux 2008) Gagneux, M. et al. (2008). Pour un Dossier Patient Virtuel et Partage et une Stratégie Nationale des Systèmes d'Information de Santé. Rapport de la mission de relance du DMP. [www.sante.gouv.fr/IMG/pdf/Rapport\\_DMP\\_mission\\_Gagneux.pdf](http://www.sante.gouv.fr/IMG/pdf/Rapport_DMP_mission_Gagneux.pdf)
- (Galarraga 2007) Galarraga, M. et al. (2007). Proposal of an ISO/IEEE11073 Platform for Healthcare Telemonitoring: Plug-and-Play Solution with new Use Cases. In the 29th Annual International Conference of the IEEE EMBS, pp 6709-6712.
- (Ganguly 2005) Ganguly, Probnab, Ray, P. & Parameswaran, N. (2005). Semantic interoperability in telemedicine through ontology-driven services. *Telemedicine Journal and E-Health: The Official Journal of the American Telemedicine Association*, Vol. 11, n° 3, pp 405-412.
- (Georgalas 2007) Georgalas, N. et al. (2007). Towards a Model-Driven Approach for Ontology-Based Context-Aware Application Development: A Case Study. In Fourth International Workshop on Model-Based Methodologies for Pervasive and Embedded Software, MOMPES'07, pp 21-32.
- (Giansanti 2007) Giansanti, D., Morelli, S. & Macellari, V. (2007). Experience at Italian National Institute of Health in the assessment of the quality in telemedicine: submission to a quality control process. In The 29th Annual International conference of the IEEE EMBS, pp 3725-3728.
- (Gomez-Peroz 1999) Gomez-Peroz, A. (1999). Evaluation of taxonomic Knowledge in ontologies and Knowledge bases. In Proceeding of the 12th workshop on knowledge acquisition, Modeling and Management (KAW'99), Vol. 2, pp 611-618.
- (Gruber 1993) Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, Vol. 5, n° 2, pp 199-220.
- (Gu 2004) Gu, Tao et al. (2004). An ontology-based context model in intelligent environments. In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, pp 270--275.
- (Gu 2005) Gu, Tao, Pung, Hung Keng & Zhang, Da Qing. (2005). A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, Vol. 28, n° 1, pp 1-18.
- (Guarino 1998) Guarino, N. (1998). Formal ontology and information systems. In *Formal Ontology in Information Systems*, IOS Press, pp 3-15.
- (Healy 2006) Healy, J. C. (2006). Editorial of the Special Health and Internet for All. *International Journal of Medical Informatics*, Vol.75, n° 1, pp 5-7.
- (Healy 2010) Healy, P. et al. (2010). Web-based Remote Monitoring of Live EEG. In *IEEE HealthCom 2010*, pp 169-174.
- (Heijst 1997) Heijst, G. van, Schreiber, A.T. & Wielinga, B.J. (1997). Using explicit ontologies in KBS development. *Int. J. Hum.-Comput. Stud.*, Vol. 46, n° 2-3, pp 183-292.
- (Hewson 2007) J. Hewson, D., Duchène, J., Charpillet, F. et al. (2007). The PARACHute Project: Remote Monitoring of Posture and Gait for Fall Prevention. *EURASIP Journal on Advances in Signal Processing*, Vol. 2007, pp 1-16.
- (Hong 2007) Hong, C.-S. et al. (2007). Context Modeling and Reasoning Approach in Context-Aware Middleware for URC System. *International Journal of Mathematical, Physical and Engineering Sciences*, Vol. 1, n° 4, pp 208-212.

- (Hsieh 2007) Hsieh, F.-S. (2007). Context-aware Workflow Driven Resource Allocation for e-Healthcare. In 9th International Conference on e-Health Networking, Application and Services, pp 34-39.
- (Huang 2008) Huang, R. et al. (2008). Semantic Web-based Context-aware Service Selection in Task-computing. In International Workshop on Modelling, Simulation and Optimization, IEEE Computer Society, pp 97-101.
- (IBM 2009) IBM. (2009). The Benefits of Cloud Computing a New Era of Responsiveness, Effectiveness and Efficiency in IT Service Delivery. <http://public.dhe.ibm.com/common/ssi/ecm/en/diw03004usen/DIW03004USEN.PDF>
- (Ifremmont 2007) Ifremmont. (2007). Ifremmont : institut de formation et de recherche en médecine de montagne - Introduction. <http://ifremmont.com/>.
- (Jean 2005) Jena. (2005). Jena Semantic Web Framework. <http://jena.sourceforge.net/>.
- (JESS 2008) JESS. (2008). Jess, the Rule Engine for the Java Platform. <http://www.jessrules.com/jess/index.shtml>.
- (Jiang 2008) Jiang, F., Li, J. & Zhu, Z. (2008). A User-centric Task Computing Architecture for Pervasive Computing. In Third IEEE International Conference on Pervasive Computing and Applications, pp 491-469.
- (Judd 2003) Judd, G. & Steenkiste, P. (2003). Providing Contextual Information to Pervasive Computing Applications. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, pp 133-142.
- (Jumaa 2010) Jumaa, H.A. (2010). XML and Relational Databases Mediation Automation. Lyon, France: Thesis, INSA-Lyon.
- (Jung 2007) Jung, M.Y. et al. (2007). Implementation of the Relay Server for the Ubiquitous Integrated Biotelemetry System for Emergency Care (UIBSEC) based on 3.5G HSDPA Technology. In the 29th Annual International Conference of the IEEE EMBS, pp 6713-6716.
- (K. Yoo 2004) K. YOO, S., M. Kim, K. & M. Jung, S. (2004). Design of Multimedia Telemedicine System for Inter-hospital Consultation. In IEEE Engineering in Medicine and Biology Society, Vol. 2, pp 3109-3111.
- (Kalfoglou 2003) Kalfoglou, Y. & Schorlemmer, M. (2003). Ontology mapping: the state of the art. Knowl. Eng. Rev, Vol. 18, n° 1, pp 1-31.
- (Kang 2007) Kang, J. et al. (2007). HSDPA (3.5G)-Based Ubiquitous Integrated Biotelemetry System for Emergency Care. In the 29th Annual International Conference of the IEEE EMBS, pp 3665-3668.
- (Karavatselou 2001) Karavatselou, E.I. & Economou, G.K. (2001). OTE-TS—A New Value-Added Telematics Service for Telemedicine Applications. In IEEE Transactions on Information Technology in Biomedicine, Vol. 5, pp 210-224.
- (Kariyawasam 2010) Kariyawasam, A.A. et al. (2010). Mobile Healthcare Network and Distance Consulting on 3G. In IEEE Healthcom 2010, pp 148-153.
- (Kifer 2005) Kifer, M. (2005). Rules and Ontologies in F-Logic. In Reasoning Web, Springer, pp 22-34.
- (Kim 2006) Kim, W. et al. (2006). Activity Policy-Based Service Discovery for Perva-

- sive Computing. Lecture Notes in Computer Science, Vol. 4254, pp 756-768.
- (Kolovou 2004) Kolovou, L.T. et al. (2004). TM-Provider: A new Communicatory Model for Telemedicine Services support. In the 26th Annual International Conference of the IEEE EMBS, pp 3112-3115.
- (Koutkias 2005) Koutkias, V.G. & Chouvarda, Ioanna. (2005). A Multiagent System Enhancing Home-Care Health Services for Chronic Disease Management. IEEE Transactions on Information Technology in Biomedicine Vol. 9, pp 528-537.
- (Kumar 2003) Kumar, M. et al. (2003). PICO: A Middleware Framework for Pervasive Computing. IEEE Pervasive Computing, Vol. 2, n°. 3, pp 72-79.
- (Kumar 2008) Kumar, B. & Mohan, A. (2008). Novel Efficient and Secure Medical Data Transmission on WiMAX. Telemedicine and e-Health, Vol. 14, n°. 10, pp 1063-1069.
- (Lage 2004) Lage, A. L. et al. (2004). A Quality of Service Framework for Tele-Medicine Application. In IEEE Proceedings WebMedia and LA-Web, pp 18-20.
- (Latfi 2007) Latfi, F., Lefebvre, B. & Descheneaux, C. (2007). Ontology-Based Management of the Telehealth Smart Home, Dedicated to Elderly in Loss of Cognitive Autonomy. In Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Vol. 258.
- (Leblevec 2008) Leblevec, Y. (2008). Towards the conception of extended information systems organized around a web service platform. Thesis, Lyon, France.
- (Lee 2004) Lee, C. C. & Shih, C. Y. (2004). An Adaptive Multimedia Transmission Control for Telemedicine Application. In Computer Symposium, pp 467-470.
- (Lee 2009) Lee, D. & Meier, R. (2009). A hybrid approach to context modelling in large-scale pervasive computing environments. In the Fourth International ICST Conference on COMMunication System softWARE and middlewaRE, pp 1-12.
- (Lee 2006) Lee, S. et al. (2006). Emergency Care Management with Location-Aware Services. In Pervasive Health Conference and Workshops, pp 1-6.
- (Leite 2009) Leite, C. et al. (2009). Middleware for remote healthcare monitoring. In International Conference on Innovations in Information Technology, pp 185-189.
- (Lhotska 2007) Lhotska, L. (2007). Multi-Agent System as a Platform for Management of Medical Documentation. In the 29th Annual international conference of the IEEE EMBS, pp 3661-3664.
- (Li 2003) Li, L. & Horrocks, I. (2003). A software framework for matchmaking based on semantic web technology. In the 12th international conference on World Wide Web, pp 331-339.
- (LOINC 1994) LOINC. (1994). Logical Observation Identifiers Names and Codes (LOINC®). <http://www.loinc.org/>
- (Lu 2010) Lu, S. H. et al. (2010). Pervasive Health Service System: insights on the development of a Grid-based personal health service system. In IEEE HealthCom2010, pp 61-67.
- (Lu 2010) Lu, W. P., Leung, H. & Estrada, E. (2010). Transforming Telemedicine for

- Rural and Urban Communities Telemedicine 2.0 – Any Doctor, Any Place, AnyTime. In IEEE HealthCom2010, pp 379-385.
- (Luepker 2005) Luepker, R. (2005). Delay in Acute Myocardial Infarction: Why don't they come to the hospital more quickly and what can we do to reduce delay?. American Heart Journal, Vol.150, n° 3, p 368-370.
- (M. Wiig 2003) M. Wiig, K. (2003). A knowledge Model for Situation-Handling. <http://www.kwork.org/Resources/SituationHandlingModel.pdf>
- (Macal 2006) Macal, C.M. & North, M.J. (2006). Tutorial on agent-based modeling and simulation part 2: how to model with agents. In Proceedings of the 38th conference on Winter simulation, pp 73-83.
- (Maglaveras 2005) Maglaveras, N. et al. (2005). The Citizen Health System (CHS): A Modular Medical Contact Center Providing Quality Telemedicine Services. In IEEE Transactions on Information Technology in Biomedicine, Vol. 9, pp 353-362.
- (Maglaveras 2006) Maglaveras, N. et al. (2006). Quality Home Telemedicine Services for Chronic Cardiac Disease Patients through the INTERLIFE Platform. Computers in Cardiology, Vol. 33, pp 245-248.
- (Malatras 2009) Malatras, A. & Hirsbrunner, B. (2009). A Context-Aware Framework to Enable Adaptation in Pervasive Computing Environments. In International Conference on Network-Based Information Systems, pp 182-187.
- (Martinez 2007) Martinez, I. et al. (2007). Implementation experience of a patient monitoring solution based on end-to-end standards. In Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp 6426-6429.
- (McGinn 2005) McGinn, A. et al. (2005). Trends in Prehospital Delay Time and Use of Emergency Medical Services for Acute Myocardial Infarction: Experience in 4 US Communities from 1987-2000. American Heart Journal. Vol. 150, n° 3, p 392-400.
- (McIlraith 2001) McIlraith, S.A., Son, T.C. & Zeng, H. (2001). Semantic Web Services. IEEE Intelligent Systems, Vol. 16, n° 2, pp 46-53.
- (Mizoguchi 2000) Mizoguchi, R. et al. (2000). Construction and developement of a plant ontology. In the 12th International conference, EKAW 2000, pp 113-128.
- (Morvan 2009) Morvan, F. and Hameurlain, A. (2009). Dynamic Query Optimisation: towards decentralised methods. International Journal of Intelligent Information and Databases Systems. Vol.3, n° 4, pp 461-482.
- (Nageba 2007) Nageba, E., Fayn, J. & Rubel, P. (2007). A generic task-driven multi-agent telemedicine system. In Conference Proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp 3733-3736.
- (Nageba 2008) Nageba, E., Fayn, J. & Rubel, P. (2008). An ontology-based telemedicine tasks management system architecture. In the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp 1494-1497.
- (Nageba 2009a) Nageba, E., Fayn, J. & Rubel, P. (2009). A Model Driven Ontology-based Architecture for Supporting the Quality of Services in Pervasive Telemedicine Applications. In IEEE, 3rd International Conference on Pervasive Com-

- puting Technologies for Healthcare, pp 1-8.
- (Nageba 2009b) Nageba, E., Fayn, J. & Rubel, P. (2009b). A knowledge model driven solution for web-based telemedicine applications. *Studies in Health Technology and Informatics*, Vol. 150, pp 443-447.
- (Nageba 2011a) Nageba, E., Fayn, J. & Rubel, P. (2011). A Telemedicine System for Hostile Environments. In *Advances in Telemedicine: Technologies, Enabling Factors and Scenarios*, InTech, pp 397-412.
- (Nageba 2011b) Nageba, E., Defude, B., Morvan, F., et al. (2011). Data Privacy Preservation in Telemedicine: The PAIRSE Project. *Studies in Health Technology and Informatics*, Vol. 169, pp 661-665.
- (Nageba 2011c) Nageba, E., Fayn, J. & Rubel, P. (2011). Adaptation des processus de prise de décision au contexte dynamique des ressources en informatique ubiquitaire. Application en eSanté. In *Actes des septièmes journées francophones Mobilité et Ubiquité 2011*, pp 117-122.
- (Nanda 2007) Nanda, P. & C. Fernandes, R. (2007). Quality of Service in Telemedicine. In *IEEE First International Conference on the Digital Society*, pp 2-2.
- (Nee 2008) Nee, O. et al. (2008). SAPHIRE: intelligent healthcare monitoring based on semantic interoperability platform: pilot applications. *IET Communications* Vol. 2, n° 2, pp 192-201.
- (Norris 2002) Norris, A.C. (2002). *Essentials of Telemedicine and Telecare*. Wiley. 188 p.
- (O'Brien 2010) O'Brien, H. et al. (2010). Smart Phone Interfaces to Wireless Health Sensors. In the 12<sup>th</sup> IEEE International Conference on eHealth Networking Applications and Services, Healthcome, pp 180-186.
- (ODM 2005) ODM. (2005). *Ontology Definition Metamodel*. [www.omg.org/docs/ad/05-08-01.pdf](http://www.omg.org/docs/ad/05-08-01.pdf).
- (Oemig 2009) Oemig, F. & Blobel, B. (2009). Semantic interoperability between health communication standards through formal ontologies. In *Stud Health Technol Inform*, Vol. 150, pp 200-204.
- (OWL API 2007) OWL API. (2007). The OWL API. <http://owlapi.sourceforge.net/>.
- (Paganelli 2007) Paganelli, F. & Giuli, D. (2007). An Ontology-Based Context Model for Home Health Monitoring and Alerting in Chronic Patient Care Networks. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications*, IEEE Computer Society, Vol. 2, pp 838-845.
- (Paganelli 2008) Paganelli, Federica, Spinicci, E. & Giuli, Dino. (2008). ERMHAN: A Context-Aware Service Platform to Support Continuous Care Networks for Home-Based Assistance. *International Journal of Telemedicine and Applications*, Vol. 2008, pp 1-13.
- (Paolucci 2002) Paolucci, M. et al. (2002). Semantic Matching of Web Services Capabilities. In the *First International Conference on The Semantic Web*, Springer ISWC'02, pp 333-347.
- (Pellet 2007) Pellet. (2007). Pellet: The Open Source OWL Reasoner. <http://clarkparsia.com/pellet>.
- (Pharow 2009) Pharow, P. et al. (2009). Portable Devices, Sensors and Networks: Wireless Personalized eHealth Services. *Stud Health Technol Inform*, Vol. 150, p

- 1012-1016.
- (Pinto 2001) Pinto, H.S. & Martins, J.P. (2001). A methodology for ontology integration. In the 1st international conference on Knowledge capture, pp 131–138.
- (Plas 2006) Plas, D. J. et al. (2006). Manipulating context information with SWRL. Project A-MUSE, Deliverable D3.12, 44p.
- (Qureshi 2005a) Qureshi, A. & Gutttag, J. (2005). Horde: separating network striping policy from mechanism. In Proceedings of the 3rd international conference on Mobile systems, applications, and services, pp 121-134.
- (Qureshi 2005b) Qureshi, A., Shoeb, A. & Gutttag, J. (2005). Building a High-Quality Mobile Telemedicine System using Network Striping over Dissimilar Wireless Wide Area Networks,. In IEEE 27th Annual International Conference of the Engineering in Medicine and Biology Society EMBS, pp 3942- 3945.
- (Rubel 2005) Rubel, P. et al. (2005). Toward personal eHealth in cardiology. Results from the EPI-MEDICS telemedicine project. Journal of Electrocardiology, Vol. 38, n° 4, pp 100-106.
- (Saha 2003) Saha, D. & Mukherjee, A. (2003). Pervasive Computing: A Paradigm for the 21st Century. Computer, Vol. 36, n° 3, pp 25-31.
- (Sain 2010) Sain, M. et al. (2010). Designing context awareness middleware architecture for personal healthcare information system. In The 12th International Conference on Advanced Communication Technology, Vol. 2, pp 1650-1654.
- (Sankaran 2000) Sankaran, S.R. & Bui, T. (2000). A Web-Based Correctional Telemedicine System with Distributed Expertise. In the 33rd International Conference on System Sciences, pp 1-10.
- (SAPHIRE 2008) SAPHIRE. (2008). SAPHIRE Project. Intelligent Healthcare Monitoring. <http://www.srdc.metu.edu.tr/webpage/projects/saphire>
- (Sasa 2008) Sasa, A., Juric, M.B. & Krisper, M. (2008). Service-Oriented Framework for Human Task Support and Automation. IEEE Transactions on Industrial Informatics Vol. 4, n° 4, pp 292-302.
- (Satyanarayanan 2001) Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. Personal Communications, IEEE, Vol 8, n° 4, pp 10-17.
- (Schilit 1994) Schilit, B. & Theimer, M. (1994). Disseminating Active Map Information to Mobile Hosts. IEEE NETWORK, n° 8, pp 22-32.
- (Schmolze 1985) Schmolze, J.G., Beranek, B. & Inc, N. (1985). An overview of the KL-ONE knowledge representation system. Cognitive Science, Vol. 9, pp 171-216.
- (Scholl 2009) Scholl, J., Lambrinos, L. & Lindgren, A. (2009). Rural Telemedicine Networks Using Store and-Forward Voice-over-IP. In Study in Health Technology and Informatics, Vol. 150, pp 448-452.
- (Shirazi 2004) Shirazi, B., Kumar, M. & Sung, B.Y. (2004). QoS Middleware Support for Pervasive Computing Applications. In IEEE Proceedings of the 37th International Conference on System Sciences, pp 1-10.
- (Sim 2006) Sim, J. et al. (2006). Applying Inference Engine into Context-Aware Computing Services. In the 2<sup>nd</sup> International Workshop on Personalized Context Modeling and Management for UbiComp Applications. UbiPCMM06, pp 1-5.

- (Simon 1991) Simon, H. (1991). Organizations and Markets. *Journal of Economic Perspectives* Vol. 5, n°. 2, pp 25-44.
- (SNOMED 1999) SNOMED. (1999). IHTSDO: International Health Terminology Standards Development Organisation. <http://www.ihtsdo.org/snomed-ct/>
- (Stair 2003) Stair, R. and Reynolds, G. (2003). *Principals of Information Systems*. Cours Technology. Edition 6, 132 p.
- (Strang 2004) Strang, T. & Linnhoff-Popien, C. (2004). A Context Modeling Survey. In *The Sixth International Conference on Ubiquitous Computing, Workshop on Advanced Context Modelling, Reasoning and Management*, pp 1-8.
- (Strimpakou 2006) Strimpakou, M.A., Roussaki, I. & Anagostou, M. (2006). A Context Ontology for Pervasive Service Provision. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, Vol. 2, pp 775-779.
- (SUMO 2011) SUMO. (2011). The Suggested Upper Merged Ontology (SUMO) - Ontology Portal. <http://www.ontologyportal.org/>.
- (Sun Microsystems 2009) Sun Microsystems. (2009). *Interoduction to Cloud Coputing Architecture*. White Paper, Sun Microsystems.
- (Sutterer 2008) Sutterer, M., Droegehorn, O. & David, K. (2008). UPOS: User Profile Ontology with Situation-Dependent Preferences Support. In *First International Conference on Advances in Computer-Human Interaction*, pp 230-235.
- (Takahashi 2007) Takahashi, H. et al. (2007). A Ubiquitous Care-Support Service based on Agent-oriented Multiple-Context Coordination. *Journal of Networks*, Vol. 2, n° 5, pp 33-45.
- (Taleb 2009) Taleb, T. et al. (2009). A Context-Aware Middleware-Level Solution towards a Ubiquitous Healthcare System. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp 61-66.
- (Telisson 2006) Telisson, D. et al. (2006). A new shared care telemedical solution dedicated to elderly patient nursing services for remote GP decision support. In *Computers in Cardiology*, Vol. 33, pp 229-232.
- (Thiyagarajan 2006) Thiyagarajan, C.A. & Clarke, M. (2006). A Systematic Review of Technical Evaluation in Telemedicine Systems. In *Proceedings of the 28th IEEE EMBS Annual International Conference*, pp 6320-6323.
- (Toledo 2006) De Toledo, P., Jiménez, S. & Del Pozo, F. (2006). Telemedicine Experience for Chronic Care in COPD. *IEEE Transactions on Information Technology in Biomedicine*, Vol. 10, n° 3, pp 567-573.
- (Tran 2007) Tran, V.X. & Tsuji, H. (2007). OWL-T: A Task Ontology Language for Automatic Service Composition. In *IEEE International Conference on Web Services*, pp 1164-1167.
- (UMLS 2009) UMLS. (2009). Unified Medical Language System (UMLS). <http://www.nlm.nih.gov/research/umls/>
- (Uschold 1996) Uschold, M. & Gruninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, Vol. 11, n° 2, pp 93-155.
- (Verdier 2006) Verdier, C. (2006). *Health Information Systems: from Local to Pervasive*

- Medical Data. SAS, Information Decision Patient. Vol.9, pp 87-108.
- (Wang 2004) Wang, X.H. et al. (2004). Ontology based context modeling and reasoning using OWL. In the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp 18-22.
- (Weiser 1999) Weiser, M. (1999). The computer for the 21 century. SIGMOBILE Mob. Comput. Commun. Rev, Vol. 3, n° 3, pp 3-11.
- (WHO 2005) WHO. (2005). WHO: Connecting for Health: Global Vision, Local Insight. [http://www.who.int/kms/resources/wsis\\_report/en/](http://www.who.int/kms/resources/wsis_report/en/).
- (Wootton 2006) Wootton, R., Craig, J. & Patterson, V. (2006). Introduction to Telemedicine. 2nd ed. Royal Society of Medicine Press, 224 p.
- (Wyatt 2005) Wyatt, J.C. & Sullivan, F. (2005). eHealth and the future: promise or peril? BMJ (Clinical Research Ed.), Vol. 331, n°. 7529, pp 1391-1393.





# *Appendix*



# Appendix A: Comparison of Different Inference Engines

The following tables summarize the rule engine survey conducted by (Sim 2006) and (Plas 2006).

Table 1 recapitulates the main specifications of different inference engines such as Pellet, Racér, Jena, etc. Table 2 and Table 3 focus on the general properties of rule engines, more particularly on technical properties like the algorithms which are used, whether and how SWRL is supported, etc., as well as on general issues like licensing and availability.

Table 4 provides a comparison of different inference engines in terms of input/output (formats, classes and instances), and it also shows how far these rule engines support real-time changes of distributed data.

Table 1 Comparison of Different Inference Engines. From (Sim 2006)

|          | Import                 | Inference  |                 |  | Repository                                       |         | Query                                  | Version                 | Source                      |
|----------|------------------------|------------|-----------------|--|--|---------|--|-------------------------|-----------------------------|
|          | Ontology Language      | Capability | Internal Engine | External Engine                          | Persistent Storage                               | Memo-ry | Language                               |                         |                             |
| Minerva  | OWL                    | DL         | O               | Racer, Pellet                            | DB2, Derby, HSQLDB                               | X       | SPARQL                                 | Semantics Toolkit-1.1.1 | IBM                         |
| DLDB-OWL | OWL                    | DL         | X               | Racer                                    | MS-Access, PostgreSQL                            | O       | KIF-Like language                      | HAWK 1.3 beta           | Lehigh University SWAT Lab  |
| Pellet   | DIG tell document, OWL | DL         | O               | X  | X  | O       | DIG ask document, RDQL                 | pellet-1.3              | mindswap                    |
| Racer    | DIG tell document, OWL | DL         | O               | O  | X  | O       | DIG ask document, Racer Query Language | RacerPro 1.9.0          | Racer Systems GmbH & Co. KG |
| JENA     | OWL                    | DL         | O               | Racer Pro, Pellet, Fact++ (JenaOntModel) | MySQL, HSQLDB, PostgreSQL, Oracle, MS-SQL Server | O       | SPARQL                                 | Jena 2.4                | HP Labs                     |

Table 2 General Properties of Rule Engines (1). From (Plas 2006)

| Rule Engine | Forward chaining | Backward chaining | Algorithm     | Availability                                | Licensing       | IDE                         |
|-------------|------------------|-------------------|---------------|---|-----------------|-----------------------------|
| Jess        | Yes              | No                | enhanced RETE | commercial (30day eval)                     | negotiable      | eclipse                     |
| Drools      | Yes              | no                | RETE OO       | open source                                 | Werken Specific |                             |
| Algernon    | yes              | yes               |               | open source                                 | MPL             | Protégé plugin              |
| Jena-2      | Yes              | yes               |               | open source                                 | BSD-style       |                             |
| CommonRules | yes              | no                | no RETE yet   | commercial (90day eval)                     | commercial      |                             |
| Bossam      | yes              | yes?              | RETE          | commercial free for non-commercial purposes |                 | API, command-line interface |

Table 3 General Properties of Rule Engines (2). From (Plas 2006)

| Rule Engine | SWRL Support                           | JSR94 Support | Java Integration                | Maturity   | Comments   |
|-------------|--|---------------|---------------------------------|--|--|
| Jess        | no                                     | beta          | ++                              | ++   | language based on CLIPS  |
| Drools      |  | yes           | ++                              | ++   |  |
| Algernon    |  |               | ++                              |  |  |
| Jena-2      | no                                     |               | ++                              |  | Framework for semantic web apps<br>OWL Lite reasoner   |
| CommonRules | no                                     | no            | ++                              | ? version 3.3  | rule-based framework<br>courteous logic programs (CLP)   |
| Bossam      | yes (but limited support for builtins) | no            | ++ (jars and javadoc available) | alpha version (latest release 0.8b84 - frequently new updates) | supports OWL, SWRL and RuleML. "The latest release uses RIO rdf parser and Piccolo SAX parser instead of Jena/ARP and Xerces." |

Table 4 Input/Output and Run-time Properties. From (Plas 2006)

|             | Mode of working                 | Input  |   |                            | Output  | Special   |
|-------------|---------------------------------|--|---|----------------------------|---|---|
|             |                                 | Classes and instances  | SWRL  | Real-time instance changes | Classes and instances                                       |   |
| Jess        | Shell or batch                  | CLIPS file   | No, rules integrated in CLIPS (Jess) format             |                            |   | Eclipse plugin available  |
| Drools      | Run from JVM (full control)     | Java objects   | No, Drools DRM format                                   | Yes, if introduced in JVM  | Java objects  |   |
| Algernon    | Plugin in Protégé or standalone | Protégé database/ CLIPS files from Protégé                                   | No, Algernon specific                                   | Yes, for Algernon plugin   | Changes in Protégé database for plugin and forward chaining | Backward chaining slow for first request (minutes) faster for subsequent (seconds)<br>No firing of rules on inferred facts. External Knowledge Base |
| Jena-2      | Run from JVM (full control)     | RDF file (Protégé format)  | No, rules in separate file in JENA format               | Yes if introduced in JVM   | RDF file (Protégé format)                                   |   |
| CommonRules | batch mode from JVM             | Files in Courteous Logic Program (CLP) format                                | Included in CLP format with instances                   | No                         | Inferred facts in CLP format                                |   |
| Bossam      | Shell or batch                  | Batch: input formats are Buchingae, OWL, SWRL/ RDF SWRL/ OWL, RuleML, and N3 | Yes in docs; only working in shell mode (takes minutes) | Yes in shell mode          | Batch: only initial data in different formats               | No firing of rules on inferred facts.   |
| Hoolet      | Java GUI                        | RDF file (not Protégé format)  | Yes, separate RDF file (not Protégé format)             | No                         | No  |   |

# Appendix B: Extract of the OWLManager Class Java Source Code

```

1  +  /*...*/
5
6  package metier.gestionOWL;
7  import java.util.*;
8  import com.hp.hpl.jena.ontology.*;
9  import com.hp.hpl.jena.rdf.model.*;
10 import com.hp.hpl.jena.reasoner.*;
11 import com.hp.hpl.jena.query.*;
12 import com.hp.hpl.jena.vocabulary.ReasonerVocabulary;
13 import com.hp.hpl.jena.reasoner.rulesys.GenericRuleReasonerFactory;
14 import com.hp.hpl.jena.datatypes.xsd.*;
15 import com.hp.hpl.jena.util.FileManager;
16 import java.io.InputStream;
17 import donnees.actor.*;
18 import donnees.task.*;
19 import donnees.parameter.ClinicalParameter;
20 import donnees.clinicalstatus.*;
21 import donnees.location.Location;
22 import donnees.message.Message;
23 import donnees.parameter.Parameter;
24 import donnees.healthcareorganization.Organization;
25 import donnees.parameter.PreciseClinicalParameter;
26 import donnees.actor.Patient;
27 import donnees.resource.CommunicationResource;
28 import donnees.resource.MaterialResource;
29 import donnees.resource.Ressource;
30 import ressources.TMATypes;
31 import ressources.Md5;

```

```

34
35 public class OwlManager {
36
37     public OwlManager() {
38         model = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM, null );
39         InputStream in =FileManager.get().open( modelURL );
40         model.read (in,null);
41         taille=modelURL.length();
42         runEngine();
43     }
44
45     public void runEngine() {
46         try{
47             configuration = model.createResource();
48             configuration.addProperty(ReasonerVocabulary.PROPruleMode, "hybrid");
49             configuration.addProperty(ReasonerVocabulary.PROPruleSet, rulesURL);
50             reasoner = GenericRuleReasonerFactory.theInstance().create(configuration);
51             infModel = ModelFactory.createInfModel(reasoner, model);
52             infModel.prepare();
53             java.io.OutputStream o=new java.io.FileOutputStream(modTempUrl);
54             infModel.write(o);
55             o.close();
56             dataStorage.copy();
57         }
58         catch(Exception e){
59             e.printStackTrace();
60         }
61     }
62

```



```

63 //Méthodes d'insertion de nouvelles occurrences dans le document OWL
64
65 public String getClassUser(String id){
66     String classe;
67     classe=TMATypes.Cardiologist;
68     try{
69         java.io.InputStream in = new java.io.FileInputStream(new java.io.File(modTempUrl));
70         Model model1 = ModelFactory.createDefaultModel();
71         model1.read(in,null);
72         in.close();
73         String request=
74             "PREFIX onto: <http://www.owl-ontologies.com/OntologyTelemedicineKnowledgeModel.owl#>" +
75             "SELECT ?y " +
76             "WHERE " +
77             "{" +
78             "?y onto:organizationID \""+id+"\" ." +
79             "}";
80         Query query = QueryFactory.create(request);
81         QueryExecution qe = QueryExecutionFactory.create(query, model1);
82         ResultSet results = qe.execSelect();
83         QuerySolution q1;
84         if(!results.hasNext()){
85             in = new java.io.FileInputStream(new java.io.File(modTempUrl));
86             model1 = ModelFactory.createDefaultModel();
87             model1.read(in,null);
88             in.close();
89             request=
90
91             "PREFIX onto: <http://www.owl-ontologies.com/OntologyTelemedicineKnowledgeModel.owl#>" +
92             "SELECT ?y " +
93             "WHERE " +
94             "{" +
95             "?y onto:login \""+id+"\" ." +
96             "}";
97             query = QueryFactory.create(request);
98             qe = QueryExecutionFactory.create(query, model1);
99             results = qe.execSelect();
100             q1=results.nextSolution();
101             Resource r=model1.getResource(modelURI+q1.get("y").toString().substring(taille));
102             String type=r.toString();
103             System.out.println("type de l'objet : "+ type);
104             Individual v=model1.getIndividual(type);
105             if(v.getOntClass().toString().substring(taille).compareTo(TMATypes.Cardiologist)==0){ // la methode
106                 System.out.println("L'utilisateur est un cardiologue");
107             }
108             if(v.getOntClass().toString().substring(taille).compareTo(TMATypes.Emergency_Physician)==0){
109                 System.out.println("L'utilisateur est un urgentiste");
110                 classe=TMATypes.Emergency_Physician;
111             }
112             if(v.getOntClass().toString().substring(taille).compareTo(TMATypes.General_Practitioner)==0){
113                 System.out.println("L'utilisateur est un généraliste");
114                 classe=TMATypes.General_Practitioner;
115             }
116             if(v.getOntClass().toString().substring(taille).compareTo(TMATypes.Nurse)==0){
117                 System.out.println("L'utilisateur est une infirmière");
118                 classe=TMATypes.Nurse;
119             }
120         }
121     }
122 }

```

```

119         if(v.getOntClass().toString().substring(taille).compareTo(TMATypes.Rescue_Team_Member)==0){
120             System.out.println("L'utilisateur est un membre d'équipe d'urgence");
121             classe=TMATypes.Rescue_Team_Member;
122         }
123     }
124     else{
125         q1=results.nextSolution();
126         classe=TMATypes.Organization;
127         System.out.println(q1.toString());
128     }
129 }
130 catch(Exception e){
131     e.printStackTrace();
132 }
133 return classe;
134 }

```

```

222 + public Actor getActor(String id){...}
297
298 + public void insertMaterialActor(MaterialActor mActor,String type){...}
377
378 + public void insertClinicalStatus(ClinicalStatus clinicalStatus, String type){...}
440
441 + public ClinicalStatus getClinicalStatus(String statusId){...}
508
509 + public String insertLocation(Location location){...}
549
550 + public Location getLocation(String name){...}
580
581 + public String insertMessage(Message message){...}
681
682 + public Message getMessage(String id){...}
708
709 + public void insertOrganization(Organization organization){...}
759
760 + public Organization getOrganization(String id){...}
837
838 + public Vector<Organization> getOrganizations(){...}
851
852 + public void insertPreciseClinicalParameter(PreciseClinicalParameter parameter,String type){...}
888
889 + public void insertDyspnea(ClinicalParameter parameter){...}
921
922 + public void insertAccidentParameter(Parameter parameter, String type){...}
957
958 + public ClinicalParameter getClinicalParameter(String id){...}

```

```

1007 + public static Date stringToDate(String sDate, String sFormat) throws Exception {...}
1011
1012 + public void insertPatient(Patient patient) {...}
1055
1056 + public String insertCommunicationResource(CommunicationResource ressource,String type) {...}
1124
1125 + public String insertMaterialRessource(MaterialResource ressource,String type) {...}
1189
1190 + public Ressource getResource(String id) {...}
1271
1272 + public void insertTelemedecineTask(TelemedecineTask task,String type) {...}
1367
1368 + public Vector<Actor> getActors() {...}
1436
1437 + public Resource getConfiguration() {...}
1440
1441 + public void setConfiguration(Resource configuration) {...}
1444
1445 + public InfModel getInfModel() {...}
1448
1449 + public void setInfModel(InfModel infModel) {...}
1452
1453 + public Model getModel() {...}
1456
1457 + public void setModel(OntModel model) {...}
1460
1461 + public Reasoner getReasoner() {...}
1464
1465 + public void setReasoner(Reasoner reasoner) {...}

1461 + public Reasoner getReasoner() {...}
1464
1465 + public void setReasoner(Reasoner reasoner) {...}
1468
1469
1470 //Attributs de la classe owl manager :
1471 private static final String modelURI="http://www.owl-ontologies.com/OntologyTelemedicineKnowledgeModel.owl#";
1472 //private static final String modelURI="http://localhost:8080/ApplicationWebTelemedecine/TelemedicineKnowledgeModel.owl";
1473 private static final String modelURI="C:\\Stage\\ApplicationWebTelemedecine\\src\\java\\ressources\\TelemedicineKnowledgeModel.owl";
1474
1475 //private static final String modelURI="ftp://utilisateur:admin/194.167.219.180@ApplicationWebTelemedecine/TelemedicineKnowledgeModel.owl";
1476 private static final String rulesURI="./ressources/TemporaryRessource/rules_list.rules";
1477 private static final String modTempUri="C:\\Stage\\ApplicationWebTelemedecine\\src\\java\\ressources\\TemporaryRessource\\TelemedicineKnowledgeModel.owl";
1478 private int taille=0;
1479 private Reasoner reasoner;
1480 private OntModel model;
1481 private Resource configuration;
1482 private InfModel infModel;
1483 private OwlSavingData dataStorage=new OwlSavingData();

```



## FOLIO ADMINISTRATIF

### THESE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

NOM : NAGEBA

DATE de SOUTENANCE : Le 07 Décembre 2011

Prénoms : Ebrahim

TITRE : Modèle d'architecture personnalisable pour l'optimisation de l'accès à des ressources et services pervasifs. Application à la télémédecine.

NATURE : Doctorat

Numéro d'ordre : 2011-ISAL-0128

Ecole doctorale : InfoMaths – Informatique et Mathématiques

Spécialité : Informatique

#### RESUME :

Le développement et l'usage croissants de systèmes pervasifs, dotés de fonctionnalités et de moyens de communication de plus en plus sophistiqués, offrent de fantastiques potentialités de services, en particulier pour l'e-Santé et la télémédecine, au bénéfice de tout citoyen, patient ou professionnel de santé. L'un des challenges sociétaux actuels est de permettre une meilleure exploitation des services disponibles pour l'ensemble des acteurs impliqués dans un domaine donné. Mais la multiplicité des services offerts, la diversité fonctionnelle des systèmes, et l'hétérogénéité des besoins nécessitent l'élaboration de modèles de connaissances de ces services, des fonctions de ces systèmes et des besoins. En outre, l'hétérogénéité des environnements informatiques distribués, la disponibilité et les capacités potentielles des diverses ressources humaines et matérielles (instrumentation, services, sources de données, etc.) requises par les différentes tâches et processus, la variété des services qui fournissent des données aux utilisateurs, et les conflits d'interopérabilité entre schémas et sources de données sont autant de problématiques que nous avons à considérer au cours de nos travaux de recherche. Notre contribution vise à optimiser la qualité de services en environnement ambiant et à réaliser une exploitation intelligente de ressources ubiquitaires. Pour cela, nous proposons un méta-modèle de connaissances des principaux concepts à prendre en compte en environnement pervasif : Acteur, Tâche, Ressource, Objet, Service, Localisation, Organisation, etc. Ce méta-modèle est basé sur des ontologies décrivant les différentes entités précitées dans un domaine donné ainsi que leurs relations. Puis, nous l'avons formalisé en utilisant un langage standard de description des connaissances. A partir de ce modèle, nous proposons alors une nouvelle méthodologie de construction d'un framework architectural, que nous avons appelé ONOF-PAS (ONtology Oriented Framework for Pervasive Applications and Services). ONOF-PAS est basé sur des modèles ontologiques, une base de règles, un moteur d'inférence, et des composants orientés objet permettant la gestion des différentes tâches et le traitement des ressources. Il s'agit d'une architecture générique, applicable à différents domaines. ONOF-PAS a la capacité d'effectuer un raisonnement à base de règles pour gérer les différents contextes d'utilisation et aider à la prise de décision dans des environnements hétérogènes dynamiques, tout en tenant compte de la disponibilité et de la capacité des ressources humaines et matérielles requises par les diverses tâches et processus exécutés par des systèmes d'information pervasifs. Enfin, nous avons instancié ONOF-PAS dans le domaine de la télémédecine pour traiter le scénario de l'orientation des patients ou de personnes victimes de problèmes de santé en environnement hostile telles que la haute montagne ou des zones géographiquement isolées. Un prototype d'implémentation de ces scénarios, appelé T-TROIE (Telemedicine Tasks and Resources Ontologies for Inimical Environments), a été développé afin de valider notre approche et le framework ONOF-PAS que nous avons proposé. Le résultat attendu de notre contribution est d'aboutir à une communication efficace des informations disponibles, procurant une qualité de services de télémédecine qui permet d'assurer une réponse prompte et fiable à chaque acteur demandeur en lui fournissant des données exhaustives et pertinentes à sa demande, et par conséquence, d'obtenir une prise en charge appropriée, plus rapide, et optimale des patients, quel que soit l'endroit où ils se trouvent, en fonction des moyens et des technologies disponibles.

#### MOTS-CLES :

Systèmes Pervasifs, Context-Awareness, Modélisation des Connaissances, Ontologie, Raisonnement à base de règles, Systèmes d'aide à la décision, Télémédecine, eSanté.

Laboratoire (s) de recherche : SFR Santé Lyon Est UMS3453 CNRS / US7 INSERM – plateforme eTechSanté

Directeur de thèse: Prof. Paul RUBEL

Co-directeur : Jocelyne FAYN, HDR

Président de jury : Prof. Jacques DUCHENE

Composition du jury :

Prof. Jacques DUCHENE, Dr. Jocelyne FAYN, Prof. Franck MORVAN, Prof. Paul RUBEL, et Prof. Christine VERDIER

Ebrahim NAGEBA

Thesis in Computer Sciences / 2011.

Institut National des Sciences Appliquées de Lyon

Cette thèse est accessible à l'adresse : <http://theses.insa-lyon.fr/publication/2011ISAL0128/these.pdf>

© [E. Nageba], [2011], INSA de Lyon, tous droits réservés