O'Neurolog – Building an Ontology for Neurology in Mobile Environment

Youssouf EL ALLIOUI† and Omar EL BEQQALI††

†, ††Computer Science Department, Faculty of Sciences Dhar el mahraz University Sidi Mohammed Ben Abdellah, Fez, 30000 Morocco

Summary
The (context-aware services). A crucial requirement for the context-aware service provisioning is the dynamic retrieval and interaction with local resources, i.e., resource discovery. The high degree of dynamicity and heterogeneity of mobile environments requires to rethink and/or extend traditional discovery solutions to support more intelligent service search and retrieval, personalized to user context conditions. Several research efforts have recently emerged in the field of service discovery that, based on semantic data representation and technologies, allow flexible matching between user requirements and service capabilities in open and dynamic deployment scenarios. Our research work aims at providing suitable answering mechanisms of mobile requests by taking into account user contexts (preferences, profiles, physical location, temporal information…).
This paper proposes an ontology, culled O'Neurolog, to capture semantic knowledge a valuable in Neurology domain in order to assist users (doctor, patient, administration …) when querying Neurology knowledge bases in mobile environment.
increasing diffusion of portable devices with wireless connectivity enables new pervasive scenarios, where users require tailored service access according to their needs, position, and execution/environment conditions
Key words:
Neurology, ontology, context-aware, semantic web, query answering, mobile environment

1. Introduction

1.1 General context

In context-aware information provisioning scenarios, it is crucial to enable the dynamic retrieval of available knowledges in the nearby of the user's current point of attachment, while minimizing user involvement in information selection. Data and knowledge discovery in pervasive environments, however, is a complex task as it requires to face several technical challenges at the state of the art, such as user/device mobility, variations (possibly unpredictable) in service availability and environment conditions, and terminal heterogeneity.
Users might need to discover knowledges whose names and specific implementation attributes cannot be known in advance, while data providers need to use several and different terms or keywords and whose technical capabilities and conditions at interaction time might be mostly unpredictable beforehand.
In medical domain, there is a great need for using mobile devices to access and retrieve neurological data concerning a patient by physicians or interested organisms (insurance, emergency …). Neurological information is available via web pages, stored in ftp sites or relational databases, and textually described in publications. Neurology (from Greek, neuron, "nerve"; and logia, "study") is a medical specialty dealing with disorders of the nervous system. Specifically, it deals with the diagnosis and treatment of all categories of disease involving the central, peripheral, and autonomic nervous systems, including their coverings, blood vessels, and all effectors tissue, such as muscle.[1] The corresponding surgical specialty is neurosurgery. A neurologist is a physician who specializes in neurology, and is trained to investigate, or diagnose and treat neurological disorders. Pediatric neurologists treat neurological disease in children. Neurologists may also be involved in clinical research, clinical trials, as well as basic research and translational research.[2]
However, mobile search engines are unable to answer questions about this massive neurological knowledge base other than identifying resources that contain some subset of the specified attributes. The main reason for this limitation is that the representation of biological information on the web is not machine understandable, in the sense that computers cannot interpret words, sentences or diagrams so as to correctly reason about the objects and the relations between them that are implicitly stated in those documents [3]. The primary goal of the semantic web is to add semantics to the current Web, by designing ontologies which explicitly describe and relate objects using formal, logic-based representations that a machine can understand and process [4]. This ongoing effort is expected to facilitate data representation, integration and question answering, of critical importance in the life sciences and hospital information system (HIS).
Therefore, returned answers scope needs to be filtered according to finer-grained criteria other than administrative or network grouping. All and only those data that are semantically compatible with the user's context should be automatically and transparently made visible to him. The exploitation of user's

Manuscript received September 5, 2010
Manuscript revised September 20, 2010

context-awareness in knowledge discovery helps mobile clients saving time and efforts in information retrieval.
On the other hand, the potential of semantic-based discovery has not been fully exploited yet because of various management issues, which seem to be still open. Access terminals usually exhibit relevant differences in resource capabilities, such as display size and resolution, computing power, memory, network bandwidth, and battery. A crucial management crucial issue remains how to provide support for semantic-based discovery to mobile devices with limited capabilities. Semantic support services, e.g., ontology repositories, inference engines and knowledge management tools, typically require a large amount of computational/memory resources that may not fit the properties of mobile devices. In particular, strict limitations exist about the kind of semantic support facilities that can be hosted on resource-constrained devices. For example, executing a reasoning process on board of a resource-limited device, such as a smart phone, might not only consume battery, but more important, it would probably monopolize all available memory resources, thus making the execution of other applications very difficult.

1.2 Runing example

Let us suppose a physician who needs to consult a patient's clinical data in order to set a proper treatment for him. If the healthcare act is taking place inside the hospital, the doctor will be allowed to access the Hospital Information System (HIS) and to retrieve all the patient's Electronic Health Records (EHRs). Having enough time and knowledge – and depending on the usability of the software system – the specialist will rule out all the useless pieces of information and will get the ones he is interested in.
In the latter situation, a brief report including those pieces of the patient's clinical data which ought to be considered would be very valuable. The clinical procedure which is going to be carried out would determine which data should be part of this summary. For example, is an another member of the patient's family has the same symptoms yet. If true the physician could propose muscular or hepatic biopsy and so realize lumbar punctures, cerebral scanner or IRM.
The patient does not walk, does not stand nor sitting does not his head. He/she did not speak, but knows how to understand, she established contacts with people who are very familiar.
We think that by joining context to domain knowledge could help improving the summarization of research results for a mobile user. This activity is called personalization and implies recommendations.

1.3 Paper organization

The remaining of this document is structured as follows. Section 2 presents related works about semantic and user-centric information. Section 3 highlights e general architecture for our application. In Section 4, we give an overview of the semantic model that we have proposed to capture neurological knowledges in pervasive environment. Section 5 is devoted to building O'Neurolog with protégé-2000, a well-known ontology editor. Finally, in Section 6 some conclusions and directions for future work are pointed out.

2. Related work

There are several research streams in personalized recommendation. One stream aims at improving the accuracy of algorithms used in recommendation systems [5]. The second stream is focused on the interaction between a recommendation system and its customers. For instance, some studies investigated the persuasive effect of recommendation messages [7], developed better data collection mechanisms [8], and enhanced awareness about privacy issues [9]. Furthermore, a few studies focused on the effect of moderating factors such as user characteristics and product features on the performance of recommendation [10].
A typical personalization process includes three steps: understanding customers through profile building, delivering personalized offering based on the knowledge about the product and the customer, and measuring personalization impact [11]. Montaner, et al. [12] simplified the process into two stages after analyzing 37 different systems: profile generation and maintenance, and profile exploitation.
One key to the performance of personalized recommendation is the nature of the mechanism it uses to build customer profiles. In previous research, a number of different algorithms have been proposed. These systems are classified based on various characteristics. For example, Beyah et al. [13] divided recommendation systems into four types: collaborative filtering (people to people correlations), social filtering, non-personalized recommendation systems, and attribute-based recommendations. Wei et al. [14] classified recommendation systems into six approaches based on the type of data and technique used to arrive at recommendation decisions. These approaches are recommendations based on: popularity, content, association, demographics, reputation and collaboration.
Schafer et al. [15] argues that recommendation systems may be categorized by the data they use, which includes original data search, expert ratings, statistical rankings, content-based recommendation, object associations, and

user associations. A system using the original data search does not analyze user profiles. Rather, it provides a flexible user interface and allows users to query the database directly. Expert ratings use comments or ratings from experts in the domain (e.g., music or movies) and make recommendations accordingly. Statistical ranking is a simple but popular method, which uses descriptive statistical data such as order frequency to rank different objects for recommendation. Content-based recommendation uses attributes of the content to match user interest profiles. Object associations use found relationships among objects to make recommendations. A popular example is the market basket analysis that finds items often ordered simultaneously by a customer.

There exist three popular methods for extracting user preferences: direct, semi-direct, and indirect extraction [16]. The direct approach asks the user to tell the system explicitly what he prefers. For instance, a knowledge management system may list all document categories and ask the user to check those of interest to him. The semi-direct approach asks the user to rate all documents he has read and gains knowledge of user preference through these ratings [17]. The indirect approach captures user preference from browsing behavior recorded by the computer, such as hyperlink clicks [18] or time spent on reading a document [19].

For making recommendations, two approaches are popular: collaborative filtering and content-based filtering [20]. Collaborative filtering is an approach to making recommendations by finding correlations among the shared likes and dislikes of the system users. It is capable of finding items of potential interest from ratings of previous users. Content-based filtering makes recommendation by analyzing the items rated by the user and the item to be recommended [21]. Generally speaking, content-based information filtering has proven to be effective in locating textual documents relevant to a topic [22]; whereas collaborative filtering is popular with e-tailors that sell physical products such as in Amazon.com [23]. The integration of these two types of filtering is reported to exhibit good performance in some domains (e.g., [5]).

Recently, with the advent of semantic technologies, personalization has received more attention based on exploitation of domain ontologies. Liang et al [37] adopts a semantic-expansion approach to build the user profile by analyzing documents previously read by the person. Once the customer profile is constructed, personalized contents can be provided by the system. An empirical study using master theses in the National Central library in Taiwan shows that the semantic-expansion approach outperforms the traditional keyword approach in catching user interests. The proper usage of this technology can increase customer satisfaction.

Personalized recommendations are applied to both physical and digital information products. Recent application domains include books [23], news [24], movies [25], advertisements [24], one-to-one marketing campaigns [6], and bundle purchases [26]. Although Amazon.com has been applauded for its success in using personalized recommendation, information goods such as news and documents in digital libraries are popular for personalization on the Internet due to its nature of content modularity.

The application of personalized recommendation to information goods also has a long history. For example, Mock and Vemuri [27] developed the Intelligent News Filtering Organization System (INFOS) that reorganizes the order of news based on revealed preferences. The results of a pilot test show that INFOS can effectively reduce the reader's search load. Sakagami and Kamba [16] developed the ANATAGONOMY system that learns reading preferences from the browsing behavior of a user; a learning engine and a scoring engine produce personalized Web news. Billsus and Pazzani [28] designed an intelligent news agent to automatically learn the user's preferences and provide personal news. Mooney and Roy [22] proposed a content-based book recommending system that utilizes information extraction for text categorization and produces accurate recommendations. Lai et al. [29] designed a news recommendation system based on customer profiles to provided customized Internet news. Fan, et al. [30] presented a method for generating profiles of news readers. Liang, et al. [31] also reported experimental findings that personalized services produced significantly higher user satisfaction for online news readers.

3. General architecture

In this section, we further describe the components in the application are listed as follows (Fig.1).

The Metadata Manager (MM) provides templates to support the user in the task of specifying user/device/service profiles. The use of templates ensures that metadata are encoded in the correct format, i.e., compliant to O'profile ontology (see Fig. 2), and offers to non technical users a friendly interface for profile specification. MM does not perform semantic reasoning, but only syntactic compliance checking.

The Discovery Manager (DM) is responsible for determining and maintaining the list of returned answers that are accessible to the mobile user according to her context.

The Context Manager (CM) is responsible for creating user contexts when new users initiate their discovery sessions, for monitoring changes in both created user contexts and in relevant external environment conditions, e.g., the addition of new tuples of informations, for notifying changes to interested entities, and for updating user contexts accordingly.

The Query Processor (QP) is in charge of collecting and processing user requests for neurological data. QP interacts with the user to specify required informations capabilities and user preferences. In particular, QPE translates value preferences expressed by the user at access request time into property restrictions.

The Profile Matching Engine (PME) is responsible for performing a matching algorithm between user/device requirements and answers capabilities, taking user preferences into account. PME is requested to perform its algorithm in two cases, i.e., when DM needs to determine the list of visible services for each user (i.e., the information whose profile is semantically compatible with user/device profile) and when QP needs to resolve a specific user query.

In Fig. 2 we further describe the components in the Process query:

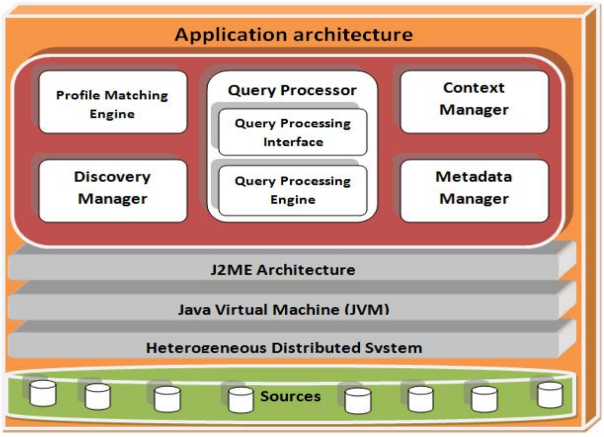

Figure 1 : Application architecture

Query reformulation: After a user issues a Query request, this query request is sent into the mediator where the query reformulation component in the mediator receives it. The query reformulation component passes a reasoning request to the inference engine and accesses the O'Neurolog in order to find out the relationships that are implicit in the user original Query request.

Query decomposition: After the query decomposition receives the reformulated query, the reformulated query is decomposed with the assistance of the mapping. So query decomposition could use such information to decompose the query into several sub-queries that are respectively applicable to their target information sources.

Query translation: The wrapper receives the corresponding sub-queries. The wrapper translates the generic sub-queries into native queries. Afterwards, such native queries are issued to the corresponding source in order to find out the data.

Result packaging: It is a one-on-one mapping between a wrapper and a type. The wrapper packages the results into a normal form.

Result composition: The query composition component collects several packaged results sent back from several wrappers according to the previous decomposed result.

Answer reduction: The number of combined answers will be reduced by exploiting inference rules available in O'profile. The aim is to take into account limited characteristics of mobile devices (weak bandwidth, cache memory, reduced screen) when processing queries and displaying answers.

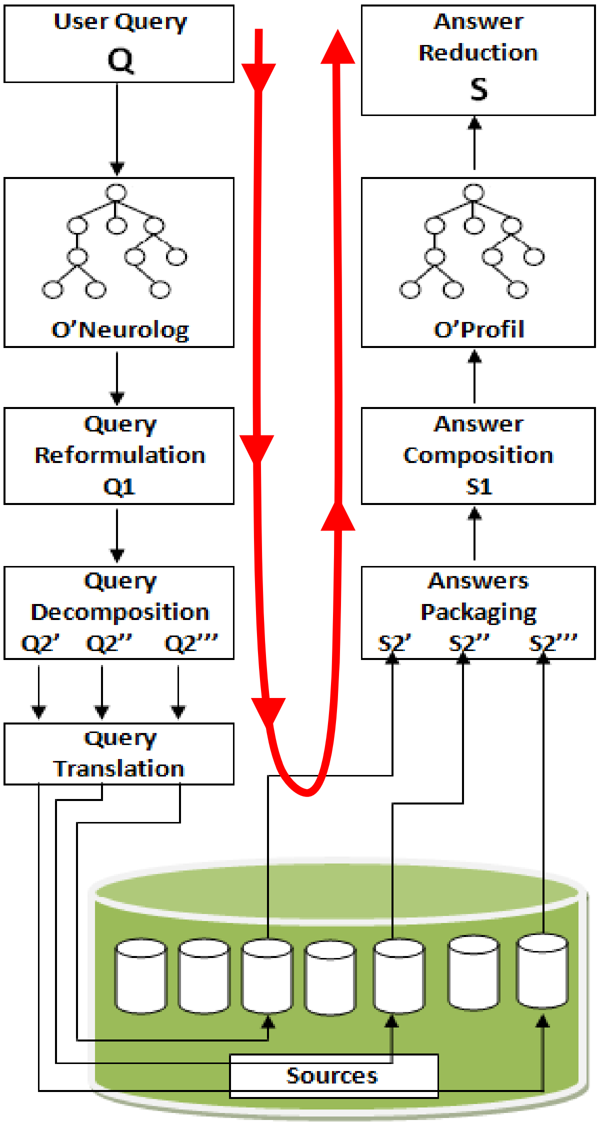

Figure 2 : Process Query

4. Preliminaries

4.1 The semantic model and Formalization of User contexts

The semantic model is based on the use of O'Neurolog, an Ontology of Neurology science, and a formal specification of user contexts. Consequently, two different kinds of knowledge are to be managed by such mobile system :

- Domain knowledge about the neurology sciences.
- Context knowledge about the users where the domain knowledge will be used; for our doctor, this would be a vocabulary to briefly describe the situation of the patient he is going to attend.

In this work we will use ontologies to materialize our model since they have been remarked to be a suitable formalism to build a Knowledge Base including context knowledge.

Ontologies are defined as ‘‘formal, explicit specifications of a shared conceptualization’’ [32], encode machine-interpretable descriptions of the concepts and the relations in a domain using abstractions as class, role or instance, which are qualified using logical axioms. Properties and semantics of ontology constructs are determined by Description Logics (DLs) [33], a family of logics for representing structured knowledge which have proved to be very useful as ontology languages.

Context is a complex notion that has many definitions. Here, we consider context as any key information characterizing the user, e.g., user preferences, needs, location, and any useful information about the environment where he operates, e.g., date, time, on-going activities and interactions with services and/or other users.

Let us give some basic definitions to more limit the frontiers of our study:

Personalization is defined as "the ability to provide content and services tailored to individuals based on knowledge about their preferences and behavior" or "the use of technology and customer information to tailor electronic commerce interactions between a business and each individual customer" [11]. A major vehicle that makes personalization possible is the recommendation system that matches potential products with customer preferences.

A recommendation system is a computer-based system that uses profiles built from past usage behavior to provide relevant recommendations. For instance, a video rental store may analyze historical rental data to recommend new movies to individual customers. An online newspaper may customize news services to different readers based on their reading interests. The objective of a recommendation system is to retrieve information of interest to users from a large repository. Such systems can reduce the search efforts of users and mitigate the encumbrance of information overload [31].

In order to overcome user-context requirements, we focused on a preference-driven formalization:

In fact, mobile clients are described in terms of profiles and preferences. User profiles are composed of both dynamic and static metadata. Dynamic properties include, for example, user locality and state, while static properties are grouped into three categories (analogously to the case of a service): identification, capabilities and requirements. Identification information, such as an ID code, a string or an URL, is used to name and identify the user. Capabilities represent the user's abilities and capacities, such as speaking a language. User requirements describe user-defined constraints that must be always satisfied during service provisioning. Fig. 2 shows an example of user static profile.

```
<rdf:RDF>
<profile:User rdf:ID="Alice">
   <profile:hasProfile>
     <profile:Profile rdf:ID="Alice_Profile">
       <profile:profile_id>
          <id:Name
          rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
          Alice Brown</id:Name>
       </profile:profile_id>
       <profile:profile_cap>
          <user_cap:LanguageCapability rdf:ID="LanguageCap_1">
            <user_cap:speaks rdf:resource="&language-ont;English"/>
            <user_cap:speaks rdf:resource="&language-ont;Italian"/>
          </user_cap:LanguageCapability>
       </profile:profile_cap>
       <profile:profile_req>
          <user_cap:Requirement rdf:ID="Requirement_1">
            <profile:requires>...</profile:requires>
          </user_cap:Requirement>
       </profile:profile_req>
     </profile:Profile>
   </profile:hasProfile>
 </profile:User>
</rdf:RDF>
```

Figure 3 : Static user profile specification

Preferences allow the user to express desired choices about search results and to relax constraints on resources requests [34].

The semantic model defines two types of preference: value preferences and priority preferences.

Value preferences specify the preferred value for a resource property. Depending on the property value type, a preference might be either object or data type. Value preferences are modeled using an ontological approach and expressed in OWL. In particular, object preferences are modeled by means of the OWL ObjectProperty construct, while data type preferences are modeled by means of the OWL DatatypeProperty construct. The basic preference ontology is represented in Fig. 4 [35]. As shown in the diagram, a value preference defines the desired property value for a given capability, one or more explicit alternative values, and the kind of acceptable constraint relaxation. Let us note that, in case of a datatype preference, such relaxation defines an interval of acceptable values, based on the preferred numerical value, while in case of an object preference.

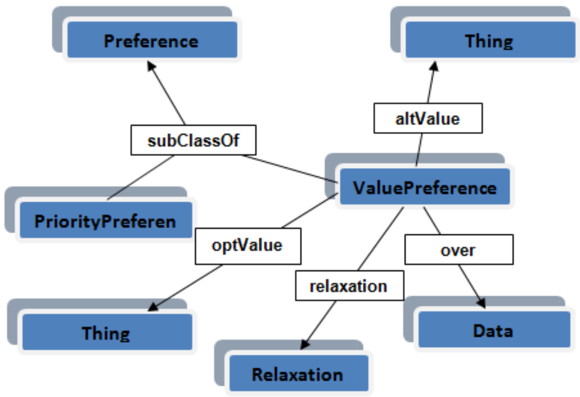

Figure 4 : Basic preference ontology

We define three possible semantic relations between the ideal value expressed in the request and other acceptable values, namely: exact (default), plug-in, subsumes [36]. "Exact" represents the case of a perfect match between required and offered capability for the property constrained by the preference. In the "plug-in" case, request subsumes offer, therefore the user may consider acceptable a service providing a more specific value than the requested one. For example, a user looking for a service providing news might accept a service providing news only provided by newspapers. In the "subsumes" case, offer subsumes request, hence the user may decide to accept a service whose capability is more general than the requested one.
Since a request for service may include several capabilities and a capability might have multiple properties, semantic model allows the user to define a priority order amongst capabilities/properties by means of priority preferences. For example, let us consider the capability of providing news. If a preference assigns higher priority to "language" rather than "topics", the former will be tested first when performing matching and a service with a good value for this property will be considered more compatible than another having a better value on the "topics" capability.

4.2 Neurology

The field of computer consultation has passed through three historical phases. In the first, attempts were made to improve on human diagnostic performance by rejecting the methods used by clinicians and substituting various statistical techniques [38]. Statistical methods proved to be accurate for small diagnostic domains, but impractical for application to realworld problems [39]. In the second phase, it was recognized that human problem solving methods were deceptively powerful [40] and attempts were made to capture diagnostic logics as fixed decision protocols [41]. Although these met with success in some areas, it was recognized that such protocols suffered from inflexibility [42]. At present, efforts are directed towards building systems which incorporate expert problem solving strategies, but which retain flexibility - 'artificial intelligence' systems [43].
Neurology is a medical specialty concerned with the diagnosis and treatment of all categories of disease involving the central, peripheral, and autonomic nervous systems, including their coverings, blood vessels, and all effector tissue, such as muscle. [44]
Neurological disorders are disorders that can affect the central nervous system (brain and spinal cord), the peripheral nervous system, or the autonomic nervous system.
Conditions can include but are not limited to:

- Brain injury, spinal cord and peripheral nerves
- Cerebral palsy
- Cerebrovascular disease, such as transient ischemic attack and stroke.
- Epilepsy
- Headache disorders such as migraine, cluster headache and tension headache
- Infections of the brain (encephalitis), brain meninges (meningitis), spinal cord (myelitis)
- Infections of the peripheral nervous system
- Movement disorders such as Parkinson's disease, Huntington's disease, hemiballismus, tic disorder, and Gilles de la Tourette syndrome
- Sleep disorders

4.3 Noy's approach - driven Ontology design

All ontologies were created using OWL language and Protégé Ontology Editor
Proposals to design ontologies from the theoretical and practical point of view can be found in [45] and they include brain storming, scope definition, motivating scenarios, competency questions, creation of bag of concepts, creating concept hierarchy, properties, domain and range, amongst others. Authors of these papers agree in the fact that there is no a correct methodology to design an ontology, but they also agree in seeing this process iterative and application oriented. In the ontology design process described in this paper their proposed criteria is reused when possible and mapped to the OWL language.
The O'Neurolog ontology was designed following semantic web best practices [46] and implemented using the OWL-DL. Briefly, we follow the methodology described by [45], but apply a three layer ontology design with increasing expressivity [47].
The ontology design process can be summarized in the following steps:

- Define the scope and requirements (use cases) of the ontology.
- Determine essential concepts, reusing existing ontologies when possible.

- Construct a primitive hierarchy of concepts.
- Map to an upper level ontology.
- Assign relations between concepts and attributes.
- Design a second layer of complex definitions that imposes additional logical restrictions and establishes necessary and/or necessary and sufficient conditions.
- Repeat steps 2–6 until requirements are satisfied from semantic query answering over populated ontologies.

Step 1. Define the scope and requirements of the ontology : The scope of the O'Neurolog project was to model the entities and their relations found in the neurology such that one could effectively search this knowledge base using expected relations rather than with a collection of keywords.
Step 2. Determine essential concepts, reusing existing ontologies when possible : An initial set of concepts was manually collected to represent the type and attributes of the data located in the data files. Concept definitions were obtained from the neurology science glossary, WordNet and Wikipedia and were manually added to the class using the "rdfs:comment" annotation property for human consumption.
Step 3. Construct a primitive hierarchy of concepts : A hierarchy of concepts for the O'Neurolog primitive ontology was developed by iteratively categorizing the set of necessary classes.
Step 4. Map to an upper level ontology : Upper level ontologies promise increased semantic coherency by helping to identify the basic type of domain entities and imposing restrictions on the relationships that these entities may hold. An upper level ontology should be carefully selected according to the purpose and scope of the ontology designed.
Step 5. Assign relations between concepts and attributes : An essential part of the modeling process is to establish which relations exist between the set of concepts. To do this, we need two things: (1) a set of basic relations to draw from and (2) generate a kind of topics map.
Step 6. Design a second layer of complex definitions that imposes additional logical restrictions and establishes necessary and/or necessary and sufficient conditions. The primitive hierarchy of concepts was augmented and refined with logical restrictions to reflect knowledge.. To ensure the proper classification of data and knowledge discovery, necessary and sufficient conditions were added to the ontology. Necessary conditions were added where obvious. Necessary and sufficient conditions were added for single property varying entities or in the design of a value partition.

5 Building O'Neurolog with Protégé-2000

5.1 Presentation of Protégé-2000 [48]

Protégé-2000 is an integrated software tool used by system developers and domain experts to develop knowledge-based systems. Applications developed with Protégé-2000 are used in problem-solving and decision-making in a particular domain.
While our earlier Protégé/Win, like a classical database system, separately defined classes of information (schema) and stored instances of these classes, Protégé-2000 makes it easier to work simultaneously with both classes and instances. Thus, a singular instance can be used on the level of a class definition, and a class can be stored as an instance. Similarly, slots, which earlier were employed only within classes, are now elevated to the same level as classes. With this new knowledge model we also facilitate conformance to the Open Knowledge Base Connectivity (OKBC) protocol for accessing knowledge bases stored in knowledge representation systems. Finally, applications on top of these components are also executed within the integrated Protégé-2000 environment.
The Protégé-2000 tool accesses all of these parts through a uniform GUI (graphical user interface) whose top-level consists of overlapping tabs for compact presentation of the parts and for convenient co-editing between them. This "tabbed" top-level design permits an integration of (1) the modeling of an ontology of classes describing a particular subject, (2) the creation of a knowledge-acquisition tool for collecting knowledge, (3) the entering of specific instances of data and creation of a knowledge base, and (4) the execution of applications. The ontology defines the set of concepts and their relationships. The knowledge-acquisition tool is designed to be domain-specific, allowing domain experts to easily and naturally enter their knowledge of the area. The resulting knowledge base can then be used with a problem-solving method to answer questions and solve problems regarding the domain. Finally, an application is the end product created when the knowledge base is used in solving an end-user problem employing appropriate problem-solving, expert-system, or decision-support methods.
The main assumption of Protégé-2000 is that knowledge-based systems are usually very expensive to build and maintain. For example, the expectation is that knowledge-based system development is a team effort, including both developers and domain experts who may have less familiarity with computer software. Protégé-2000 is designed to guide developers and domain experts through the process of system development. Protégé-2000 is designed to allow developers to reuse domain ontologies and problem-solving methods, thereby shortening the time needed for development and program maintenance. Several applications can use the same

domain ontology to solve different problems, and the same problem-solving method can be used with different ontologies.
Protégé-2000 is currently being used in clinical medicine and the biomedical sciences, although it can be used in any field where the concepts can be modeled as a class hierarchy.

5.2 O'Neurolog's screenshots

According to this ontology (see Fig. 5), the support of a comatose patient is based on the following steps:

- Immediate measures which are consisting to maintain vital functions
- Look for and treat shock
- Find and treat the cause of diagnosis easy
- Clinical examination of the patient
- Request additional tests depending on the orientation CLQ
- etiological treatment
- Monitoring

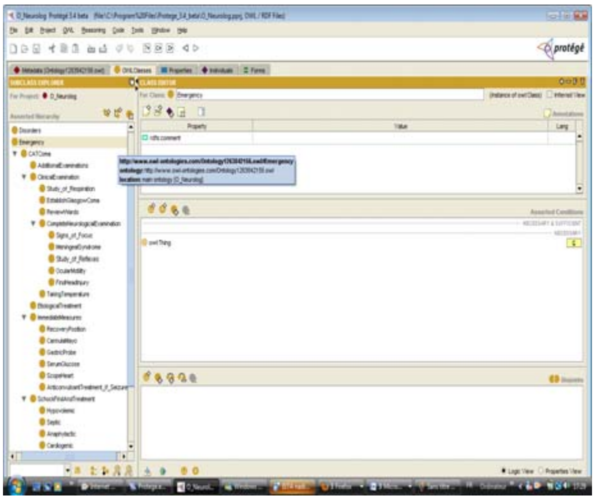

Figure 5 : O'Neurolog ontology

For example, immediate measures (Fig. 6) can be distinguished:

- Recovery position
- Cannula Mayo
- gastric probe
- Oxygen: nasal intubation-mechanical ventilation (depending on the depth of coma)
- Two paths of large caliber venous glucose + serum glucose + Vitamin B1 (chronic alcoholic)
- Scope heart + heart monitoring Tension

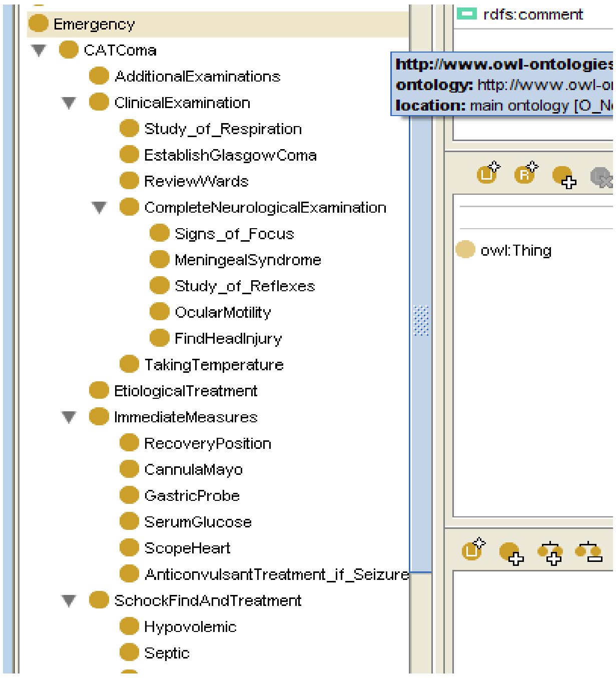

Figure 6 : Example of cathegory "immediate measures"

6. Conclusion and perspectives

In this work, we described a first approach to describe logically, integrate and query neurological data from biological databases using the OWL-DL ontology language. Several features make this work unique. First, we designed a domain specific ontology, called O'Neurolog that incorporates concepts drawn from raw data and expert knowledge.
We described the methodology used for semantically augmenting biological databases in absence of domain specific ontologies and discuss its scope and characteristics. Our methodology is based on Noy's approach.
However, significant challenges remain in realizing the potential of the semantic web, such as the automated creation and population of ontologies, the efficient storage and reasoning with large amounts of ontological data for reasoning and the development of intuitive interfaces among others. In fact, Data and Knowledge Discovery is a crucial activity in pervasive environments where mobile users need to be provided with personalized results due to limited physical characteristics of portable devices. Semantic languages seem to represent a suitable means to realize advanced discovery solutions that overcome the intrinsic limitations of traditional models.
As future work, we plan to provide suitable and relevant answering mechanisms of mobile requests, including

user's context and personalization items. Another interesting future issue we envision to deal with is the resolution of conflicts that may arise between value or priority preferences. We believe that a possible approach may be the definition of meta-preferences, as we have began the formalization in section 3.

Acknowledgment

The authors would like to express their cordial thanks to Dr. Mitsuo Ohta for his valuable advice.

References

[1] http://www.acgme.org/acWebsite/downloads/RRC_progReq/180neurology07012007.pdf . Last see 11/04/2010.

[2] http://en.wikipedia.org/wiki/Neurology. Last see 11/04/2010.

[3] Robu I, Robu V, Thirion B. An introduction to the Semantic Web for health sciences librarians. J Med Libr Assoc 2006;94(2):198–205.

[4] Berners-Lee T, Hall W, Hendler J, Shadbolt N, Weitzner DJ. Computer science. Creating a science of the Web. Science 2006;313(5788):769–71.

[5] H.J. Ahn, Utilizing popularity characteristics for product recommendation, International Journal of Electronic Commerce 11 (2) (2006) 59–80.

[6] S.S. Weng, M.J. Liu, Feature-based recommendation for one-toone marketing, Expert Systems with Applications 26 (4) (2004) 493–508.

[7] S.N. Singh, N.P. Dalal, Web home pages as advertisements, Communications of the ACM 42 (8) (1999) 91–98.

[8] B.P.S. Murthi, S. Sarkar, The role of the management sciences in research on personalization, Management Science 49 (10) (2003) 1344–1362.

[9] N.F. Awad, M.S. Krishnan, The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization, MIS Quarterly 30 (1) (2006) 13–28.

[10] L.P. Hung, A personalized recommendation system based on product taxonomy for one-to-one marketing online, Expert Systems with Applications 29 (2) (2005) 383–392.

[11] G. Adomavicius, A. Tuzhilin, Personalization techniques: a process-oriented perspective, Communications of the ACM 48 (10) (2005) 83–90.

[12] M. Montaner, B. Lopez, L.D.A. Mosa, A taxonomy of recommender agents and the Internet, Artificial Intelligence Review 19 (2003) 285–330.

[13] G. Beyah, P. Xu, H.G. Woo, K. Mohan, D. Straub, Development of an instrument to study the use of recommendation systems, Proceedings of the Ninth Americas Conference on Information Systems, Tampa, FL, USA, 2003, pp. 269–279.

[14] C.P. Wei, M.J. Shaw, R.F. Easley, A survey of recommendation systems in electronic commerce, in: R.T. Rust, P.K. Kannan (Eds.), e-Service: New Directions in Theory and Practice, M.E. Sharpe Publisher, 2002.

[15] J.B. Schafer, J.A. Konstan, J. Riedl, E-commerce recommendation applications, Data Mining and Knowledge Discovery 5 (1) (2001) 115–153.

[16] H. Sakagami, T. Kamba, Learning personal preferences on online newspaper articles from user behaviors, Computer Networks and ISDN Systems 29 (1997) 1447–1455.

[17] K. Lang, et al., NewsWeeder: learning to filter netnews, Proceedings of the 12th International Conference on Machine Learning, Tahoe City, California, 1995, pp. 331–339.

[18] D.W. Oard, J. Kim, Implicit feedback for recommender systems, Proceedings of AAAI Workshops on Recommender Systems, Madison, WI, 1998, pp. 81–83.

[19] T.P. Liang, H.J. Lai, Discovering user interests from web browsing behavior: an application to Internet news services, Proceedings of the 35th Annual Hawaii International Conference on System Sciences, Big Island, Hawaii, USA, 2002, pp. 203–212.

[20] M. Balabanović, Y. Shoham, Fab: content-based, collaborative recommendation, Communications of the ACM 40 (3) (March 1997) 66–72.

[21] M.J. Pazzani, A framework for collaborative, content-based and demographic filtering, Artificial Intelligence Review 13 (5–6) (1999) 393–408.

[22] R.J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, Proceedings of the fifth ACM conference on Digital libraries, San Antonio, Texas, United States, 2000, pp. 195–204.

[23] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing 7 (1) (2003) 76–80.

[24] J.W. Kim, K.K. Lee, M.J. Shaw, H.L. Chang, M. Nelson, R.F. Easley, A preference scoring technique for personalized advertisements on Internet storefronts, Mathematical and Computer Modeling 44 (2006) 3–15.

[25] R. Mukherjee, N. Sajja, S. Sen, A movie recommendation system—an application of voting theory in user modeling, User Modeling and User-Adapted Interaction 13 (2003) 5–33.

[26] R. Garfinkel, R. Gopal, A. Tripathi, F. Yin, Design of a shopbot and recommender system for bundle purchases, Decision Support Systems 42 (2006) 1974–1986.

[27] K.J. Mock, V.R. Vemuri, Information filtering via hill climbing, WordNet, and index patterns, Information Processing & Management 33 (5) (1997) 633–644.

[28] D. Billsus, M.J. Pazzani, A personal news agent that talks, learns and explains, Proceedings of the Third Annual Conference on Autonomous Agents, Seattle, Washington, USA, 1999, pp. 268–275.

[29] H.J. Lai, T.P. Liang, Y.C. Ku, Customized Internet News Services Based on Customer Profiles, Proceedings of the Fifth International Conference on Electronic Commerce, Pittsburgh, PA, USA, 2003, pp. 225–229.

[30] W. Fan, M.D. Gordon, P. Pathak, Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison, Decision Support Systems 40 (2005) 213–233.

[31] T.P. Liang, H.J. Lai, Y.C. Ku, Personalized content recommendation and user satisfaction: theoretical synthesis and empirical findings, Journal of Management Information Systems 23 (3) (2006) 45–70.

[32] Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. Data and Knowledge Engineering, 25(1–2), 161–197.

[33] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (2003). The Description Logic handbook: Theory, implementation, and applications. Cambridge University Press. Semantic'Discoveries

[34] W. Kiessling, Foundations of preferences in database systems, in: Proceedings of the 29th International Conference on Very Large Data Bases, Hong Kong, China, 2002.

[35] IHMC CMaps, http://cmap.ihmc.us/. Last see 2009.

[36] M. Paolucci, T. Kawmura, T. Payne, K. Sycara, Semantic matching of web services capabilities, in: Proceedings of the First International Semantic Web Conference, Springer LNCS 2342, Sardinia, Italy, 2002.

[37] A semantic-expansion approach to personalized knowledge recommendation Decision Support Systems, Volume 45, Issue 3, June 2008, Pages 401-412 Ting-Peng Liang, Yung-Fang Yang, Deng-Neng Chen, Yi-Cheng Ku

[38] R. D. Couch, Computer diagnosis review and comment, Path. Ann. 11, 141-159 (1976).

[39] G. A. Gerry. Knowledge based systems for clinical problem solving, Decision Making and Medical Care, F. T. deDombal,-Ed., p. 23.North-H&land, New York (1976).

[40] D. Swanson, A. S. El&n, L. S. Schulman and S. A. Sprafka, Psychological analysis of physician expertise: implications for the design of physician support systems, Medinfo 77, p. 161. North-Holland, New York (1977).

[41] H. L. Bleich, Computer evaluation of acid-base disorders, J. clin. Invest. 48, 1689-1696 (1969).

[42] V. A. Catanzarite, A. G. Greenburg and H. J. Bremermann, Computer aided diagnosis: decision strategies and the clinical acceptability of consultation systems, Proceedings of the 14th Annual Hawaii international Confmence on System Sciences, Vol. III-Medical Information Processing, pp. 588-600. Honolulu, Hawaii (1981).

[43] P. Szolovits and S. Pauker, Categorical and probabilistic reasoning in medical diagnosis, Artif Intell. 11, 115-144 (1978).

[44] http://www.acgme.org/acWebsite/downloads/RRC_progReq/180neurology07012007.pdf . Last see 11/04/2010.

[45] Noy NF, McGuinness DL. Ontology development 101: a guide to creating your first ontology; 2001. Report No.: Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Stanford Medical Informatics Technical Report SMI-2001-0880.

[46] Smith B, Ceusters W, Klagges B, Kohler J, Kumar A, Lomax J, et al. Relations in biomedical ontologies. Genome Biol 2005;6(5):R46.

[47] Dumontier M, Villanueva-Rosales N. Three-layer OWL ontology design. In: Second international workshop on modular ontologies (WOMO07), Whistler, Canada, 2007.

[48] http://protege.stanford.edu/doc/users_guide/ . Last see 11/04/2010.