**INFORMATION SOCIETIES TECHNOLOGY (IST) PROGRAMME**

# HOMEY

**"Home Monitoring through Intelligent Dialog System"**

**DELIVERABLE D9 - PUBLIC**

**WORKPACKAGE: WP5 - Integration of semantic dictionaries**

# Specification for semantic dictionary integration

**Author:** Werner Ceusters (Language & Computing nv)
Martin Beveridge (Cancer Research UK)
David Milward (Cancer Research UK)
Daniele Falavigna (ITC-IRST)

**Submission Date: 31/07/2002**

**Partners: Engineering Ingegneria Informatica, (I), Reitek (I), Consorzio Bioingegneria e Informatica Medica (I), Istituto Trentino di Cultura(I), Cancer Research United Kingdom (UK), Language & Computing (B)**

# SUMMARY

This document is part of the result of the research project HOMEY funded by the IST Programme within the 5th Framework Programme as project number IST-2001-32434.

One of the goals of the HOMEY project is to develop a technology to be used for deploying innovative tele-medicine services. These new services will be based on an Intelligent Dialog System (IDS), designed and developed to effectively manage an incremental dialog between a tele-medicine system and a patient, taking into account user needs, preferences and time course of her/his disease.

The purpose of work package 5 of HOMEY is to provide the required medico-linguistic domain knowledge for the various scenario's in which the dialogue system will be used. Exploitation of this knowledge will be realised by integrating Language & Computing nv's (L&C). existing OntologyBrowser (OB) into the IDS such that it manages at runtime semantic dictionary look-ups as a service to the dialogue manager.

This deliverable describes the requirements put forward by the future users of the Homey system at the one hand, and the developers of the dialogue manager and dialogue specification engine at the other hand, that the semantic dictionairy module should adhere to. This deliverable is not the end-result of WP5. Rather it describes how the end-result will look like in the future.

# CONTENTS

# 1. Abstract

## 1.1. Purpose of the HOMEY project

One of the goals of the HOMEY project is to develop a technology to be used for deploying innovative tele-medicine services. These new services will be based on an Intelligent Dialog System (IDS), designed and developed to effectively manage an incremental dialog between a tele-medicine system and a patient, taking into account user needs, preferences and time course of her/his disease. Intelligent dialog requires the representation of goals, intentions, and beliefs about the effectiveness of the interaction in terms of quality of health care management. As a consequence, the dialog system will require dynamic adaptation, including understanding of the patient's medical problems and physician's goals, misunderstandings of therapy suggestions, and argumentation of therapy changes suggested by the system. To adapt the dialog, a representation of the medical domain knowledge, evolution of the disease of the specific patient, and the history of user-system interactions need to be represented.

## 1.2. Purpose of work package 5

The purpose of work package 5 is to provide the required medico-linguistic domain knowledge for the various scenario's in which the dialogue system will be used. Exploitation of this knowledge will be realised by integrating Language & Computing nv's (L&C). existing OntologyBrowser (OB) into the IDS such that it manages at runtime semantic dictionary look-ups as a service to the dialogue manager. In addition, a second software is developed to generate the necessary information out of the existing ontology management system LinkFactory, using the medical core-ontology of LinkBase, both exploited by L&C as part of their normal business. This guarantees portability of the approach to other (medical) domains and languages.

## 1.3. Purpose of this deliverable

This deliverable describes the requirements put forward by the future users of the Homey system at the one hand, and the developers of the dialogue manager and dialogue specification engine at the other hand, that the semantic dictionary module should adhere to. This deliverable is not the end-result of WP5. Rather it describes how the end-result will look like in the future.

The first part of deliverable provides a description of the various tools that are used either to collect, manage, and organise the semantic information in the future dictionary module, or to develop some necessary components of the dialogue manager and dialogue specification engine that the semantic dictionary module must communicate with. In the second part, a detailed account is given of the kind problems that might occur at runtime, and how detailed semantic information about the domain covered may help to solve these problems. In the third and last part, a functional description is given of the functions and knowledge components of the semantic dictionary module indicating how from a computational point of view the problems can be resolved.

## 2. List of abbreviations

| | | |
|---|---|---|
| IDS | Intelligent Dialog System | One of the main outcomes of the HOMEY project, provding innovative telemedicine services |
| L&C | Language & Computing nv | Homey partner responsible for WP5. |
| OB | Ontology Browser | Software package developed by L&C to query ontologies. Used, with minor adaptations, as a component in the IDS. |
| VisualTeSSI | Terminology Supported Semantic Indexing tool | Software packaged developed by L&C to validate automatic term extractions from free text.. |
| CRUK | Cancer Research UK | Homey partner responsible for WP6 and WP8 |
| PFC | PRO*forma* Composer | Graphical authoring tool for PRO*forma* specifications |
| PFE | PRO*forma* Engine | Enactment engine for PRO*forma* specifications |
| DSE | Dialogue Specification Engine | System being developed as part of WP6 to generate dialogue specifications based on a high-level task specification language and an ontology. |
| UMLS | Unified Medical Language System | Set of "Knowledge Sources" developed by the US National Library of Medicine to assist in the development of applications programs that help health professionals and researchers retrieve and integrate electronic biomedical information from a variety of sources. |
| DM | Dialogue Manager | |

## 3. State of the art

In this section, we describe shortly the various components that are used in the development of the semantic dictionaries, as well as those components that communicate directly with the OB.

### 3.1. Components used in the development of the semantic dictionaries

#### 3.1.1 LinkFactory®: formal ontology management

LinKFactory® [1, 2]. is a flexible and scalable solution for ontology management developed and used by L&C nv. It has been extensively used for modelling large medical terminologies. Key features include:

- Easy modelling of concepts with all relevant relationships and definitions.
- Seamless cross mapping toward external coding systems
- Versioning to ensure that references can be made to older versions of objects without losing information.

- Multilingual representation of the language-independent concepts, so that concepts can be stored with multiple surface representations in each of any number of languages.

- Multi-user architecture, which ensures integrity of the ontology during editing by multiple modellers.

LinKFactory® is designed as a platform-independent, 100% Pure Java™ application. The system consists of the LinKFactory® Server and the client-side LinKFactory® Workbench. LinKFactory® has been successfully tested on Windows, Solaris, and Linux.

At the back end, LinKFactory® stores the ontology in any Java-compatible relational database, and has been successfully tested on Oracle, Sybase, and SqlServer. Access to the database is abstracted away by a set of functions that are "natural" when dealing with ontologies, such as get-children, find-path, join concepts, and get terms for a specified concept.

The workbench is a dynamic framework for the LinKFactory® Beans. Each bean has its own specific functionality; combining multiple beans into the freely configurable workspace provides the user with a powerful tool for viewing and managing the data stored in the semantic network. The workbench allows the user to create multiple layouts for different tasks: searching, editing, attaching codes, etc.

Specific examples of LinKFactory Beans include Concept tree, Concept criteria with full definitions, Linktype tree, Criteria list, Term list, Search panel, Properties panel, and Reverse relation.

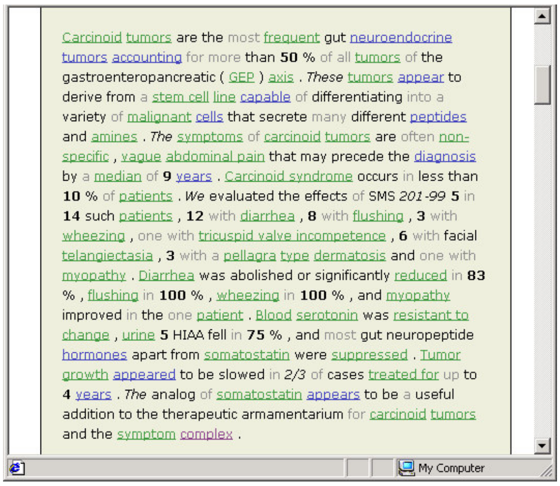
LinKFactory contains a number of features for ensuring ontology quality, including versioning, user tracking, user hierarchies, configurable user privileges, formal sanctioning with possibility to overrule, sibling-detection, and a linktype hierarchy.

Using LinkFactory®, L&C has build LinkBase®. LinKBase® contains approximately one million language-independent medical and general-purpose concepts, linked to natural language terms in several languages, including English. These concepts are linked together into a semantic network like structure using approximately 350 different link types for expressing formal relationships. These relationships are based on logics dealing with issues such as mereology and topology [3, 4], time and causality [5] and models for semantics driven natural language understanding [6, 7]. It is very important to note that in LinkBase® the formal subsumption relationship covers about 15% of the total number of relationships amongst concepts. As such, LinkBase® is a much richer structure than terminological systems in which term-relationships are expressed as strictly "narrower" or "broader". LinkBase®, or at least relevant extractions from it, is the driving force behind all our applications.

Within HOMEY, the LinkFactory® is used to develop the situated ontologies for hypertension and breast cancer, and to map these ontologies to LinkBase®.

### 3.1.2 VisualTeSSI®: term extraction and annotation

TeSSI® is designed by L&C nv for *Terminology-Supported Semantic Indexing* [8]. In order to perform semantic indexing, TeSSI® first segments a document into individual words and phrases. It then matches words and phrases in the document to individual LinKBase® concepts via the associated terms. This step introduces ambiguity, since some concepts have terms in common. To resolve cases of ambiguity, TeSSI® uses domain knowledge from LinkBase® to identify which concept out of the set of concepts that are linked to a homonymous phrase best fits with the meaning of the surrounding terms in the document. Figure 1a shows the output of TeSSI® at the end of this stage. Identified words and phrases are underlined. Figure 1b shows for the same text, the results obtained by a generic statistics-based phrase extractor that does not enjoy the wealth of a rich domain ontology such as TeSSI®.



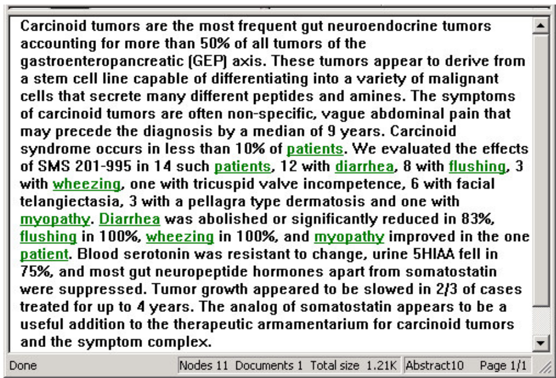




*Figure 1a and 1b: comparison of ontology-based concept extraction (left figure) versus statistical concept extraction (right figure).*

In the next step, TeSSI® uses the relationships between concepts identified in the document and the domain knowledge in LinkBase® to infer additional concepts which do not explicitly occur in the document. The end result of that process is a graph structure in which nodes correspond to concepts present (or inferred to be implicitly present) in the document, and arcs to semantic relationships derived from the domain ontology or co-occurrence relationships derived from the position of terms in the document. The arcs are weighted according to semantic distance in LinkBase® and term proximity in the document. The nodes are weighted based on their occurrence in the document. Having identified all the medical (and many non-medical) concepts in a document, TeSSI® then ranks these concepts in order of their relevance to the document as a whole, hence identifying the topic(s). Relevance scores are on a scale of 0 to 100, with 100 representing the most relevant concept. To determine these scores, TeSSI® uses a constraint spreading activation algorithm on the constructed graph [9]. In this way, semantically related concepts "reinforce" each others' relevance rankings. The rationale for this algorithm stems from the observation that the concepts in any particular document will vary in their semantic independence from each other. For example, a document might contain one mention each of "heart failure," "aortic stenosis," and "headache." The first two of these concepts are clearly more closely related to each other than to the third. An indexing system based entirely on term or concept frequency will treat these three concepts independently, thus assigning them all the same relevance. Yet intuitively, based on this limited description, the document has twice as many mentions of heart disease as of headache. TeSSI® takes advantage of its underlying medical ontology to more accurately represent this type of phenomenon. The relevance ranking algorithm is non-linear, and so the behaviour cannot be described analytically. It is, however, important to characterize the behaviour in order to normalize and optimise the rankings for incorporation into information retrieval systems and other applications.

Tessi® is used in HOMEY to extract the relevant concepts from free text documents in order to populate the semantic lexicons.

### 3.1.3 OntoShrinker®: alignment of situated ontologies

The OntoShrinker®, developed by L&C, shrinks a given ontology to a predefined set of concepts presented as a “situated ontology”, i.e. an ontology designed for a specific purpose. It does this by removing all concepts from the task-independent core-ontology that are not used directly or indirectly by the situated ontology. An additional set of concepts can also be kept, and the links from the initial set of concepts to those concepts, except for the IS_A link.

By default, the OntoShinker removes concepts having no more links. A parameter can be set if those concepts should be kept in the reduced ontology.

Method:

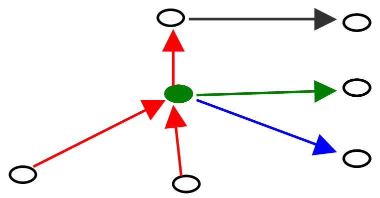
Eliminate the concepts that will not be included in the limited domain and re-link where the gap occurs.

Before a concept is removed from the ontology:

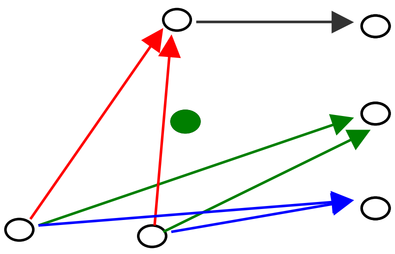
- All the outgoing links for this concept are moved to the children of that concept: each child will receive the outgoing links of the removed parent. A child normally inherits this information from its parent, so by moving the relations to each of the children, this information can be kept. If the concept has no children, the information covered by the outgoing links is lost.

E.g.: In the example the green concept is removed. Both of its children now get the outgoing links (green and blue arrow), and will become a child of the green concept’s parent. The red arrow symbolised the IS_A link.

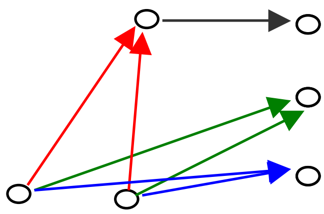
Before:



After relocating the links: The children of the green concept now have all the links the green concept had:



The green concept is now detached from its related concepts, and can safely be removed:



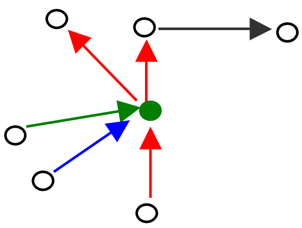
- All the incoming links for this concept are moved to the parents of that concept: each parent will receive the incoming links of the removed child. A concept X referring to another concept Y indirectly refers to the parent(s) of Y, so by moving the relations to each of the Y's parents, this information can be kept. If the concept Y has no parents, the information covered by the incoming links (a link from X to Y is an incoming link for Y) is lost.

  In the previous example, one can already see how both children of the green concept become direct child of the parent of the green concept, after relocating the links. The IS_A link however is actually a specially treated link: it is also the path followed for the relocation of the other links.

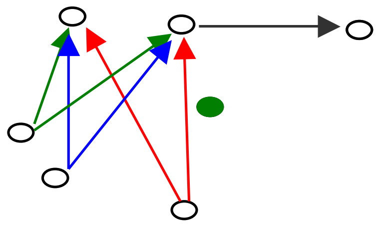
  The follow example shows how incoming links for the green concept are relocated.

E.g.: In the example the green concept is removed. The green concept has 2 parents, both of the parents now get the incoming links (green and blue arrow). The child of the green concept will become a child of both parents. The red arrow symbolised the IS_A link.

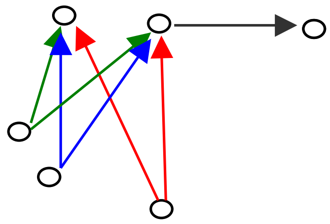
Before:



After relocating the links: the concepts that were linked to the green concept are now linked to the parents of the green concept:



The green concept is now detached from its related concepts, and can safely be removed:



After removing all the unnecessary concepts, the addition set of concepts is processed. The processing of these concepts is quite similar to the processing of the concepts marked for removal, but with these exceptions:

- The incoming links are added to the parents instead of moved. This to ensure no information-loss in the remaining ontology with the initial set of concepts. The IS_A link gets a special treatment: it is moved to the parent instead of added.
- The concept is not removed from the ontology.

The OntoShrinker® is used in HOMEY to extract the relevant concepts out of the LinkBase® core-ontology on the basis of the situated ontologies developed within HOMEY.

### 3.1.4 OntologyBrowser (OB): querying frozen ontologies

This tool, developed by L&C, allows the user to browse a given set of medical data, which was extracted from an ontology, added from file or inserted by the user using the browser and is stored in a well-structured formation. Its functionality is not limited to standard browsing, but allows a set of queries to be made giving detailed information or even making assumptions to approach the wanted result. It permits the user to work with fixed terminological data (T-box data) and insert temporary application data (A-box data), allowing instances to be made from the different items inside the structure. At the end, whilst discarding the application data, the system can keep the T-box data unchanged. This browser can also perform a set of classification functions to enrich entered data (opposed to the classification made during input or insertion). The whole browser is built on the structure called the **L&C-Graph ™**.

The OB is used in HOMEY to provide semantic information to the dialogue manager.

## 3.2. Components communicating with the OB

### 3.2.1 Dialogue Manager

The spoken dialog system developed is provided in the project by ITC-irst. It can handle mixed initiative interactions over the telephone for information access in restricted domains. Since last year the system has been used by partner CBIM of the Homey project for developing a tele-medicine service that will provide health assistance to chronic patients affected by hypertension. In this system, the patients periodically call a dedicated telephone number and engage a spoken dialog with the system, which interacts with them to acquire clinical data, monitors their style of life and asks about the presence of side effects. The dialog is as close as possible to the interaction between a physician and a patient; the service also gives advice, by issuing alerts and prompts as appropriate, to keep the patient in a low risk class. For doing this, the system is connected to a database that records the detailed medical histories of the patients being followed. The goal of the periodical examination is to monitor blood pressure values, heart rate, weight, habits and other variables used to estimate certain standard risk indicators. As suggested in [10], the domain knowledge has been extracted from a set of world-widely accepted guidelines for the hypertension and dyslipidemia pathologies.

The basic idea underlying the dialog engine lies on the definition of a set of ``contexts", containing ``concepts" associated to sub-grammars. Therefore, a direct relation between concepts and grammars, activated during the various dialog turns, has to be established.

Our approach for language modeling makes use of recursive transition networks [11]. These are finite state networks whose arcs allow linking other grammars in a recursive way. The resulting language is context free. Since the decoding step of a speech utterance can backtrack both the grammars and the words along the best path of the language graph, the recognized string consists of a mix of words and structured information. Semantic tags are included in the recognized string as reported below:

*Hi, my name is (**NAME**(Ivano Azzini)**NAME**) and I have a (**PRESSURE**( pressure of (**VALUE**(2 hundred and 10)**VALUE**) )**PRESSURE**) and today I (**WEIGHT**(weigh (**VALUE**( one hundred )**VALUE**) kilos)**WEIGHT**).*

In the string above the semantic tags NAME, PRESSURE and WEIGHT identify the related variables to collect. Note that the string above can be seen as a parse tree.

The development of the understanding part of an application basically consists in designing a set of grammars for each elementary information (concept) to collect, hence each basic concept has associated one or more grammars. In the example above, a possible grammar for the concept NAME can be a list of the names and surnames linked in several ways (i.e. name-surname, surname-name, surname); for the concept PRESSURE the grammar could be a list of numbers ranging within a predefined interval, and so on. On the other hand, part of sentences that do not convey useful information (such as: ``my weight is", ``I weigh", ``the weight today is", etc.) can be efficiently represented with stochastic language models (i.e. n-gram language models [12]). In our approach we arrange grammars into two separate levels (see figure 2 below). A bigram grammar is at the top level, and includes links to regular grammars at a lower level; each of the latter is associated to a concept. Initially, bigrams are trained on a small set of hand-written sentences representing (in the designer's intention) what the users will say. Once a prototype is built, field data can be recorded from user interactions. Then, the collected data are transcribed and used to update bigram probabilities.

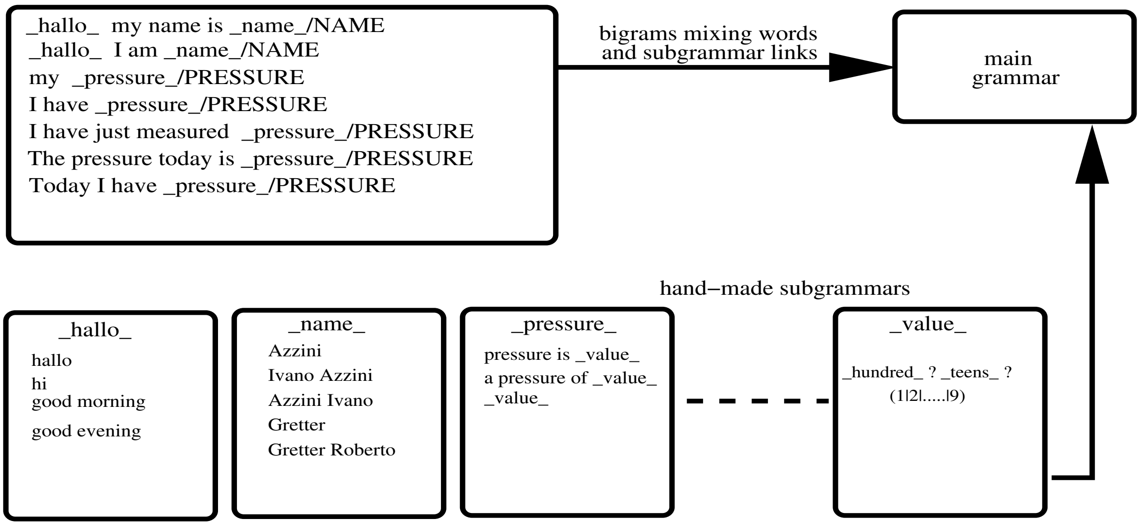



*Figure 2. Example of language models used in the Dialog Manager.*

The dialog system is ``frame based'', i.e. basic concepts are represented by records whose fields specify a set of actions to be executed by a dialog engine. For instance, fields can contain the name of a grammar to activate for an Automatic Speech Recognition (ASR) system, the text of a voice prompt to be uttered by a Text To Speech (TTS) synthesizer, a procedure for processing the ASR output, a procedure for doing a database query, etc. In particular each concept record contains a field to be filled with the values obtained by processing the ASR output. An application is defined through a set of records and the final goal of the dialog engine is to fill some of the concepts in a consistent way. Consistency is defined through a set of Boolean expressions involving both patient specific data or data collected during the interaction. Note that in this way the dialog architecture is independent of the application, since the only module to change across different domains is a file containing the description of the application itself, i.e. the concepts to be recorded and some actions associated to them) (see [13] for more details).

A recent version of the ITC-irst Dialog Manager can handle dialog descriptions defined in the VoiceXML Markup Language (see **www.voicexml.org**). It is worth noting that the Dialog manager used in the Homey project is the same for the partners involved: the PROforma Dialogue Specification Engine, described below, is a tool that can produce automatically description documents for the dialog.

For the hypertension domain two actors are allowed to enter data into a clinical database. The physician uses a conventional (graphics, keyboard and mouse) interface to store and update data. The patient is allowed to enter the data that she/he can acquire at home by using the telephone. To accomplish this task, the patient is instructed to dial, at fixed time intervals, a toll free number. One dedicated server with adequate hardware is connected to the telephone network and runs the commercial ``call center'' software (provided by partner Reitek), which answers phone calls and routes them to the ASR and TTS. When one call arrives, the dialog engine is started. After authenticating the user, it reads from a ``state vector'' a set of discrete variables encoding a patient specific target function. This last one determines which data must be acquired coherently with the evolution of the pathology on the basis of the knowledge related to the patient clinical history. In general, some fields of the basic concepts in the dialog description could not be filled according to patient habits (e.g. if the patient is not a smoker the system will not ask if she/he smokes, if the patient is not subjected to a pharmacological therapy the system will not ask for it, and so on). Hence, the state vector tell us which fields of the dialog description have to be acquired (mandatory) or not. Typically the target function has the following form:

- *Sport (mandatory|optional)*
- *Smokes (mandatory|optional)*
- *Blood Pressure (Dia) (mandatory|optional)*
- *Blood Pressure (Sys) (mandatory|optional)*
- *Weight (mandatory|optional)*
- *Heart Rate (mandatory|optional)*
- *Pharmacological Therapy (mandatory|optional)*
- *Day (mandatory|optional)*
- *Month (mandatory|optional)*
- *Time of Day (mandatory|optional)*

The target function is updated according to information contained in the database. This task is accomplished by a software agent (``adaptivity agent'') connected to a relational database, which holds the detailed medical records of each patient. As previously seen, the issues considered by the agent include habits, e.g. whether she/he is a smoker; history of prescribed drugs, because one wants to check for the presence of relevant side effects; whether she is on diet, etc.

Exceptional events like hang-up, multiple recognition errors and missing values (``I don't know'') are also handled by the agent; this allows the relevant state information to be carried on between calls, so that the missing information can be asked the next time the patient calls, if it is appropriate. The whole system therefore exhibits *long-term* adaptation behaviour described in [10]. This long-term adaptation for each patient takes place *after* she/he has completed the conversation with the system. The system is therefore able to adapt itself to the evolution of the disease, new patient habits or features, new physician directions, etc. A different kind of adaptation strategy is also considered, called *short-term* (or punctual) adaptation. This kind of adaptation takes *during* the course of a dialog; it can handle a range of user's behaviours.

### 3.2.2 Dialogue Specification Engine

The DSE is being developed as part of WP6 and is intended to map a high-level task specification language (based on existing knowledge representation schemes used in medicine) to dialogue specifications which can be sent either to a VoiceXML browser or to a generic dialogue manager such as that provided by ITC-IRST (described above). Currently the TSL is based on the PRO*forma* language and so the PRO*forma* toolset can be used for authoring and enacting task specifications. In addition the DSE makes use of a situated ontology in order to assist in mapping the TSL to a dialogue specification. The following sections therefore provide a brief overview of the PRO*forma* language and toolset, followed by a description of the DSE and how it relates to the PRO*forma* tools and the ontology browser (OB). A complete definition of the TSL will be provided in deliverable D11 and the final version of the DSE will be provided as deliverable D12.

#### 3.2.2.1 PRO*forma* Language

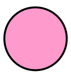
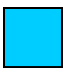
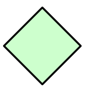
The PRO*forma* language is a derivative of the $R^2L$ language [14]. The syntax and semantics of this specification language have been formally defined and shown to be both sound and complete [15]. The PRO*forma* language supports the definition of clinical guidelines and protocols in terms of:

- A well-defined set of tasks that can be composed into networks representing plans or procedures carried out over time. These enable the high level structure of a guideline to be represented.

- Logical constructs (such as situations, events, constraints, pre- and post-conditions, and inference rules) which allow the details of each task and inter-relationships between tasks to be defined using templates.

The PRO*forma* language defines an ontology of four types of task, each of which is also associated with a graphical representation (icon) which is used in the graphical authoring tools for creating guidelines (described later). These tasks are described in the table below:

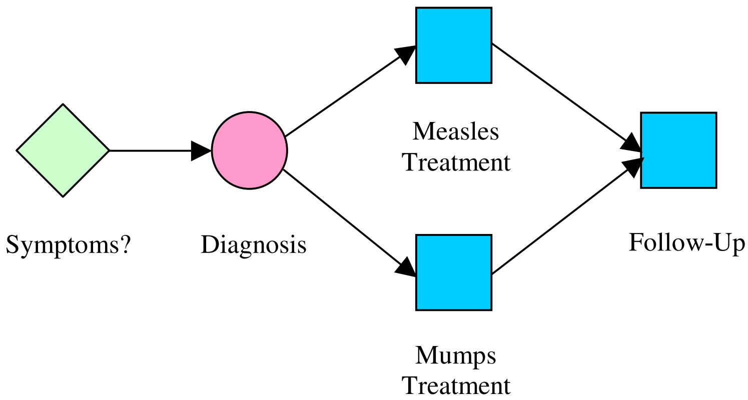
Each task type is implemented as a sub-class of an abstract 'task' super-class. The attributes of each sub-class determine the behaviour of its members during enactment of a guideline. All sub-classes inherit some generic attributes from the super-class which define: a goal that the task is to achieve, a trigger that causes a task to be considered for enactment (asynchronously), pre-conditions for enacting the task, post-conditions that should hold after enactment, a cycling schema to control task iteration and whether or not the task must be authorised by another agent before enactment.

| Icon | Task | Description |
|---|---|---|
|  | Plan | Sets of tasks to be carried out to achieve a clinical goal. Plans are the basic building blocks of a guideline, and may contain any number of tasks of any type (including plans). |
|  | Decision | Tasks which involve choices of some kind, such as a choice of investigation, diagnosis or treatment. Choices are made by argumentation over candidate proposals. |
|  | Action | Typically clinical procedures (such as the administration of an injection) which need to be carried out. |
|  | Enquiry | Actions returning required information; typically requests for information or data from the user. |

In addition to generic attributes, each sub-class defines some class-specific attributes. The plan class has additional attributes which define: the tasks that compose the plan (note that these may be of any type, including plan itself, hence allowing recursion to be represented), constraints on the scheduling order of tasks, conditions for successful termination, and conditions for unsuccessful termination. The decision class has attributes defining: decision candidates to be considered, arguments for and against candidates, whether one or many candidates can be chosen and the scheme for combining arguments. Actions have an attribute to define a procedure to be carried-out by another agent (either human or machine) and enquiries have attributes to define the set of data items for which values are to be obtained from another agent.

#### 3.2.2.2 Authoring and Enactment

In order to assist in the creation of PRO*forma* guidelines, the PRO*forma* toolset contains a graphical authoring tool, the PRO*forma* Composer (PFC), which allows guidelines to be specified by drawing a high-level diagram depicting the tasks involved (using the icons shown earlier) and the relationships, e.g. scheduling constraints, between them (represented by arrows). The authoring tool also supports the definition of generic task attributes (e.g. pre-condition, goal etc) and task-specific attributes (e.g the data sources for an enquiry). An example graphical representation of a guideline is shown in the figure below.



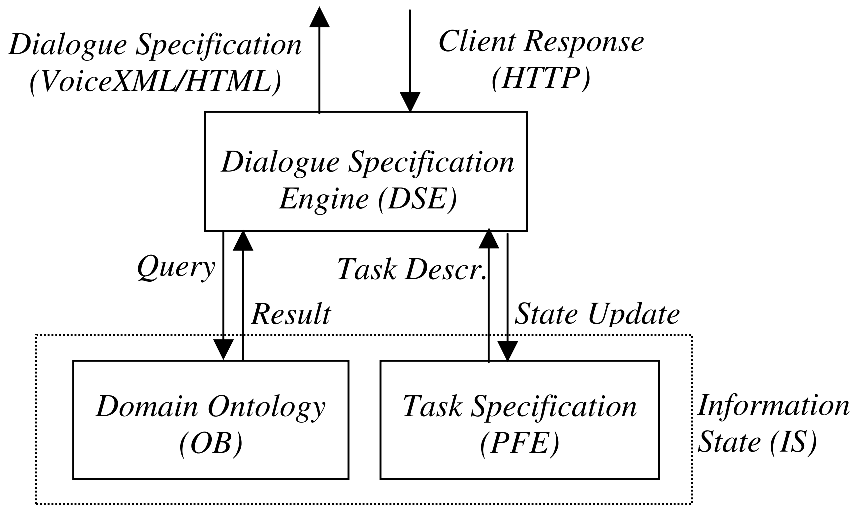

Once the guideline has been authored it can be submitted to the PRO*forma* engine (PFE) for enactment, at which point the individual tasks and attributes are used to generate procedures to carry-out. In the case of the example guideline given in the figure above, the enactment engine will request data regarding a patient's symptoms, make a decision based on that data as to whether the patient has measles or mumps, then, depending on the decision, provide instructions on treating the disease followed by some instructions on follow-up treatment. The PFE also provides an application programming interface (API) which allows other systems to interact with a guideline during its enactment, in order to provide a user interface for example.

#### 3.2.2.3 Relation of the DSE to the OB and PRO*forma* tools

As described previously, the DSE's function is to generate specifications for dialogues based on a high-level task specification language (TSL) representing a clinical guideline, and an ontology representing the domain associated with the guideline. Since the TSL is based on the PRO*forma* language, the PRO*forma* enactment engine (PFE) can be used to interact with the task specification. Similarly, the ontology supplied by L&C will be accessed via the OB.

The dialogue specifications generated are currently encoded either as VoiceXML or HTML documents depending on the type of client (browser) being used, but will be further developed to support the multimodal browser being developed in WP3. The relationship between the DSE, PFE and OB is shown in the figure below.

As shown in the figure, PFE and OB are considered to provide complementary parts of an information state (IS) which determines the structure and sequence of the generated dialogues. Currently it is envisaged that PFE will provide aspects of dynamic information state such as the process specification and state model (through its support for data item definitions) whilst OB will provide static information state (domain model) via its T-box. However, a further possibility that may be worth investigation is use of a case ontology (A-box) to represent instance information rather than the data items provided by PFE.




# 4. Functional specifications for the OB

## 4.1. Necessary knowledge

The OB should contain the following knowledge elements:

1) A semantic representation of the relevant concepts involved in (a) the guideline for referral of patients with breast problems, as specified in [16; Section 4.5, pp 6-7] and [17], (b) the guideline for identifying women at substantial increased risk of breast cancer [16; Annex 1, p 43] and (c) the WHO's Hypertension guideline. This representation must contain the terminology required to describe these guidelines, as well as a formal description of the various relationships amongst the relevant concepts. The terminology provided should include inflected forms of terms, or alternatively some mechanism should be provided for application developers to add and retrieve inflected forms.

2) A semantic representation of those concepts NOT found in the guidelines mentioned above, but that are relatively closely related to it. For example: if the guideline refers to the

concept of "breast cancer", then all concepts referring to various types of breast cancer should be present, even if they are not explicitly mentioned in the guidelines.

3) The terminology of those concepts in English and Italian, both covering professional terms and laymen terms where applicable.

4) Instances of concepts (case ontology) determined by applications at run-time. Such instances constitute temporary application data which should be kept separate from the underlying ontology (fixed data).

## 4.2. Essential queries

### 4.2.1 Mapping between concepts and lexical terms

In order to generate speech grammars dynamically for a dialogue and interpret user responses it will be necessary to map domain concepts into lexical terms and lexical terms into domain concepts. Furthermore the relationships between concepts and terms will be many-to-many in order to capture both polysemy (one term mapping to several concepts) and synonymy (one concept mapping to several terms). For example, consider the following exchange (utterances marked "S" are system utterances and those marked "U" are utterances by the user):

(1) S: Are there any <u>skin changes</u>?

U: Yes there is some <u>distortion</u>

The term "distortion" supplied by the user is polysemous, i.e. it might refer to various concepts in the ontology (e.g. skin distortion, nipple distortion etc) and so the system has to determine which concept is intended. Since the context is skin changes, the ontology can be searched to find a concept which is related to skin change and for which "distortion" is a term, hence disambiguating the user's reply.

In addition to the problem of polysemy, the user may also use synonyms of expected terms in their response. For example:

(2) S: Does the patient have cardiac arrhythmia?
*[system expects one or more of: <u>atrial fibrillation</u>, <u>sinus tachycardia</u>, <u>AV block</u>, …]*

U: Yes, <u>auricular fibrillation</u>

S: Is it paroxysmal or persistent?

In this case the valid responses specified in the TSL included "atrial fibrillation", but the user replied with "auricular fibrillation". However, the system has recognized "auricular fibrillation" as being a synonym of "atrial fibrillation" and has therefore accepted the reply and resolved it to the expected concept. The system then proceeds to request further information regarding atrial fibrillation.

Due to the many-to-many nature of the mapping between concepts and terms, it will also be necessary to capture the notion of a 'preferred term' for a concept (as used in UMLS [18]), to provide a default term to be used when there is no basis for choosing between synonyms. For example:

(3) S: Is there any atrial fibrillation?

In this case both "atrial fibrillation" and "auricular fibrillation" would be acceptable terms for the concept being prompted for, but "atrial fibrillation" has been chosen as it is defined as the preferred (default) term.

### 4.2.2 Finding inheritance relations between concepts

This section describes queries against the OB which require traversal both up and down the inheritance hierarchy of concepts in the ontology in order to determine hyponymy and hyperonymy relations between terms.

#### 4.2.2.1 Use of hypernyms and hyponyms in input

During dialogue the user may provide a response to a prompt that does not match any of the expected range of responses (as defined by the TSL) but is a hypernym or hyponym of an expected response. In this case the system should be able to discover the relation between the expected terms and the user response in order to resolve the discrepancy.

In the case where the user's reply was a hypernym of an expected term then their reply can be considered under-specified and the system can issue clarificatory questions in order to obtain a more detailed reply. For example:

(4) S: Do you have a family history of chronic disease?
*[system expects one or more of: lung cancer, leukemia, sarcoma, …]*

U: Yes, cancer

S: What type of cancer?

U: Lung cancer.

In this example the system expected a specific disease but the user replied with a more generic term. The system therefore formulated a more specific question in order to elicit an answer at the expected level of the is-a hierarchy.

In the case where the user's reply was a hyponym of an expected term then their reply can be considered over-specified and the system can (a) find a more general related term which matches the expected responses in order to answer the current question and (b) avoid asking subsequent more specific questions that have already been answered. For example:

(5) S: Have you had any chronic diseases?
*[system expects one or more of: cancer, hypertension, diabetes, …]*

U: Yes, Leukemia

S: # Have you had cancer?

In this example the system expected a more generic answer to the question (such as "cancer") but the user replied with a more specific term. The system, however, was able to match the answer to the question on the basis that leukemia is a cancer which is a chronic disease and also avoids asking a subsequent question as to whether the patient has had cancer, which has already been answered and would therefore be pragmatically ill-formed (indicated by a "#").

#### 4.2.2.2 Disambiguation of definite referring expressions

Use of inheritance relations can also be used to resolve definite referring expressions where the user refers to a previously introduced concept via a hyponym or hypernym. For example:

(6) S: What diseases have you been diagnosed with?

U: High blood pressure and <u>Leukemia</u>

S: Have you undergone any treatments?

U: Yes, I had treatment for <u>the cancer</u> last year

S: So you haven't been treated for high blood pressure?

In this example the system recognizes that the expression "the cancer" refers to "Leukemia" and not "high blood pressure" by virtue of the fact that Leukemia is a cancer. It therefore correctly proceeds to clarify whether the user has had treatment for their high blood pressure.

### 4.2.3 Finding associative relationships amongst concepts

This section describes queries against the OB which require traversal not just of the inheritance hierarchy of concepts in the ontology but also other non-inheritance (associative) links between concepts.

#### 4.2.3.1 Under-specified and over-specified input

During dialogue the user may provide a response to a prompt that does not match any of the expected range of responses (as defined by the TSL) and furthermore is not a hypernym or hyponym of an expected response, but is associated with one or more of the expected responses by a non-isa link which imposes an ordering on concepts (e.g. part-whole links). In this case the system should be able to discover the associative relation between the expected terms and determine whether the supplied term is more or less specific than the expected terms according to the ordering imposed by the relation.

In the case where the user's reply can be considered under-specified with respect to the expected terms the system can issue clarificatory questions in order to obtain a more specific reply. For example:

(7) S: Where does it hurt?
*[system expects one or more of: <u>elbow</u>, <u>wrist</u>, <u>shoulder</u>, …]*

U: In my <u>arm</u>.

S: Where in your arm?

U: In my elbow.

In this example the system initially expected a more specific body-part than the user supplied, but recognized that the supplied term "arm" was related to the expected terms in a part-whole hierarchy and was more general then the expected terms. The system therefore did not simply repeat the original question "where does it hurt?" but instead formulated a new question to elicit an answer at the expected level of the part-whole hierarchy.

In the case where the user's reply can be considered over-specified with respect to the expected terms the system can (a) find a more general term which matches the expected responses in order to answer the current question and (b) avoid asking subsequent more specific questions that have already been answered. For example:

(8) S: Where does it hurt?
*[system expects one or more of: arm, leg, head, …]*

U: In my elbow

S: # Does your arm hurt?

In this example the system expected a more generic answer to the question (such as "arm") but the user replied with a more specific term ("elbow"). The system, however, was able to match the answer to the question on the basis that an elbow is a part of an arm, and so avoided asking a subsequent question as to whether the patient's arm hurt, which had already been answered and would therefore be pragmatically ill-formed.

**4.2.3.2 Interpreting non-literal language use**

It has been suggested [19, 20] that inferences based on relations can be used to resolve non-literal uses of language such as metonymy (e.g. by finding paths between concepts). For example:

(9) U: What was the result of the angioplasty of the stenosis?

S: The segment was successfully enlarged

(10) U: What was the result of the angioplasty of Mr Jones?

S: The segment was successfully enlarged

An angioplasty is an action performed on an artery segment to enlarge it, whilst stenosis describes the state of an artery that has reduced diameter. In both the above cases, therefore, the Theme of the action "angioplasty" is different from the expected type of 'artery segment' and so both utterances can be considered metonyms. Such usage could be resolved by finding a path from 'angioplasty' to 'stenosis' (e.g. angioplasty of a segment of an artery which is in a state of stenosis) or from 'angioplasty' to 'Mr Jones' (e.g. angioplasty of a segment of an artery of a patient called Mr Jones) hence providing a literal interpretation.

#### 4.2.3.3 General inference

Knowledge of associative relations might also be used to perform more general inference to disambiguate user replies. For example:

(11) S: What diseases have you been diagnosed with?

U: High blood pressure and Leukemia

S: Have you undergone any treatments?

U: Yes, I underwent chemotherapy last year

S: So you haven't been treated for high blood pressure?

In this example the system recognizes that chemotherapy is a treatment for cancer and so the user must be referring to treatment for their Leukemia (because Leukemia is a cancer) and not high blood pressure. A more complex example is:

(12) S: What diseases have you been diagnosed with?

U: Prostate cancer.

S: # What is your sex?

In this case the system has determined from the ontology that prostate cancer is a disease that only effects men and so the subsequent question regarding the patient's sex has implicitly been answered and is therefore pragmatically ill-formed.

#### 4.2.3.4 Interpreting noisy input

General inference from the ontology might also be used to help interpret noisy input where the result returned from the speech recogniser contains omissions. For example:

(13) U: The patient…65 years…lump…left breast

In this case the system may be able to infer from the noun phrases returned that the patient is 65 years old and has a lump in their left breast. This inference could be achieved on the basis of part-whole relations for the particular domain whereby lumps occur on body parts, a body part is part of a patient and patients have ages measured in years. This would exclude interpretations in which, for example, the lump is associated directly with the patient (leaving the term "left breast" uninterpreted) or in which the phrase "65 years" is associated with an entity other than the patient (e.g. a 65 year-old lump). A further example is:

(14) U: The lump…the arm…has grown

Once again, assuming an ontology situated in the cancer domain, it could be assumed that the lump is associated with the arm and that it is the lump, not the arm, that has grown (since the growth of lumps, rather than arms, is of interest in the cancer domain and so is the interpretation that would be captured by the ontology).

### 4.3. Insert/update/delete functionality

The OB should support insert, update and delete functions on both the underlying ontology (fixed data) and instances (application data). Update operations may be required on the ontology in order to amend it as necessary to meet application requirements. Such updates will therefore be made as part of the authoring process rather than at run-time. Update operations will be required against concept instances in order to allow application data to be captured without changing the underlying ontology. With respect to the DSE, this functionality is required in order to investigate the use of instance data (case ontology) to store state information rather than the data items provided by the PFE.

### 4.4. Load/Save functionality

The OB should allow an ontology specification to be loaded from and saved to file so that amendments to the ontology can be saved and re-used. The OB should also allow instance data to be loaded from and saved to file so that application data can be saved and subsequently re-used. For example, information gathered regarding a patient during one dialogue could be saved and then re-loaded when the system next engages in a dialogue with that patient. The system would then not have to repeat in every dialogue requests for data that is unlikely to have changed (e.g. the patient's sex).

Furthermore the OB should provide the ability to save out the full ontologies developed for the cancer and hypertension domains in a format that will allow them to be imported into other ontology management systems (to allow further exploration of the possibilities for use of the ontological data).

### 4.5. Run time specification

The OB will provide both a Java and C++ API that will connect the Dialog Manager with the OB itself. The API will be capable to handle simultaneously multiple connections with the DM (i.e. it will be possible to have multi-thread connections). In practice, the communication will take place through a TCP-IP connection, while the data format are ASCII strings coded using the HTPP protocol.

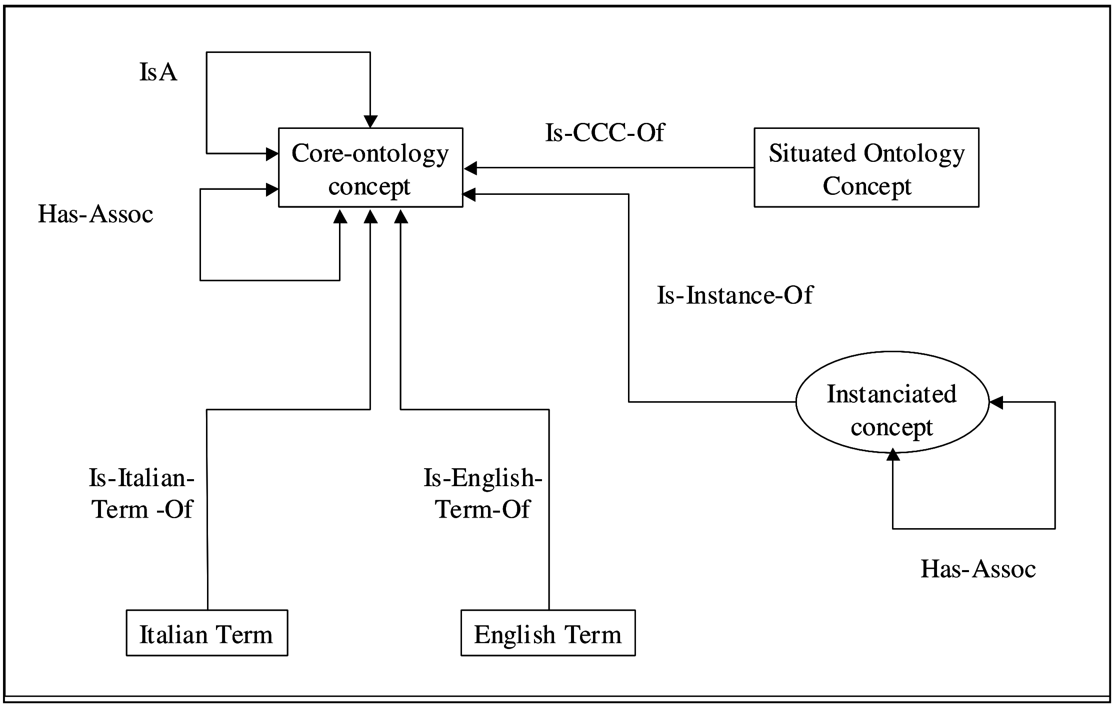
The DM will provide to the OB a query (essentially a word or a list of words) in an XML format; the OB will reply with the list of related concepts also in XML format. The words contained in the list returned by the OB will also have a "semantic" structure, that is one or more attributes, referring some medical knowledge, will be associated to the words in the list.

The OB will also provide the capability to insert, not in run time, new terms and knowledge related to the hypertension domain.

## 5. How the OB addresses the requirements

### 5.1. HOMEY semantic lexicon representation

The semantic lexicon will contain a dynamic and a static component. The static component contains all the information that is available to the DM prior to any actual dialogue. The dynamic component is able to contain the information gathered during a session with the DM.




Schema of the Semantic Lexicon

### 5.1.1 Static component of the semantic lexicon

The static component consists of three main information blocks: a situated ontology, a core-ontology, and a lexicon.

The situated ontology is a direct representation of the clinical guidelines. E.g. for the breast cancer referral guideline, it will contain a set of concepts that each represent a woman with a specific configuration of symptoms and signs upon which a specific referral advice is to be decided. The core-ontology contains all the concepts that are required to formally understand the meaning of the concepts in the situated ontology, and in addition the concepts that are subsumed by these concepts. The lexicon contains the Italian and English terms associated with the concepts in the core-ontology.

Concepts from the situated ontology are linked to concepts in the core-ontology by means of the IS-CCC-OF relationship.

Concepts in the core-ontology are linked using the formal subsumption relation ISA and associative relationships. The concepts linked by ISA form together a directed acyclic graph which is a generalisation over a tree allowing multiple parents for a concept. The associative relationships provide additional information over the concepts from which they point outwards. Associative relationships exists in pairs, the general naming convention being "Has-X" versus "Is-X-Of". Within the static component of the lexicon, no assumptions may be made with respect to the possible validity of "C1 Has-X C2" in case only "C2 Is-X-Of C1" is present. The semantics of the static component stipulate that when a relationship is assigned to a concept, it necessarily is valid for all instances of that concept, including all instances of the descendants of that concept. E.g., if the static component contains the statement "breast cancer Is-Spatial-Part-Of" breast", then this means that this statement is true for all instances of breast cancer. It is (obviously) however NOT true that all instances of breast Has-Spatial-Part breast cancer.

Besides a list of relationships towards other concepts, there are also full definitions available, specifying the necessary and sufficient conditions candidate instances of specific concepts must adhere to in order to qualify as an instance. Full definitions are useful for inferencing over instances at runtime[1].

Finally, most of the concepts are linked to English and Italian terms using the Has-X-Term relationship.

### 5.1.2 Dynamic component of the semantic lexicon

The dynamic component of the semantic lexicon is a traditional A-Box reasoner that at runtime should contain the information accumulated during a DM session. It is also possible to preload known data over a specific patient, was the patient has been identified. The dynamic component has the facilities for preloading, but is of course not responsible for maintaining these collections outside DM sessions.

Whereas – as explained above – associative relationships must not be inversed at the level of the concepts, they are inversed in the dynamic component. Indeed, if a specific woman has a specific cancer in her breast, then that specific breast is the seat of a breast cancer. The A-box reasoner is capable of using this information.

At the other hand, there are no ISA relationships between instances, but only is-instance-of relationships between instances and the relevant concepts in the static component.

## 5.2. Responding to essential queries

In this section, we describe from a functional perspective how the requirements put forward by the DM and the DSE are met by the OB. The actual queries to perform are described in the API-documentation of the OB, which is confidential information (background IP) that is only disclosed on a need-to-know basis to parties that signed an NDA with L&C nv.

### 5.2.1 Mapping between concepts and lexical terms

As described above, the relationships between concepts and lexical terms are realised by means of the Has-English-Term and Has-Italian-Term links. To retrieve all terms associated to a specific concept, the host application must query for all terms starting from the concept

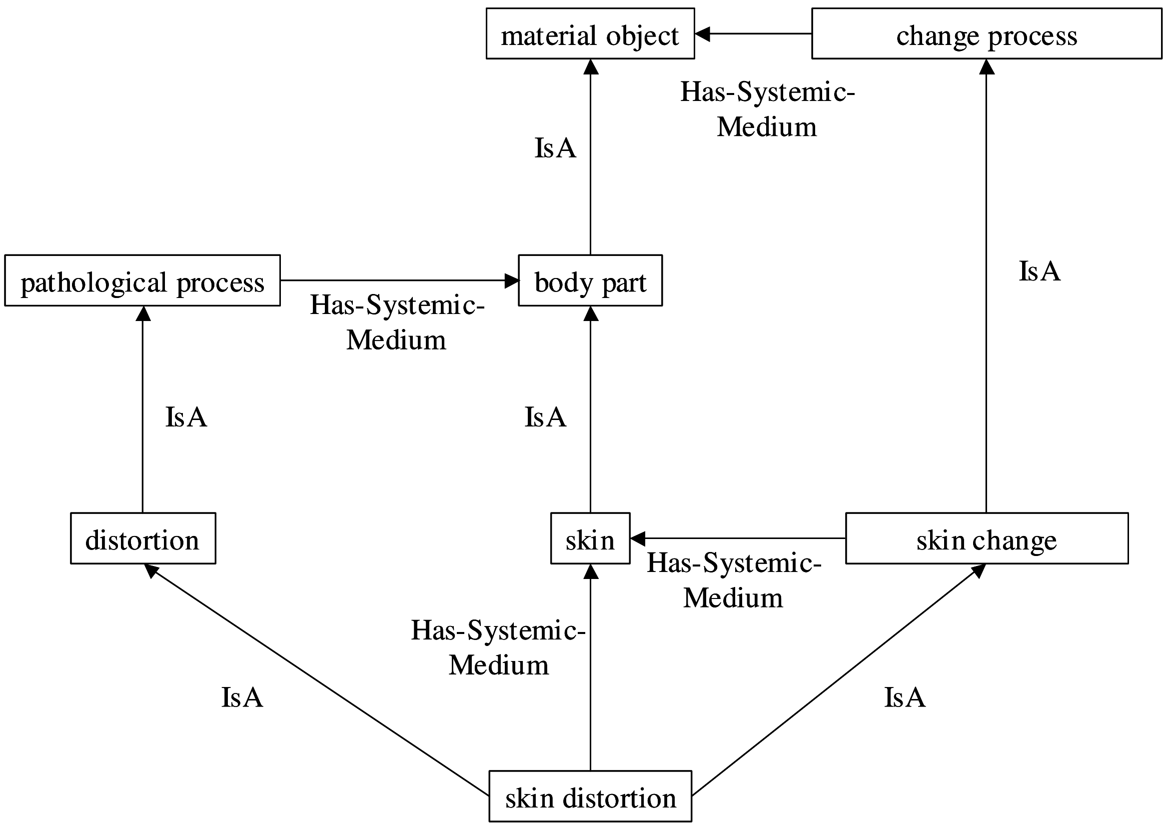
[1] COPYRIGHT NOTICE: the situated ontologies and the mappings from the situated ontologies towards the core-ontology are foreground knowledge in Homey. The core-ontology is background information being part of the IPR of Language & Computing nv and cannot be disclosed to any party outside Homey without prior consent of Language & Computing nv. By the Homey partners, it may only be used for the specific purposes of the Homey-project until the end of the project. Afterwards, the conditions explained in Annex II of the contract and the Consortium Agreement will be applied.

under scrutiny. To find out whether a term is polysemous, it suffices to query for all concepts that have a term-link to the term under scrutiny.

For the "skin change/distortion" example, this would work as such:

S: Are there any skin changes?

U: Yes there is some distortion




This is not as much an example of polysemy, but rather hierarchical underspecification. Depending on whether or not the concept "skin distortion" is in the semantic lexicon, the "FindPath" or "Assumption" methods of the OB will return the relationship between "skin change" and "distortion".

If "skin distortion" is present, the "FindPath"-method will be successful. This method searches for concepts that have a path through outgoing links towards the concepts for which a semantic relationship is requested. This search method allows the user to set a lot of parameters. It allows the user either to choose direction of links on the path to both concepts separately, or just to ignore the direction of the links. It is possible to specify the maximum distance for any concept on the found path to either of the input concepts. The minimal distance option will stop at the first distance it finds a path, as long as it is smaller than the maximum distance given (for each concept). It is also possible to include or exclude link-types from the search.

If the FindPath-method would fail, then the "assumption"-method will infer that the concept "skin distortion" might exist based on the following evidence: 1) skin IsA body-part, 2) distortion Has-Systemic-Medium body-part; hence it is possible to define a concept that IsA distortion and that Has-Systemic-Medium skin.

### 5.2.2 Finding inheritance relations between concepts

#### 5.2.2.1 Use of hypernyms and hyponyms in input

These are trivial situations that can be resolved by querying the OB for child and/or parent concepts recursively.

#### 5.2.2.2 Disambiguation of definite referring expressions

This requires the same type of querying as in the previous paragraph.

### 5.2.3 Finding associative relationships amongst concepts

All query-specifications listed above come down to perform a search for a path through the semantic network in the OB. It doesn't matter whether or not these links are IsA-links or associative links.

As an example, we describe here the first situation as presented above. Suppose the DM asks the user "Where does it hurt? " and the system expects as answer one or more of:elbow, wrist, shoulder. When the user answers "In my arm.", a sensible question would then be "Where in your arm?". The DM can be given the information that this is a sensible question by asking the OB for the relationship between "elbow" and "arm", "wrist" and "arm", etc. Using the FindPath-method (or any more specialised method) the OB will return linktypes of the type "Is-Proper-Material-Part-Of", or "Is-Linear-Division-Of", i.e. linktypes indicating a mereotopological relationship between the concepts entered. From this information, the DM may decide a "Where"-question being appropriate.

The domain model used to describe the guidelines contains several associative links. They are grouped in a few larger groups that correspond with typical questions. The differences are sometimes minimal, important for detailed reasoning, though not important for the purposes of the DM.

Below, the used links are presented in three-column tables. The first column contains the base-link, the second the inverse link (may be the same as the first one in case of mathematical reflexivity of the link, or may not exist in which the 2nd column is empty). The third column gives some additional information where needed.

#### 5.2.3.1 Temporal relationships

Links that specify that a temporal relationship between the related entities exist.:

| | | |
|---|---|---|
| HAS-TEMPORAL-RELATION | HAS-TEMPORAL-RELATION | just specifies there is one |
| HAS-TEMPORAL-LOCATING | | most generic temporal link that answers the "When ?" question. |

Links that relate the start of two temporal periods

| | | |
|---|---|---|
| HAS-START-REFERENCE-WITH | HAS-START-REFERENCE-WITH | |
| HAS-START-LATER-THAN | HAS-START-EARLIER-THAN | |
| HAS-START-NOT-EARLIER-THAN | | |
| HAS-SAME-START-THAN | HAS-SAME-START-THAN | |

Links that relate the end of two temporal periods

| | | |
|---|---|---|
| HAS-FINISH-NOT-LATER-THAN | | |

Links that relate the totality of a temporal period to the totality of another one:

| | | |
|---|---|---|
| HAS-HAPPENING-LATER-THAN | HAS-HAPPENING-EARLIER-THAN | |
| HAS-CEN-OCCURRENCE-SINCE | | |
| HAS-CEN-OCCURRENCE-CO-CONTINUES | | |
| HAS-CEN-OCCURRENCE-BEFORE | HAS-CEN-OCCURRENCE-AFTER | |
| HAS-CEN-OCCURRENCE-UNTIL | HAS-CEN-OCCURRENCE-FOLLOWS | |
| HAS-CEN-OCCURRENCE-INCLUDES | HAS-CEN-OCCURRENCE-DURING | |
| HAS-CEN-OCCURRENCE-COSTARTS | | |
| HAS-CEN-OCCURRENCE-AT | HAS-CEN-OCCURRENCE-AT | |
| HAS-OVERLAPPING-TEMPORAL-PERIOD-WITH | HAS-OVERLAPPING-TEMPORAL-PERIOD-WITH | |
| HAS-TEMPORAL-CONNECTING-PERIOD-WITH | HAS-TEMPORAL-CONNECTING-PERIOD-WITH | |
| HAS-NO-TEMPORAL-CONNECTING-PERIOD-WITH | | |
| IS-TEMPORAL-CO-OCCURRING-PERIOD-OF | | |

| | | |
|---|---|---|
| HAS-CEN-DURATION | | Specifies the duration of the source concept |

| | | |
|---|---|---|
| IS-FURTHER-EVOLUTION-OF | | Specifies that the target concept is an evolution of the source concept. |

### 5.2.3.2 Mereological relationships

Links specifying parthoods of various kinds:

| | | |
|---|---|---|
| HAS-GENERIC-PARTITION | IS-GENERIC-PARTITION-OF | generic parthood without any other assumption |
| IS-SUBEVENT-OF | | temporal parthood |
| HAS-MEMBER | | membership |
| HAS-MAKING-UP-PARTITION | | a series of this link starting from a source concept indicates that the mereological sum of the target concepts completely makes up the source concept. |

### 5.2.3.3 Topological relationships

| | | |
|---|---|---|
| HAS-REGIONAL-COINCIDENCE | | just specifies that 2 entities are somehow spatially related and have at least externally connecting boundaries. |
| IS-SPATIAL-EQUIVALENT-OF | IS-SPATIAL-EQUIVALENT-OF | perfect spatial overlap |
| HAS-SPATIAL-REFERENCE | | |
| HAS-SPATIAL-POINT-REFERENCE | | |
| HAS-SPATIAL-PART | IS-SPATIAL-PART-OF | |
| HAS-DISCONNECTED-REGION | | 2 entities don't touch each other spatially |
| HAS-PARTIAL-SPATIAL-OVERLAP | | |
| HAS-CONNECTING-REGION | | |
| HAS-PROPER-SPATIAL-PART | IS-PROPER-SPATIAL-PART-OF | |
| IS-TANGENTIAL-SPATIAL-PART-OF | | visibility from the outside of the larger |

| | | |
|---|---|---|
| | | entity |
| IS-NON-TANGENTIAL-SPATIAL-PART-OF | | non-visibility from the outside of the larger entity |
| HAS-EXTERNAL-CONNECTING-REGION | | 2 entities touch each other on the outside |
| HAS-OVERLAPPING-REGION | | |

#### 5.2.3.4 Locative orientation

| | | |
|---|---|---|
| IS-IMMEDIATELY-SUPERIOR-OF | | |
| IS-NEAR-OF | | |
| IS-POSTERIOR-OF | IS-ANTERIOR-OF | |
| IS-INFERIOR-OF | | |
| IS-PROXIMAL-OF | | |

#### 5.2.3.5 Mereotopological relationships

The following links combine spatial relationship and parthood:

| | | |
|---|---|---|
| HAS-MATERIAL-PART | IS-MATERIAL-PART-OF | |
| HAS-LINEAR-DIVISION | IS-LINEAR-DIVISION-OF | restricted to objects with prominent orientation towards length (eg tubes, long bones, …) |
| HAS-LAYER | IS-LAYER-OF | |
| HAS-INNER-LAYER | IS-INNER-LAYER-OF | |
| IS-SPECIFIC-LAYER-OF | | |
| HAS-TANGENTIAL-MATERIAL-PART | IS-TANGENTIAL-MATERIAL-PART-OF | part visible from the outside |
| HAS-NON-TANGENTIAL-MATERIAL-PART | IS-NON-TANGENTIAL-MATERIAL-PART-OF | |
| IS-MATERIAL-EQUIVALENT-OF | | is mereotopologically the same thing |
| HAS-PARTIAL-MATERIAL-OVERLAP | | |
| HAS-OUTER-LAYER | IS-OUTER-LAYER-OF | |
| HAS-PROPER-MATERIAL-PART | IS-PROPER-MATERIAL-PART-OF | |
| HAS-CHEMICAL-STRUCTURE-PART | | eg the carbon-ring in hydrocarbons |
| HAS-SUBSTANCE | IS-SUBSTANCE-OF | specifying the stuff of which an object is made |
| HAS-INGREDIENT | IS-INGREDIENT-OF | stuff that is "part" of an object in different form (eg flower in bread) |
| HAS-ACTIVE-INGREDIENT | IS-ACTIVE-INGREDIENT-OF | the pharmacologic active substance in a composite |

#### 5.2.3.6 Complex spatial relationships

Links describing holes and containments of (usually, but not necessarily) three-dimensional) containers

| | | |
|---|---|---|
| HAS-TOPO-INSIDE | IS-TOPO-INSIDE-OF | |
| HAS-INSIDE-CONVEX-HULL | IS-INSIDE-CONVEX-HULL-OF | |
| HAS-INCOMPLETE-FILLER | IS-INCOMPLETE-FILLER-OF | |
| HAS-INEXACT-FILLER | | |
| IS-FILLER-OF | | |
| IS-GEO-INSIDE-OF | | |
| HAS-HOLE | IS-HOLE-OF | |
| IS-PROPER-FILLER-OF | | |

Boundaries of objects:

| | | |
|---|---|---|
| HAS-FREE-FACE | | |
| HAS-BONAFIDE-BOUNDARY | IS-BONAFIDE-BOUNDARY-OF | |

| | | |
|---|---|---|
| HAS-SPECIFIC-AFFECTED-LOCATION | | for pathologies |

### 5.2.3.7 Cause-effect relationships

| | | |
|---|---|---|
| HAS-EFFECT | IS-EFFECT-OF | unspecified type of effect |
| HAS-SPECIFIC-CAUSE | | true cause |
| HAS-CAUSE | | causal process |
| HAS-PRECEEDING-CAUSING-PROCESS | HAS-FOLLOWING-CAUSED-PROCESS | |
| HAS-CONSEQUENCE | IS-CONSEQUENCE-OF | weak cause |
| HAS-INSTIGATOR | IS-INSTIGATOR-OF | unmaterialised effect |
| HAS-PRECLUSOR | IS-PRECLUSOR-OF | causal prevention |
| HAS-CONCESSIVE | | "despite" cause-effect relationship |

### 5.2.3.8 Activity participance

| | | |
|---|---|---|
| HAS-PARTICIPANT | IS-PARTICIPANT-OF | any participant in an activity |
| HAS-SPECIFIC-PATIENT | | undergoes the activity |
| HAS-ACTOR | IS-ACTOR-OF | does the activity |
| HAS-AUTHOR | | passive actor of an activity |
| HAS-AGENT | | purposeful actor |
| HAS-LOGICAL-ACTOR | | unspecified type of actor |
| IS-ACTOR-DIRECTED-PATIENT-OF | IS-PATIENT-DIRECTED-ACTOR-OF | |
| HAS-PROPERTY-SYSTEMIC-MEDIUM | IS-PROPERTY-SYSTEMIC-MEDIUM-OF | |
| HAS-CREATIVE-RESULT | IS-CREATIVE-RESULT-OF | created actee of an event |
| HAS-BENEFACTIVE | | for whom the action is carried out |
| HAS-ACTEE | IS-ACTEE-OF | undergoes the action |
| HAS-PROPERTY-ACTEE | | property of an object that undergoes the action (eg "lengthening of X" acts upon the length of X) |
| HAS-SYSTEMIC-MEDIUM | IS-SYSTEMIC-MEDIUM-OF | participant that must be there or the action is not possible (non-ergative readings of actions) |
| HAS-PRODUCER | IS-PRODUCER-OF | actor that produces something |
| ACTS-SPECIFICALLY-ON | | undergoes the action |

### 5.2.3.9 Motion participance

| | | |
|---|---|---|
| HAS-GROUND | | reference for a trajectory |
| HAS-SOURCE | IS-SOURCE-OF | where the motion starts from |
| HAS-TARGET-DESTINATION | IS-TARGET-DESTINATION-OF | destination |
| HAS-THEME | IS-THEME-OF | object that moves |
| HAS-TARGET | IS-TARGET-OF | destination |
| HAS-CONVEYANCE | | transport vehicle |
| HAS-SOURCE-ORIGINE | IS-SOURCE-ORIGINE-OF | where the motion starts from |
| HAS-PATH-OF-THEME | IS-PATH-OF-THEME-OF | the trajectory |
| HAS-PATH | IS-PATH-OF | the trajectory |
| HAS-SOURCE-CHEMICAL-REAGENS | | starting reagens in chemical reaction |

### 5.2.3.10 Communication and mental processing participance

| | | |
|---|---|---|
| HAS-ADDRESSEE | | to whom something is said |
| HAS-SAYING | IS-SAYING-OF | what is said |
| HAS-PHENOMENON | IS-PHENOMENON-OF | what is experienced mentally |
| HAS-SAYER | IS-SAYER-OF | who says something |
| HAS-SENSER | | who experiences something |

### 5.2.3.11 Specific processes

| | | |
|---|---|---|
| HAS-POSSESSED | IS-POSSESSED-OF | object owned |
| HAS-IN-POSSESSION | | owner |
| HAS-POSSESSOR | IS-POSSESSOR-OF | owner |
| HAS-EXISTENT | IS-EXISTENT-OF | what exists |
| HAS-ABSENCE | | what is absent |

| | | |
|---|---|---|
| HAS-PRESENCE | IS-PRESENCE-OF | what is present |
| HAS-ASSIGNED-NAME | IS-ASSIGNED-NAME-OF | what the name of an object is |
| IS-ASSIGNED-SYMBOL-OF | | what symbol is assigned to an object |
| IS-SYMBOL-OF | | idem |
| HAS-ATTRIBUTE | IS-ATTRIBUTE-OF | what is said about an object |
| HAS-ATTRIBUEND | | about what something is said |
| IS-NAME-OF | | name of an object |

#### 5.2.3.12 Circumstancial relationships

| | | |
|---|---|---|
| HAS-CIRCUMSTANCE | | any non-essential relationship of a process |
| HAS-GENERALISED-MEANS | IS-GENERALISED-MEANS-OF | means by which something is carried out |
| HAS-INSTRUMENT | IS-INSTRUMENT-OF | instrument used in an action |
| HAS-MOTIVATION | | generic motivation |
| HAS-REASON | | backward motivation ("because there was…") |
| HAS-PURPOSE | IS-PURPOSE-OF | foreward motivation ("to achieve …") |
| IS-SIMILANDUM-OF | | similarity |
| IS-ALTERNATIVE-OF | | alternative |
| HAS-EXCLUSIVE | | "without" |
| HAS-INCLUSIVE | IS-INCLUSIVE-OF | "with" |
| HAS-PARTICIPATING-PROCESS | IS-PARTICIPATING-PROCESS-OF | process participating in another process ("walking" and "putting one leg before the other") |
| HAS-OCCUPATION | IS-OCCUPATION-OF | professional occupation of a person |

#### 5.2.3.13 Healthcare related relationships

| | | |
|---|---|---|
| HAS-RESULTING-COMPLICATION | IS-RESULTING-COMPLICATION-OF | |
| HAS-FINDING | IS-FINDING-OF | |
| IS-SERVED-ENTITY-OF | | for nerves, blood vessels, … |
| HAS-FUNCTION | IS-FUNCTION-OF | |
| HAS-CONTRA-INDICATION | IS-CONTRA-INDICATION-OF | |
| HAS-TOXIC-EFFECT | IS-TOXIC-EFFECT-OF | |
| HAS-INDICATION | IS-INDICATION-OF | |
| IS-CONNECTOR-OF | | for joints, fistula's, … |
| HAS-EPONYMIC-ASSOC | | "Cogan syndrome", "Binswanger encephalopathy" |
| HAS-HEALTHCARE-PHENOMENON | IS-HEALTHCARE-PHENOMENON-OF | relates a patient to a finding or disease |
| HAS-RISK-FACTOR | IS-RISK-FACTOR-OF | |
| HAS-PROCEDURAL-APPROACH | IS-PROCEDURAL-APPROACH-OF | approach of a surgical procedure |
| HAS-BRANCH | IS-BRANCH-OF | for nerves, blood vessels, … |
| IS-HYPOTHESIS-OF | | |

#### 5.2.3.14 Object/attribute/value assignment

Series of link-types that relate properties to objects, properties to states, and states to objects:

| | | |
|---|---|---|
| INTER-WE-P-TYPE-TO-WE-ASCRIPTION-RELATION | | generic O-A-V-link |
| HAS-P-TYPE | IS-P-TYPE-OF | assigns a property to an object (eg "temperature") |
| HAS-FEATURE-P-TYPE | | |
| HAS-MODALITY-P-TYPE | IS-MODALITY-P-TYPE-OF | assigns a modality to an object |
| HAS-WE-STATE | IS-STATE-OF-WORLD-ENTITY-OF | assigns a state to an object (eg "cold" to "ice-cream") |
| HAS-WE-FEATURE-STATE | | |
| HAS-SPECIFIC-STATE-OF-WORLD-ENTITY | | |
| HAS-WE-P-STATE | IS-WE-P-STATE-OF | assigns a state to a property (eg "warm" to |

| | | |
|---|---|---|
| | | "temperature") |
| HAS-EXPRESSIVE-P-STATE | | assigns scalability to a property |
| HAS-REALISED-P-STATE | | assigns a scale appreciation to a state (eg "high scale" to "hot") |
| HAS-SELECTOR | | discriminating feature in naming (eg the "left" in "left arm" that does not mean that the arm is positioned left spatially) |
| HAS-SPECIFIC SELECTOR | | |

#### 5.2.3.15 Formal relationships

The following links must be fully understood by the DM

| | | |
|---|---|---|
| IS_A | | formal subsumption |
| DISJOINT | | if 2 entities are disjoint, there cannot exist entities that are (indirect) children of both |
| SAME-AS | | concepts describing the same entity from a different perspective (eg "childbirth" and "delivery") |
| DEFINES-SAME-SITUATION-AS | | concepts referring to same situation but not necessarily the same entity ("firing of a gun", "the bullet leaving the gun", …) |
| IS-CONJUNCTION-OF | | logical conjunction |
| IS-NEGATION-OF | | logical negation |

The following links are internally in the OB important, but can all by understood by the DM as "being associated with"

| | | |
|---|---|---|
| HAS-MINSELECTION | | |
| HAS-MAXSELECTION | | |
| HAS-RANGE | IS-RANGE-OF | |
| HAS-DOMAIN | IS-DOMAIN-OF | |
| HAS-RANGE-OF-DOMAIN | IS-RANGE-OF-DOMAIN-OF | |
| HAS-SPATIO-TEMPORAL-REFERENCE | | |
| RELATION-FROM-PROCESS-TO-ROLE | | |
| ALGOLINKS-ADD-TARGET-TO-SOURCE | | |
| IS-EVENT-RANGE-OF | | |
| IS-PROCESS-REIFICATION-OF | | |
| HAS_ASSOC | | |

#### 5.2.3.16 Foreign ontology related relationships

| | | |
|---|---|---|
| HAS-CCC | IS-CCC-OF | links the concepts of a situated ontology to the core-ontology |

### 5.3. Insert/update/delete functionality

The OB has the necessary functions in the API to load, save, and export data, both at T-Box and A-Box level. The CreateTBox() function creates an empty T-Box structure. If a structure was already created, then it will delete it from the system and create a new one. The CreateABox() function creates an empty A-Box structure. An A-Box-structure is capable of containing application data, the T-Box-structure cannot. If a structure was already created, then it will delete it from the system and create a new one.

Specific functions are available for serialization. The Load() function allows the user to load data from several files and file-types. There are three options: native OB-format (compact and fast), text files in Cassandra notation [21], and an exchange-format from the LinkFactory®.

The Save() function allows saving data to disk both at T-Box and A-Box level.

### 5.4. Run time specifications

The OB has a well documented, generic API. Integrators that want to integrate the OB in other applications can use this API to write wrappers to the need of these host applications.

The semantic dictionaries may contain too much information to be handled properly by the DM and DSE. Eg the current version for the breast cancer referal guideline contains 40 concepts in the situated ontology, but over 30.000 in the core-ontology, and more than 60.000 English terms. When this amount of terms cannot be handled by the speech recognition system, they should be eliminated from the dictionaries.

## 6. References


[1] W. Ceusters, P. Martens, C. Dhaen, B. Terzic, LinkFactory: an Advanced Formal Ontology Management System. Interactive Tools for Knowledge Capture Workshop, KCAP-2001, October 20, 2001, Victoria B.C., Canada (**http://sern.ucalgary.ca/ksi/K-CAP/K-CAP2001/**).

[2] F. Montyne, The importance of formal ontologies: a case study in occupational health. **OES-SEO2001** International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations, Rome, 14-15 September 2001 (**http://cersi.luiss.it/oesseo2001/papers/28.pdf**).

[3] B. Smith, Mereotopology: a theory of parts and boundaries, Data and Knowledge Engineering 20 (1996), 287-301.

[4] B. Smith, A.C. Varzi, Fiat and Bona Fide Boundaries, in Proc. COSIT-97, Springer-Verlag 1997, 103-119.

[5] F. Buekens, W. Ceusters, G. De Moor, The Explanatory Role of Events in Causal and Temporal Reasoning in Medicine, Met Inform Med 1993, 32: 274 - 278.

[6] W. Ceusters, F. Buekens, G. De Moor, A. Waagmeester, The distinction between linguistic and conceptual semantics in medical terminology and its implications for NLP-based knowledge acquisition. Met Inform Med 1998; 37(4/5):327-33.

[7] J.A. Bateman. Ontology construction and natural language. In Proc. International Workshop on Formal Ontology. Padua, Italy, 1993, 83-93.

[8] Jackson B, Ceusters W. *A novel approach to semantic indexing combining ontology-based semantic weights and in-document concept co-occurrences.* In Baud R, Ruch P. (eds) EFMI Workshop on Natural Language Processing in Biomedical Applications, 8-9 March, 2002, Cyprus, 75-80.

[9] J.A. Hendler, Marker-Passing over Microfeatures: Towards a Hybrid Symbolic/Connectionist Model. Cognitive Science 1989 (1) 79-106.

[10] I. Azzini, D. Falavigna, R. Gretter, G. Lanzola, M. Orlandi: *First steps toward an adaptive dialogue system in medical domain. Proc of Eurospeech,* Aalborg (Denmark), September 2001.

[11] F. Brugnara, M. Federico: *Dynamic Language Models for Interactive Speech Analysis. Proc. of Eurospeech,* pp. 1827--1830, Rhodes, Greece, 1997.

[12] R. De Mori: *Spoken Dialogues with Computers*, Academic Press, London, 1998.

[13] D. Falavigna, R. Gretter, M. Orlandi: *Mixed Language Model for a Dialogue System over the Telephone.* Proc. of ICSLP, Beijing, Vol. 1, pp. 585--588, October 2000.

[14] Das, S., Fox, J., Hammond, P., and Elsdon, D. (1997) "Decision Making and Plan Management By Autonomous Agents: Theory, Implementation and Applications". *Proceedings of the 1st International Conference on Autonomous Agents*, California, pp. 276-283.

[15] Fox, J., and Das, S. (2000) "*Safe and Sound: Artificial Intelligence in Hazardous Applications*", **AAAI**, Menlo Park, CA, and MIT Press, Cambridge Mass.

[16] SIGN (Scottish Intercollegiate Guidelines Network) Scottish Cancer Therapy Network (1998) "*Breast Cancer in Women*", SIGN Publication No. 29, SIGN Secretariat, Edinburgh UK.

[17] Austoker, J., and Mansel, R. (1999) "*Guidelines for Referral of Patients with Breast Problems*", NHS Breast Screening Program (NHSBSP), Sheffield UK.

[18] NLM (National Library of Medicine) "*Unified Medical Language System*", http://www.nlm.nih.gov/research/umls/umlsmain.html

[19] Mahesh, K., and Nirenburg, S. (1995) "*A Situated Ontology for Practical NLP*", IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Aug. 19-21, Montreal.

[20] Bouaud, J., Bachimont, B., and Zweigenbaum, P. (1996) "*Processing Metonymy: a Domain-Model Heuristic Graph Traversal Approach*", Proc. of 16$^{th}$ COLING, Jun-Ichi Tsujii (ed), Copenhagen, Denmark, pp. 137-142.

[21] Ceusters W, Waagmeester A, De Moor G. *Syntactic-semantic tagging conventions for a medical treebank: the CASSANDRA approach.* In van der Lei J, Beckers WPA, Ceusters W, van Overbeeke JJ (eds.): Proceedings MIC'97, Veldhoven, The Netherlands, 183-193, 1997.