# Enabling Reasoning on the Web: Performing Simulations of Clinical Situations

*Mercedes Argüello Casteleiro[1], Julio Des Diz[2], Maria Jesus Fernandez-Prieto[3], Rogelio Perez[4], Hilary Paniagua[5], and Andrew Paul Thompson[6]*

[1]University of Manchester (UK), [2]SERGAS (Spain), [3]University of Salford (UK), [4]Universidade de Santiago de Compostela (Spain), [5]University of Wolverhampton (UK), [6]GP Enki Medical Practice (UK)

m.arguello@computer.org

***Abstract*—The transformation of a document-based medical guideline into a computer-based decision support is a time-consuming and error-prone activity. One way to alleviate this burden is by facilitating, as much as possible, the (semi)automatic implementation and further validation of the knowledge intensive tasks embedded within medical guidelines. This paper presents a bilingual (English and Spanish) "proof of concept" simulation framework and computational test-bed, called V.A.F. Framework, that takes advantages of both CommonKADS methodology and Semantic Web technologies (OWL, SWRL, and OWL-S) to enable experiments (simulations of clinical situations) that allow overcoming the main barriers to successfully express medical guidelines in an executable form compatible with Electronic Medical Records (EMRs).To demonstrate how higher integration between EMRs and evidence-based medicine can be accomplished, this paper focuses on the "Acute Red Eye", a clinical ophthalmologic domain known by General Practitioners (GPs) that usually requires the intervention of ophthalmologists (specialised physicians), so medical referral guidelines as well as ophthalmology medical guidelines need to be codified and integrated with EMRs.**

***Electronic Medical Record; Medical Guidelines; Ontologies; Semantic Web; Reasoning services***

## I. INTRODUCTION AND CURRENT MOTIVATION

Since the 1990s many frameworks for modelling *clinical practice guidelines* (also known as *medical guidelines*) in a computer interpretable and executable format have been proposed, such as [1], [2], and more recently [3]. These frameworks concentrate on specific types of guidelines, end-users, and even organisations. In parallel to the flourish of these frameworks over the years, a variety of guideline representation languages have emerged, including PRODIGY [4], PROforma [5], Asbru [6] EON [7], and CLIF [8]. Nowadays, the various guideline representation languages and related frameworks also need to address compatibility with healthcare information systems that aim to be interoperable on nation-wide and even international levels.

Exchanging medical documents over healthcare networks is becoming a reality, as building the e-infrastructures that will enable the sharing Electronic Medical Records (EMRs) of a patient is currently the first priority of the national e-Health roadmaps of many countries. In UK, the NHS Connecting for Health [9] is creating a NHS Care Records Service to improve the sharing of patients' records across the NHS with their consent. In parallel to this initiative, the UK National Institute for Clinical Excellence (NICE) [10] promotes evidence-based medicine and lays down between 10 to 15 document-based guidelines per year that are intended to be disseminated to frontline NHS staff and make them also available to patients. However, despite these efforts, in UK there is still a lack of full integration between current EMRs and evidence-based medicine. To illustrate this: EMIS [11] is a primary care clinical system in the UK that considers each clinical problem of the patient independently and where most of the help provided is document-based guidelines (e.g. NICE medical guidelines) that are accessible separately by means of *Mentor on the web* [12].

Without full integration of medical guidelines with EMRs; the time of General Practitioners (GPs) is wasted in one or both of the following ways: 1) the practitioner, "knowing the guidelines", practices from memory and then (hopefully) prior to the next consultation realises that there is a step or investigation missed out; and 2) the practitioner spends time in each consultation looking at the guidelines to ensure compliance. Either the next consultation is spent caching up, or the initial consultation is longer. The automated integration of medical guidelines with EMRs should obviate this waste of resources and lead to more evidence based practice that would have a beneficial impact on patients' safety.

Standardisation efforts for the Semantic Web have progressed to the point of reaching a level of maturity that allows the integration of ontologies and rules, see [13]. Thus, it is worthwhile to explore to what extend Semantic Web technologies can support automatic reasoning of knowledge-intensive tasks within guidelines. The main goal is to express medical guidelines in an executable form compatible with Electronic Medical Records (EMRs).

Knowledge-intensive tasks have been traditionally addressed by Knowledge Engineering (KE), where the construction of knowledge-based systems can be seen as a modelling activity oriented towards developing structured knowledge and reasoning models. KE has a long tradition of addressing the development of Knowledge-Based Systems (KBSs) by reusing generic components, such as ontologies [14] and/or Problem-Solving Methods (PSMs) [15]. Examples of KE approaches based on reusing PSMs are: the CommonKADS methodology [16], the formal framework UPML [17], and the general-purpose framework PROTÉGÉ-II [18]. However, Semantic Web technologies have not been covered in any of these three classic KE approaches mentioned.

The research study presented here pursues to 1) facilitate (semi)automatic implementation and further validation of the knowledge intensive tasks embedded within the medical guidelines; as well as 2) promoting the integration between EMRs and evidence-based medicine. The current research approach is aligned to the one presented for e-Government in [19]. Both approaches aim to take advantages of both CommonKADS [16], one of the commonly used KE methodologies, and Semantic Web technologies (OWL [20], SWRL [21], and OWL-S [22]). However, a key difference between [19] and the current approach is in terms of the implementation of the reasoning component, where [19] uses Jena. Although, Jena [23] is one of the most popular reasoning tools, the Jena implementation of OWL did not scale for large datasets. In fact, the JVM ran out of memory by 11 million triples [24]. This can become a serious bottleneck, as anonymised primary care data is now widely available for research purposes. To illustrate this: the General Practice Research Database (GPRD) [25] contains coded diagnostic, demographic and prescribing information for over 3 million currently registered patients, and over 8 million historic patients from around 450 practices throughout the UK. Thus, the GPRD has great potential for piloting aid-decision tools; however, its current size can not be underestimated.

Based on the experiments conducted so far, the major benefit is to facilitate overcoming the main barriers mentioned in [26] to successfully express medical guidelines in an executable form compatible with EMRs.

This paper is organised as follows. Section 2 provides an approach overview. Section 3 illustrates the construction of a domain knowledge. Section 4 shows how relevant fragments of medical guidelines and some medical background knowledge can be modelled as SWRL rules. Section 5 provides details about how a knowledge-intensive task can be exposed as a modelling procedure that involves the control and data flow of OWL-S process models. Section 6 outlines the validation of the V.A.F. Framework that is performed by focusing on ‘Acute Red Eye’. Concluding remarks are in section 7.

## II. APPROACH OVERVIEW

As an outcome of the fruitful ongoing collaboration between academics and clinicians from UK and Spain, there is a bilingual (English and Spanish) “proof of concept” simulation framework and computational test-bed, called V.A.F. Framework, that intends to support rapid experimentation to foster formalisation and execution of an increasing number of medical guidelines (including the ones related to common non-life-threatening conditions) on the Web. Towards this aim, the V.A.F. Framework uses Semantic Web technologies as follows: 1) the Web Ontology Language OWL [20] to formally define concepts, relationships, properties and axioms of the domain knowledge; 2) the Semantic Web Rule Language (SWRL) [21] to represent the rule knowledge; and 3) the OWL Web Ontology Language for Services (OWL-S) [22] to formally capture the inference and task knowledge.

Currently there is a lack of health information systems where EMRs and a wide-spectrum of computer-interpretable medical guidelines (including the ones related to common non-life-threatening conditions) are fully integrated. The V.A.F. Framework proposes to reduce the overwhelming cost of deploying and developing pilot health information systems, by allowing *simulations of clinical situations* with two fundamental objectives:

1. To try out different ways of structuring and marking-up the clinical content of EMRs and to promote experiments with a mixture of different biomedical coding terms (e.g. SNOMED-CT [27]). Thus, the V.A.F. framework facilitates comparing different ways to codify the medical information within EMRs, and therefore, it can be seen as a computational test-bed. To achieve this objective, Web-based *simulations of ontology-based applications* are preformed (see figure 1).
2. To perform experiments that enable foreseeing how well EMRs and computer-interpretable guidelines (including the ones related to common non-life-threatening conditions) can be integrated. Furthermore, these experiments enable to verify the accuracy of the interpretation of general scientifically founded statements described in medical guidelines about what should be done. This is a paramount, as medical guidelines are written with a concrete patient’s (group’s) profile in mind; it is extremely useful to try out if what should be done is well suited for a diverse battery of diverse patients. To achieve this objective, Web-based *simulations of ontology-based applications* with reasoning services are preformed (see figure 1).

The V.A.F. Framework also pursues allowing interoperability with the ontology-design and knowledge acquisition tool Protégé [28] (see figure 1). Protégé has been used to author guidelines in EON, CLIF, PRODIGY, and PROforma. Although the benefits of Protégé are undeniable, and it is one of the most widely cited ontology-design and knowledge acquisition tool, ontology developers still face the gap between modelling and implementation. As in literature appears approaches to software development where ontologies have been successfully employed (e.g. see [29], [30], [31]), the current research approach proposes *simulations* as a way to reduce the existing gap between modelling and implementation.

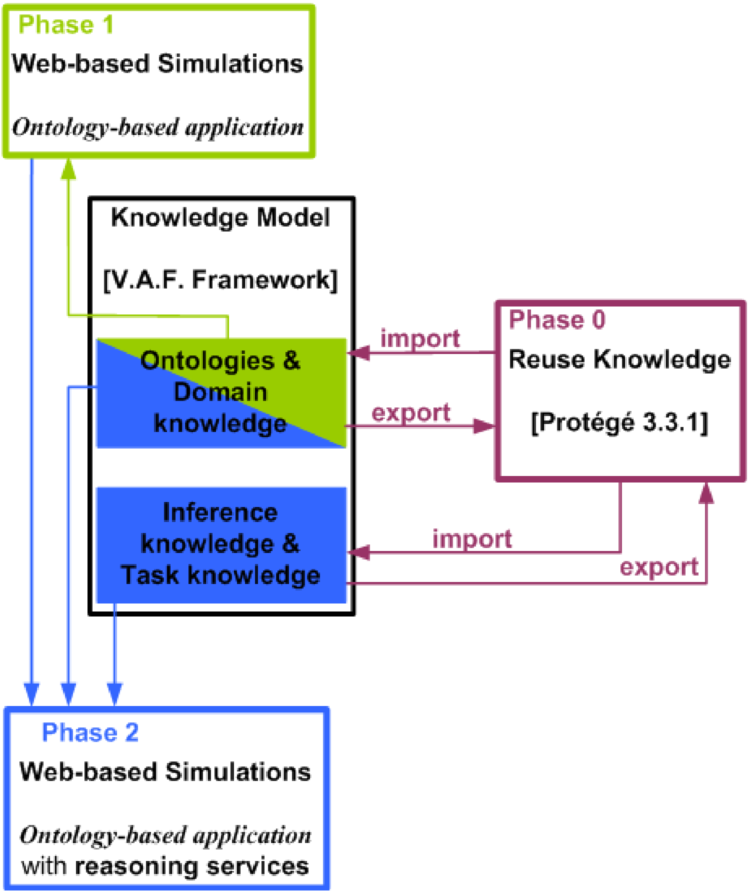


Figure 1. Approach overview

The construction of KBSs by sharing and reusing pre-existing knowledge can be twofold:

1. Reusing domain factual knowledge, i.e. knowledge about the facts in the domain of interest, where following a modular ontological design is highly recommended.
2. Reusing knowledge about the operational aspects of KBSs.

Therefore, ontologies and Problem-Solving Methods (PSMs) arise as promising candidates to facilitate knowledge-component reuse and sharing. As it is stated in [32], "*ontologies define domain knowledge at a generic level, while problem-solving methods specify generic reasoning knowledge*". A PSM can be characterized by [15]: 1) a set of actions that must be carried out in order to solve a problem; 2) a control structure over these actions; and 3) a set of knowledge roles that specify the dynamic and static knowledge needed.

There are different types of ontologies, such as: *generic ontology*; *domain ontology*; *application ontology*; *method ontology*; and *task ontology*. *Domain ontologies* [33] [34] are reusable in a given domain; while *application ontologies* [34] contain the necessary knowledge for modelling a particular application. In CommonKADS [16], application ontologies "*are in fact amalgamates of different types of ontologies*".

In CommonKADS methodology, a *knowledge model* (also known as *expertise model*) contains three knowledge categories [16]:

1. Domain knowledge – it specifies both a schematic description of the domain knowledge structure and the instantiation of the domain knowledge in this structure.
2. Inference knowledge – it describes the most basic reasoning steps.
3. Task knowledge – it provides a specification of which are the goals of the reasoning process (task) and the strategies to achieve these goals (task-method).

In CommonKADS, the development of applications by adapting PSMs consists in [16]:

- Choosing a set of task-methods (task templates) that can satisfactorily solve the application task. Usually, this involves selecting and adapting some methods from a library.
- Constructing the domain-knowledge schema.
- Completing the knowledge model specification.

From the point of view of KE, computer-based decision support in healthcare implies knowledge-intensive tasks. Due to the importance to check progress and alter the development direction if necessary, tests and revisions cannot be postponed to the final stages of knowledge-model construction. So, *simulations of clinical situations* are conceived to validate the knowledge model through its construction (see figure 1).

The V.A.F. Framework (see figure 2) considers three profiles of users: 1) *knowledge engineers* that construct the knowledge model; 2) *clinical experts* who provide examples of EMRs and interpret the knowledge embedded in medical guidelines and protocols; and 3) *clinical users* who perform *simulations of clinical situations* to try out different ways to structure and mark-up the clinical content of EMRs or to verify the accuracy of the interpretation of general scientifically founded statements described in medical guidelines. The black arrows in figure 2 represent the interactions with the V.A.F. Framework, while the grey arrows represent the interactions among the different user profiles.

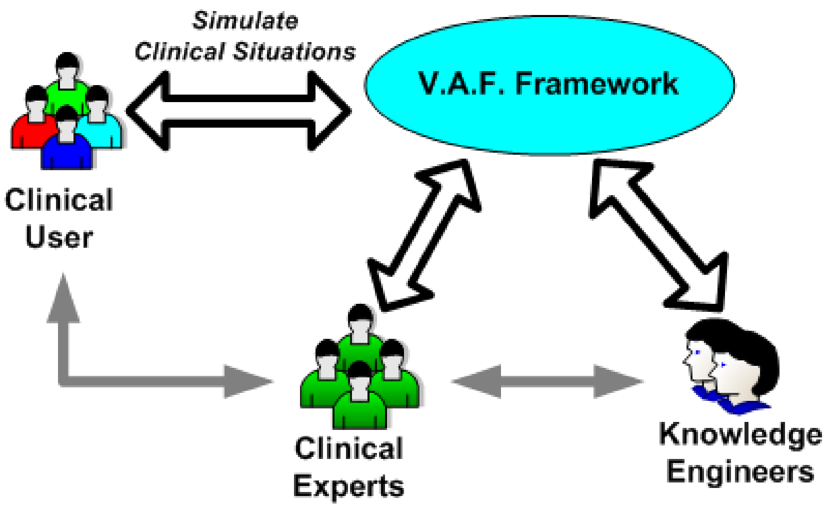


Figure 2. End-users' profiles

As pointed out in [26], there are main barriers to successfully express medical guidelines in a form compatible with EMRs. In [26] four main barriers have been distinguished: 1) differences in granularity and axes of classification; 2) need for inferential abstractions; 3) ambiguity of operational meaning; and 4) differences in encapsulation and form of expression. To overcome these barriers, the current approach considers two phases (see figure 1) where *simulations of clinical situations* can be done:

- Phase 1 – it does not take into account inference or task knowledge, only involves domain knowledge, and therefore, it defines domain concepts and relationships among them. OWL is used to formally describe concepts, relationships, properties and axioms within the domain. This phase is devoted to obtain the domain ontology and the application ontology to deal with EMRs.
- Phase 2 – it takes into account inference and task knowledge. The current research puts a lot of emphasis on this phase to: 1) provide mechanisms to model a medical guideline into a clear and structured representation that can be tested to verify the correct interpretation of the general scientifically founded statements described in the guideline about what should be done, and 2) enable adjusting guidelines in such a way as to keep them up to date, i.e. allowing the necessary mechanisms to facilitate the extension or modification of guidelines in the light of new evidence.

The research study presented follows the three stages proposed by CommonKADS to construct a knowledge model [16]:

1. Knowledge Identification – in this stage, typical activities are: a) to explore information sources that are useful for knowledge modelling, such as lexicons and/or glossaries of domain terms, and b) to list potential components for reusing, such as domain-related components and task-related components.
2. Knowledge Specification – in this second stage the construction of an initial specification of the knowledge model is made, and an initial task decomposition is provided. Typically this stage involves marking up and modelling portions of text from medical guidelines.
3. Knowledge Refinement – in this third stage, the resulting knowledge model is validated by *simulations of clinical situations*. These *simulations* must look and feel as if they were a "real" execution of the knowledge model. Typically, this stage involves knowledge adjustments as well as inserting more knowledge instances.

## III. Domain Knowledge

The first objective is to obtain the *domain ontology* and the *application ontology* to deal with EMRs. Thus, an analysis of the requirements was done. This includes:

- *Purpose*: to support edition of EMRs related to the ophthalmologic clinical domain 'Acute Red Eye'.
- *Level of formality*: initial semiformal modelling using the Unified Modelling Language UML [35], followed by a formal modelling in OWL.
- *Scope*: clinical content relevant for EMRs, where special attention is paid to gather knowledge relative to the ophthalmologic clinical domain 'Acute Red Eye'. Thus, the knowledge acquisition focuses into: a) the information to record during each patient visit; and b) the activities that the physician performs with each patient.
- *Determination of the strategy for the efficient knowledge acquisition*: following in-depth interviews with clinical experts; concrete examples of anonymised EMRs related to 'Acute Red Eye' were provided. These documents record what happens in a patient's visit: the physician should interrogate the patient (anamnesis) and ask about the history of his or her present illness (a set of symptoms which includes the chief complaint and the additional symptoms) and his or her current health state. In addition, if it is the first visit or an emergency, the physician should register as much information as possible about the patient's medical and family history.

The three basic stages of the KE methodology of CommonKADS coupled with a modularised ontology design have been followed:

1. **Knowledge Identification** – in this first stage, several knowledge sources were identified: the library described in [36] that specifies a set of medical concepts; the medical knowledge server Unified Medical Language System (UMLS) [37]; the Galen ontology [38] (from the OpenGALEN project); the International Classification of Diseases ICD-9-CM [39]; the EON ontology [40]; and SNOMED-CT [27].
2. **Knowledge Specification** – the reusable model components selected in the identification stage provide part of the specification. Taking into account the generic categories of medical knowledge proposed in [36] it is possible to obtain a domain ontology that has been broken down into several parts (ontological modules): Generic Patient, Tests, Findings, Clinical States Abstractions, Diseases, and Drugs.
3. **Knowledge Refinement** – in this third stage, the resulting domain model is validated by *simulations of clinical simulations* (phase 1, see section 2), and more concepts from the UMLS Metathesaurus are added to the domain model.

**Diseases** (see figure 3) – It consist of a hierarchy of pathology types structured by the ICD 9 CM [39].

**Generic Patient** (see figure 4) – it models: a) the information to record during each patient visit, and b) the activities that the physician performs with each patient. The physician also tends to perform a set of events, such as laboratory. These events have been modelled through "tests" from [36].

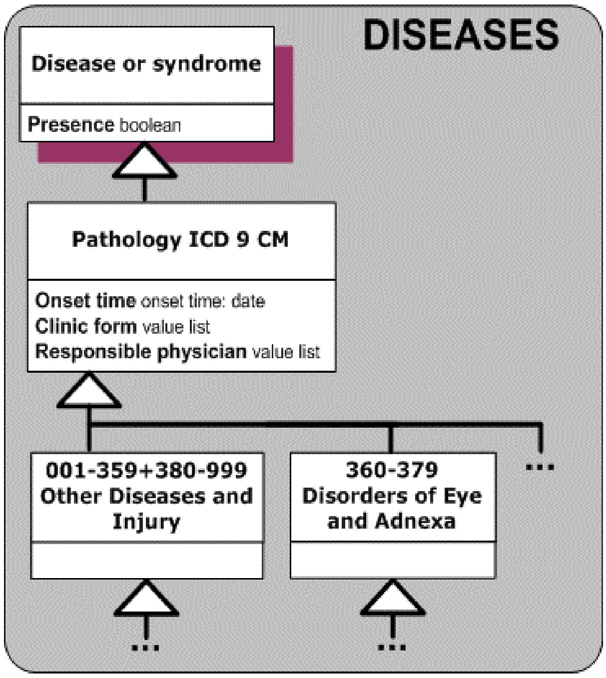


Figure 3. The ontological module Diseases

**Tests** (see figure 5) – In this research study, "tests" from [36] have been identified with "Health Care Activities", which is a type of event from the UMLS server [37]. Under "Health Care Activities", there is a hierarchy of procedures types that was made out of concepts from the SNOMED-CT [27]. Figure 5 shows the association defined between *signs* and *procedures*. The procedures are carried out according to a specific order, and they are intended to gather signs (objective evidence of the presence of a disease or syndrome). Grouping *signs* according to the *body system structure* facilitates their association with procedures, as one-by-one associations become unnecessary.

**Findings** (see figure 5 and 6) – Findings are subjective evidences (symptoms) or objectives (signs) of the presence of a disease or syndrome. Thus, the research presented has differentiated between *findings* and *clinical state abstractions*, exactly as they are collected in [36]. *Clinical state abstractions* describe evidences at a higher level of abstraction, as these are obtained from the findings by following some type of reasoning process.

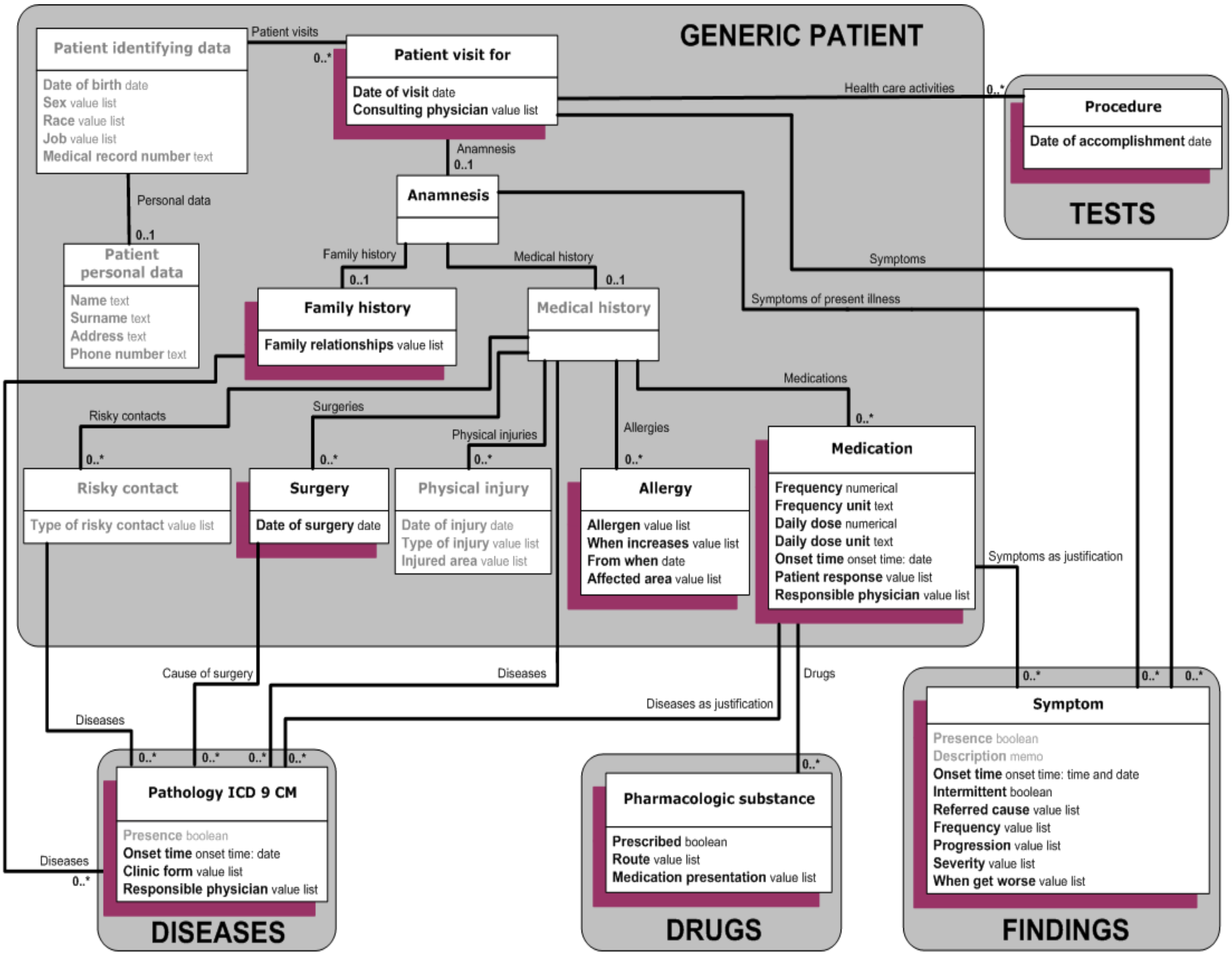


Figure 4. The ontological module Generic Patient

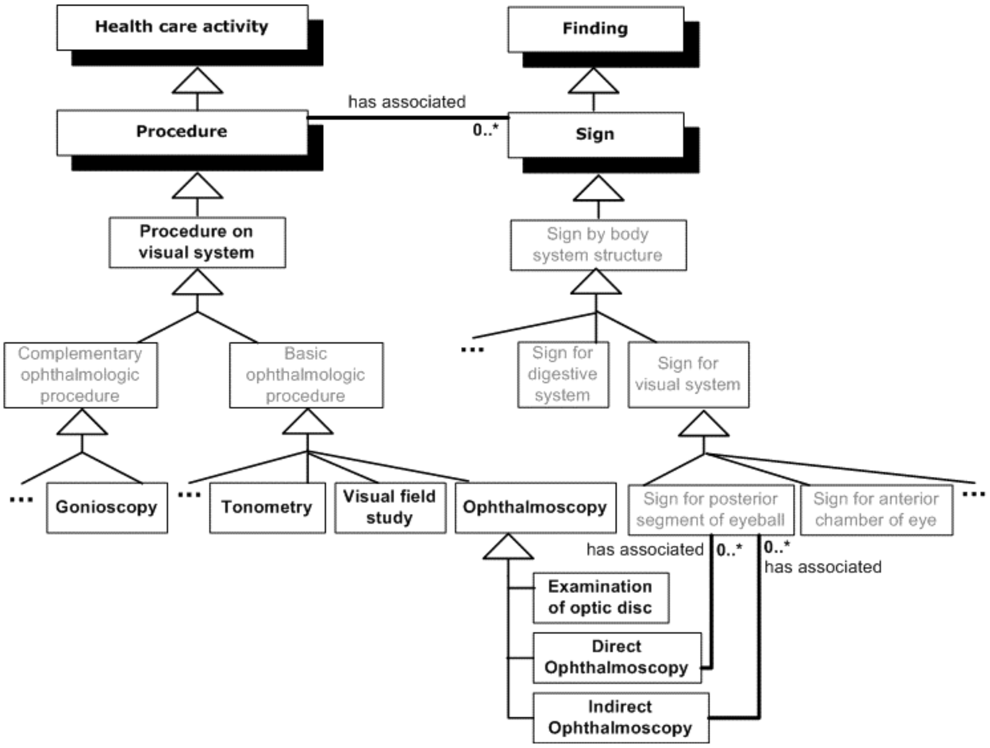


Figure 5. The ontological modules Tests and Findings

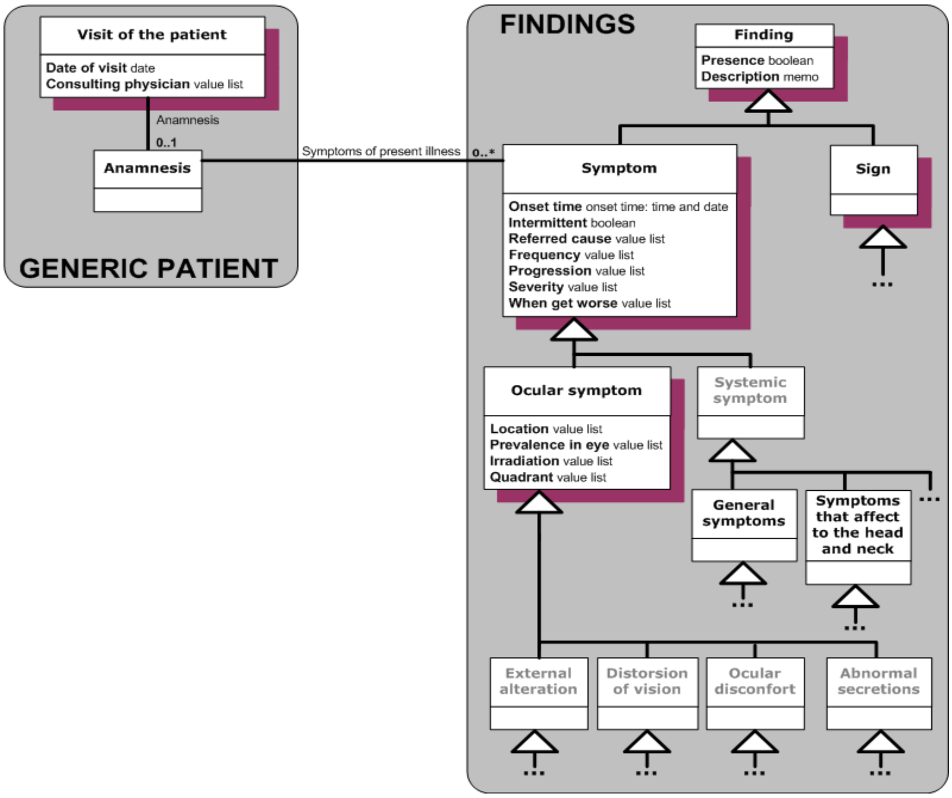


Figure 6. The ontological module Findings

Once *domain ontology* and *application ontology* have been obtained, the second objective is to express medical guidelines in an executable form compatible with EMRs (phase 2, see section 2). Section 4 and 5 show how the rule and task knowledge can be formally captured in SWRL and OWL-S respectively.

## IV. Rule Knowledge

A definition of medical guidelines that is often cited is the one by [41]: "*Clinical practice guidelines are systematic developed statements to assist practitioner and patient decisions about appropriate healthcare for specific circumstances*".

To execute medical guidelines in a computer supported way, the medical knowledge embedded in guidelines that can be in plain textual form, in tables, or represented in flow charts, has to be formalised. Some guidelines can be formalised as sets of rules.

In common with many other rule languages, SWRL [21] rules are written as antecedent-consequent pairs. In SWRL terminology, the antecedent is referred to as the rule body and the consequent is referred to as the head. The *head* and *body* consist of a conjunction of one or more atoms. At the present, SWRL does not support more complex logical combinations of atoms.

SWRL proposes both the antecedent (body) and consequent (head) of a rule consist of zero or more atoms. Atoms can be of the form C(x), P(x,y), sameAs(x,y) or differentFrom(x,y) where C is an OWL DL description, P is an OWL property, and x, y are either variables, OWL individuals or OWL data values [42]. SWRL rules reason about OWL individuals, primarily in terms of OWL classes and properties.

CommonKADS [16] also introduced the idea of multiple rule sets containing rules with similar structure. Taking this into account, the current research proposes to distinguish between 3 rule sets in typical medical referral guidelines: 1) ChiefComplain Rule set (related to activation of medical guidelines); 2) *Manifestation Rule set* (related to expected observations and findings); and 3) *Recommendation Rule set* (related to the clinical management advised). For example, the following text has been extracted from the medical referral guideline [43]:

```
Red Eye

If vision decreased/ pain/
Photophobia/ corneal staining/
perilimbal injection
→
IMMEDIATE Referral

Other cases use clinical judgement
```

The medical knowledge embedded in the textual form mentioned above can be re-written as a set of *Recommendation Rules* in SWRL. One of them is the following:

```
Patient (?x1) Λ Photophobia (?x2) Λ
presentFinding (?x1, ?x2) Λ
Patient_referral_to_specialist (?x3)
→
associatedPriorities (?x3, Immediate)
```

The V.A.F. Framework enables a bilingual use (English and Spanish). Figure 7 shows the above- mentioned SWRL rule from [43], which has been translated to Spanish, and then, loaded into the SWRL editor of Protégé [28].

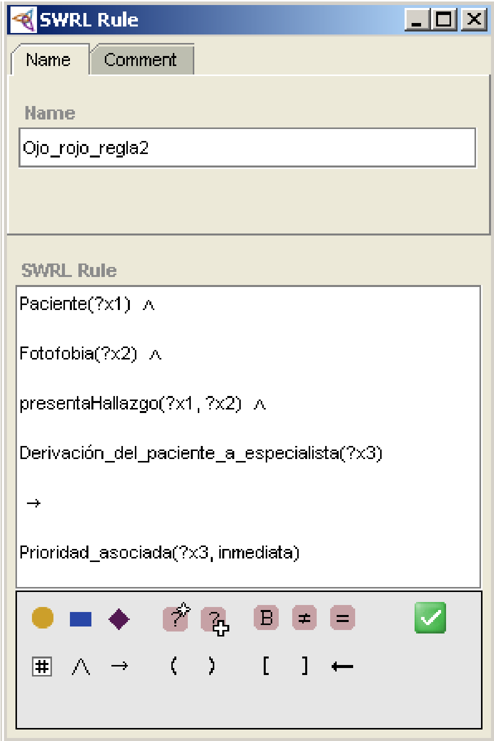


Figure 7. Snapshot of the SWRL editor of Protégé: example of *Recommendation Rule* in SWRL

Not only rule sets containing rules with similar structure can be defined based on the evidence-based knowledge encoded within medical guidelines, the current research study also postulates that some medical background knowledge can be also encoded as rules with similar structure.

To further illustrate how medical background knowledge can be also encoded as rules with similar structure: figure 8 and 9 show examples of associations between diseases or syndromes (e.g. *373.0 Belpharitis*; *371.2 Corneal edema*; *365.22 Acute angle closure glaucoma*) and symptoms. Based on this, it is possible to define a *Causal rule* (figure 10) and a *Manifestation rule* (figure 11). Extrapolating this, to other associations between a disease or syndrome and a set of symptoms is possible to build a *Causal Rule set* (Causal model) and a *Manifestation rule set* (Manifestation model).

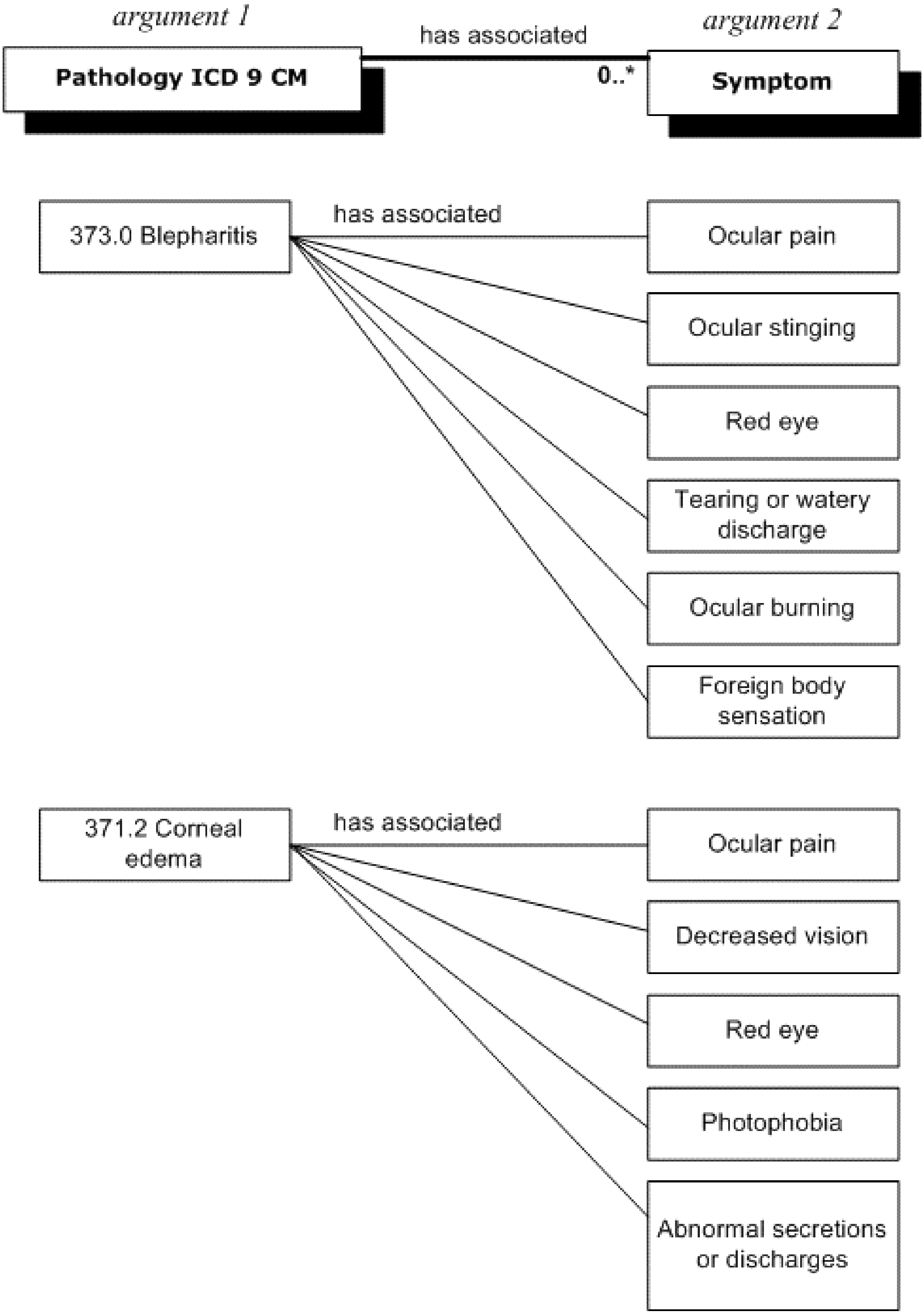


Figure 8. Examples of associations between diseases or syndromes and a set of symptoms

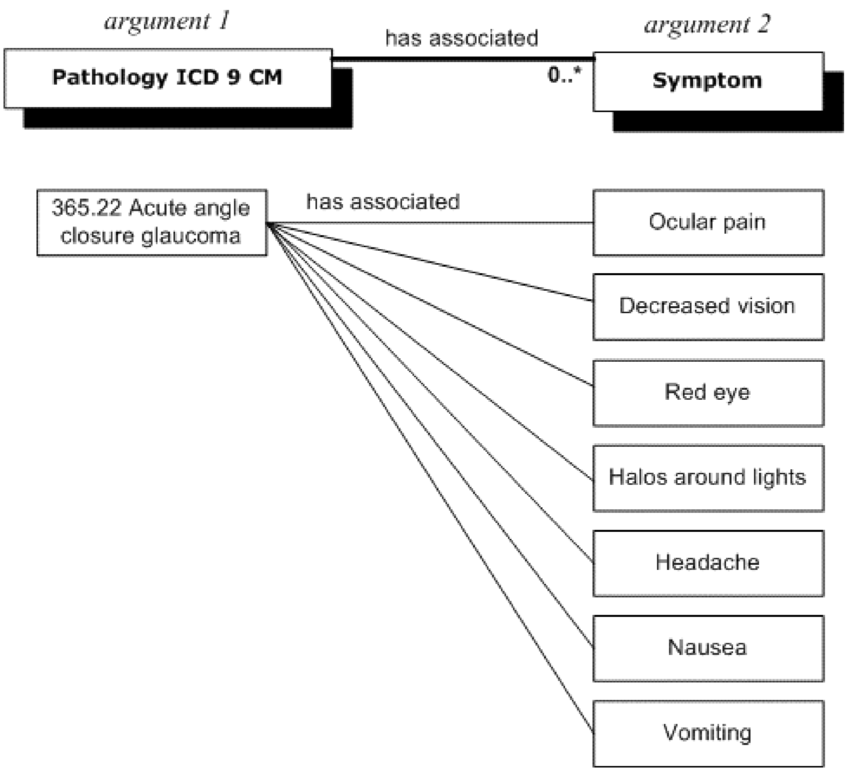


Figure 9. Example of association between one disease or syndrome and a set of symptoms

The pattern to generate a *Causal Rule set* (Causal model) based on the association between diseases or syndromes and symptoms that appears in figure 8 is the following:

```
Patient (?x1) Λ
{argument 2} (?x2) Λ
presentFinding (?x1, ?x2) Λ
Pathology_ICD_9_CM (?x3) Λ
hasAssociated (?x3, ?x2)
→ {argument 1}  (?x3)
```

Similarly, the pattern to generate a *Manifestation Rule set* (Manifestation model) based on the association between diseases or syndromes and symptoms that appears in figure 8 is the following:

```
Patient (?x1) Λ symptom (?x2) Λ
presentFinding (?x1, ?x2) Λ
{argument 1} (?x3) Λ
hasAssociated (?x3, ?x2)
→        {argument 2}  (?x3)
```

The main advantage of the use of the above introduced patterns is to automate the generation of rule sets based on medical background knowledge. These rule sets, obtained from relevant associations, may be used as static roles by PSMs (see next section).

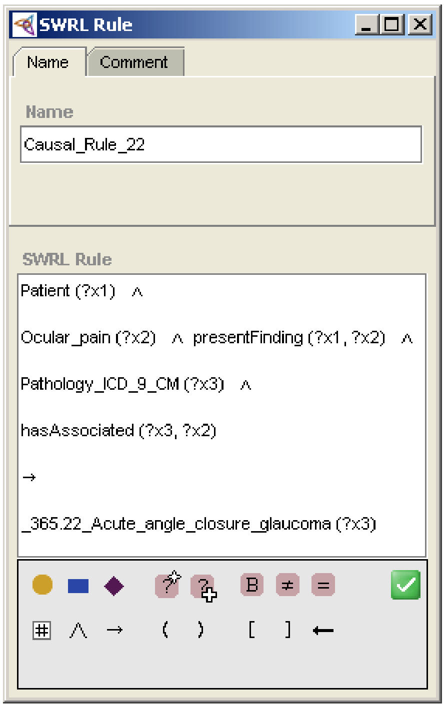


Figure 10. Snapshot of the SWRL editor of Protégé: example of *Casual Rule* in SWRL

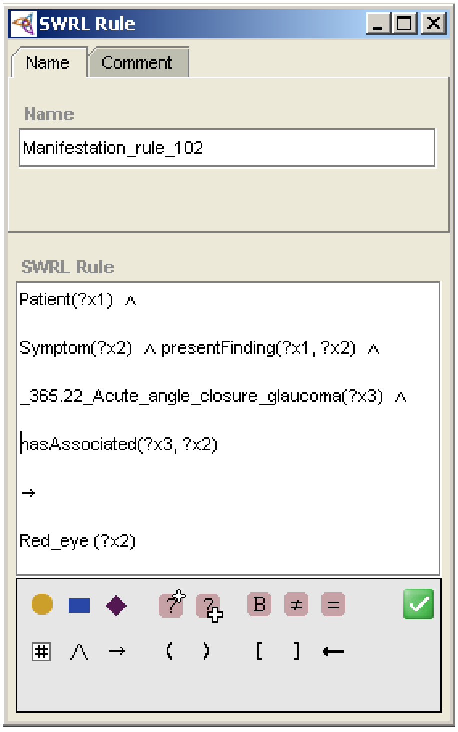


Figure 11. Snapshot of the SWRL editor of Protégé: example of *Manifestation Rule* in SWRL

## V. KNOWLEDGE-INTENSIVE TASKS

According to [32], a Problem-Solving Method (PSM) can be characterized as follows:

- A PSM specifies which inference steps have to be carried out for achieving the goal of a task.
- A PSM defines one or more control structures over these steps.
- Knowledge roles specify the role that domain knowledge plays in each inference step. There are two types of roles: a) *static roles* describe the domain knowledge needed by the PSM; b) *dynamic roles* form the input and output of inference steps.

As it is summarised in [32], most approaches agree that a PSM consists of three related parts, describing: 1) what a PSM can achieve (the PSM's *competence*); 2) how it achieves it (the PSM's *operational specification*); and 3) what it needs to achieve it (the PSM's *requirements/ assumptions*).

CommonKADS [16] embraces the idea of having task-methods (task template) catalogues, where one of the main advantages is to enable reusing "inferences". In CommonKADS *inferences* form the building blocks for a reasoning process.

According to CommonKADS [16], an *inference structure* is an abstract representation of the possible steps in the reasoning process. Figure 12 shows an *inference structure* for a simple diagnosis application that has being adapted from [16]. It should be noted the distinction that CommonKADS makes between an *inference* and a *transfer function*. This distinction is taken into account in figure 12, where 3 inferences (*Cover*; *Predict*; and *Select*) and 2 transfer functions (*Obtain* and *Present*) appear.

Nowadays, service-oriented approaches are gaining momentum. A Web service is a set of related functionalities that can be programmatically accessed through the Web [44]. To promote the re-usability of functionality while providing the necessary support to transform a document-based medical guideline into a computer-interpretable guideline, it is highly beneficial to adopt a Service Oriented Architecture (SOA) where interactivity and interoperability are enhanced. The V.A.F. Framework is built on the foundations that medical guidelines need a common underlying functionality that could be addressed by allowing compositions of Web services into added-value Web services.

The current research study uses the OWL Web Ontology Language for Services (OWL-S) [22] to formally capture the task knowledge. A service in OWL-S is described by means of three elements [22]: 1) the *Service Profile* describes what the service does; 2) the *Service Process Model* describes how to use the service; and 3) the *Services Grounding* specifies the details of how to access/invoke a service. It should be noted, the striking similarities between these three elements and the above-mentioned three related parts of a PSM.

The current approach pays special attention to the *Service Process Model* because it includes information about inputs, outputs, preconditions, and results and describes the execution of a Web service in detail by specifying the flow of data and control between the particular methods of a Web service. The execution graph of a *Service Process Model* can be composed using different types of processes (`AtomicProcess`, `SimpleProcess`, and `CompositeProcess`) and control constructs. Control constructs define specific execution orderings on the contained processes.

In medicine, diagnosis consists of observing a person who is ill and, as a result, determining what is wrong through reasoning and gathering new observations [45]. In terms of performing a differential diagnosis for 'Acute Red Eye', it should be distinguished between [46]: 1) causes of the painless red eye; and 2) causes of the painful red eye. However, a lot of general practitioners have difficulty to make a precise diagnosis.

Thus, recommendations for obtaining a differential diagnosis of the 'Acute Red Eye' include to:

1. Obtain carefully the set of symptoms that are elicited in the *history of present illness*. These symptoms (subjective evidences) have associated the possible diagnoses they represent.
2. Perform a wise selection of ophthalmologic procedures.

The *inference structure* showed in figure 12 is used to aid the differential diagnosis for 'Acute Red Eye'. Examples of a *Causal Rule* from *Causal model* (see figure 12) and a *Manifestation Rule* from *Manifestation model* (see figure 12) can be found in the previous section (see figures 10 and 11 respectively).

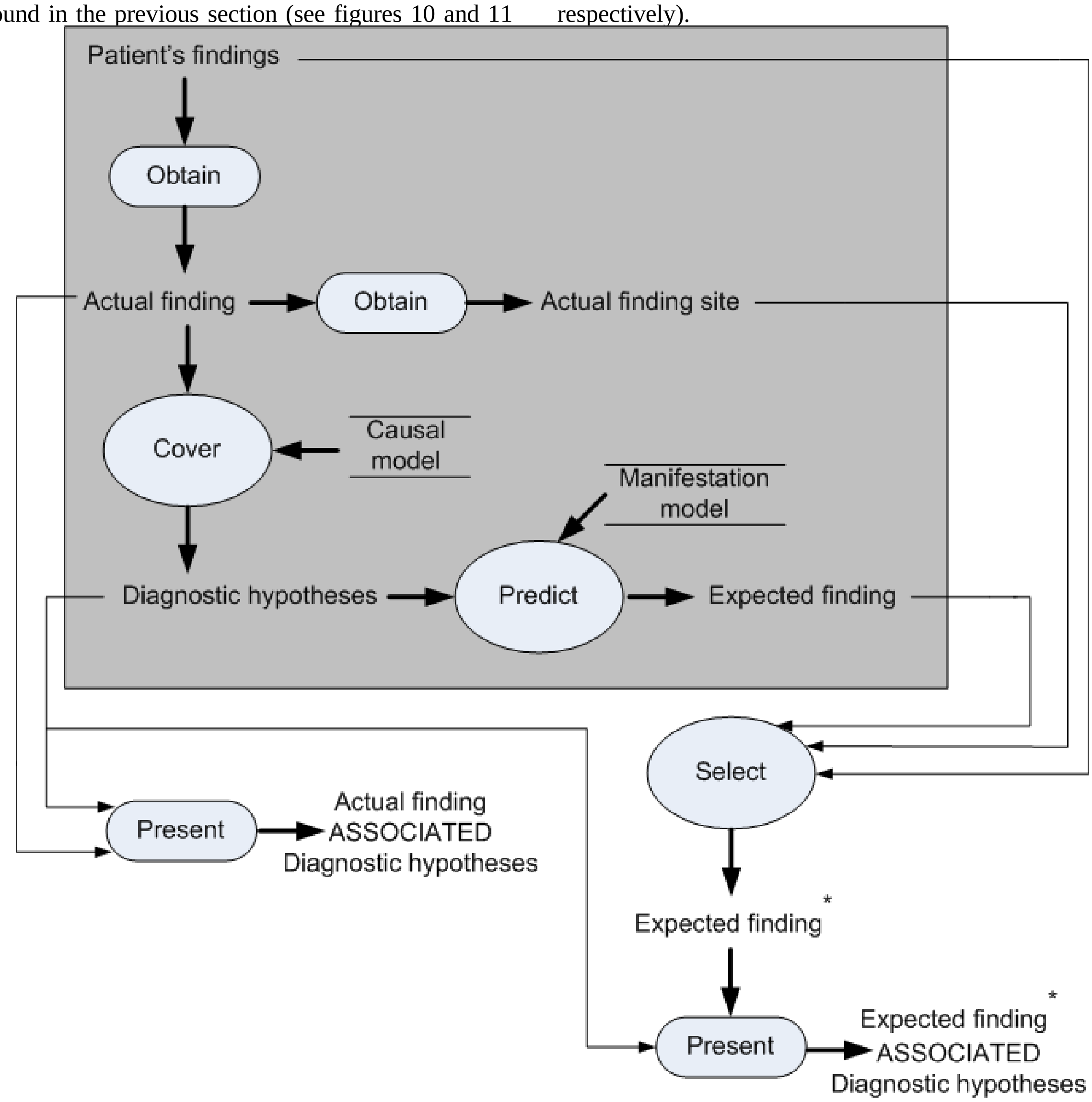


Figure 12. Inference structure for a simple diagnosis application (adapted from [16])

In the current research study, a CommonKADS *inference* or *transfer function* is considered as an OWL-S `AtomicProcess` and implemented as a Web service, where the differences between an *inference* and a *transfer function* outlined in CommonKADS have blurred.

The current research uses OWL-S to describe Web services and takes advantage of the OWL's XML presentation syntax [47] to encode OWL [20] domain ontology fragments and SWRL [21] rule fragments as the inputs and outputs of Web services. In other words, this research study uses Web services that exchange documents with a XML syntax based on a OWL's XML presentation syntax [47] and where a XML document contains OWL [20] domain ontology fragments and/or SWRL [21] rule fragments that are useful to be passed between the services and that may be needed by other components in the same workflow. Therefore, the approach presented outlines the use of the OWL's XML presentation syntax [47] to obtain Web services that provide reasoning support and easily deal with facts and rules.

OWL-S can be used to expose the combination of required activities as a modelling procedure that involves the control and data flow of OWL-S process models. Figure 13 shows the control flow and data flow of a composite process, which is constructed from 4 atomic subprocesses, and that supports the relevant activities after eliciting a "new" symptom in the *history of present illness*.

There is a correspondence between figure 13 and the grey area highlighted in figure 12, where two inferences (*Cover* and *Predict*) and one transfer function (*Obtain*) have been taken into account. The inference (*Select*) and the transfer function (*Present*), both outside of the grey area in figure 12, have been included to make clearer the reasoning process to the clinicians (see next section).

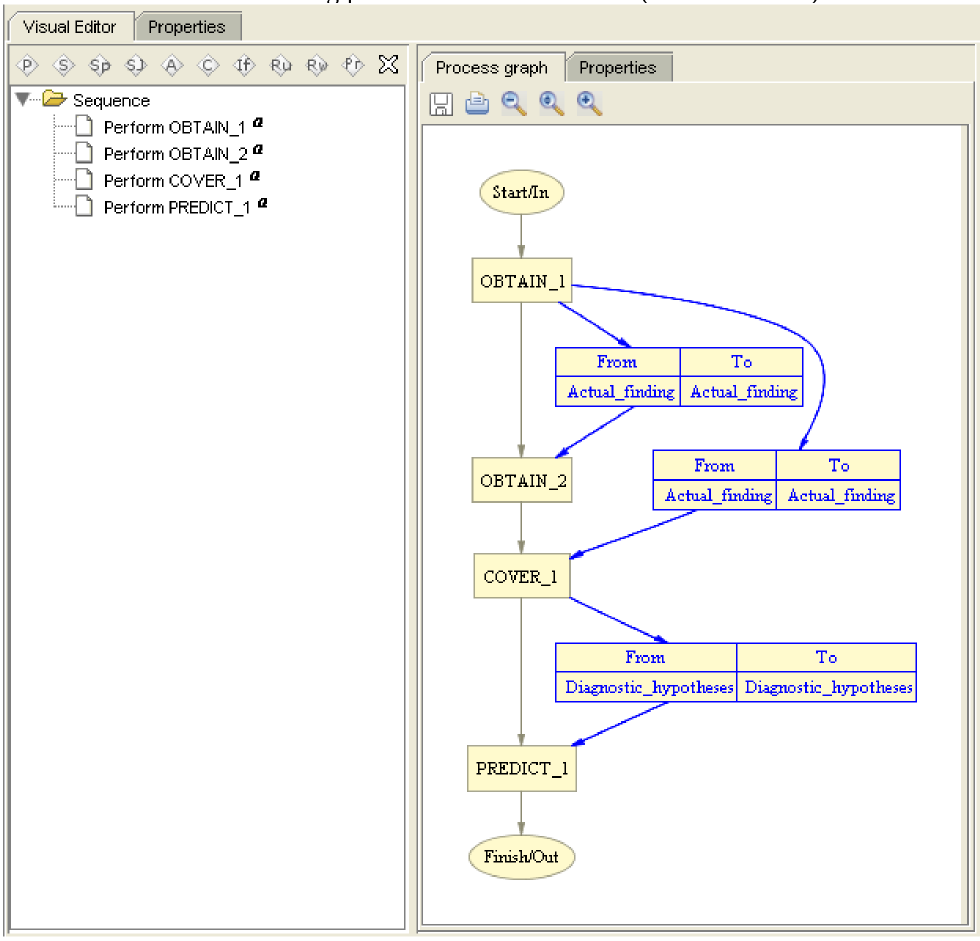


Figure 13. Snapshot of the OWL-S editor of Protégé: control and data flow of an OWL-S process model

## VI. EXPERIMENTAL VALIDATION

The ophthalmologic clinical domain 'Acute Red Eye' [46]: there are many causes of red eye, even in general practice. The principal characteristic features:

- Most red eyes in general practice are not painful.
- The acute, painful, red eye usually needs urgent referral to ophthalmology.
- The very painful, red eye should be referred immediately.

Some recommendations oriented to facilitate diagnosis of 'Acute Red Eye' have been provided [48]. These advices are focussed on helping to the physicians to:

1. Distinguish appropriately between minor and serious causes.
2. Encourage the GPs to treat minor problems themselves, as most cases are benign and can be managed efficiently by them, such as Conjunctivitis, Blepharitis; Subconjunctival Hemorrage; Episcleritis.
3. Refer appropriately those cases that require ophthalmologic consultation, such as Acute Angle Closure Glaucoma, Keratopathy, Uveitis or Scleritis, which must be referred as soon as possible.

As it was mentioned in section 2, two phases can be distinguished to foresee a suitable integration between the recommendations outlined above and EMRs:

**Phase 1** – this phase is devoted to obtain the *domain ontology* and the *application ontology*, so it will be possible to edit EMRs related to 'Acute Red Eye'. In this phase, Web-based *simulations of ontology-based applications* are performed (see figure 14 and 17) where the different ontological modules described in section 3 are considered. The *simulations* aim to a) gather the information to record during each patient visit; and b) take into account the activities that the physician performs with each patient. Both figures 14 and 15 show a selection of symptoms as they are selected by the end-user for the *history of present illness*. Both figures 16 and 17 show a report that is generated on end-user request to summarise the clinical information introduced so far for an anonymised medical record (clinical case).

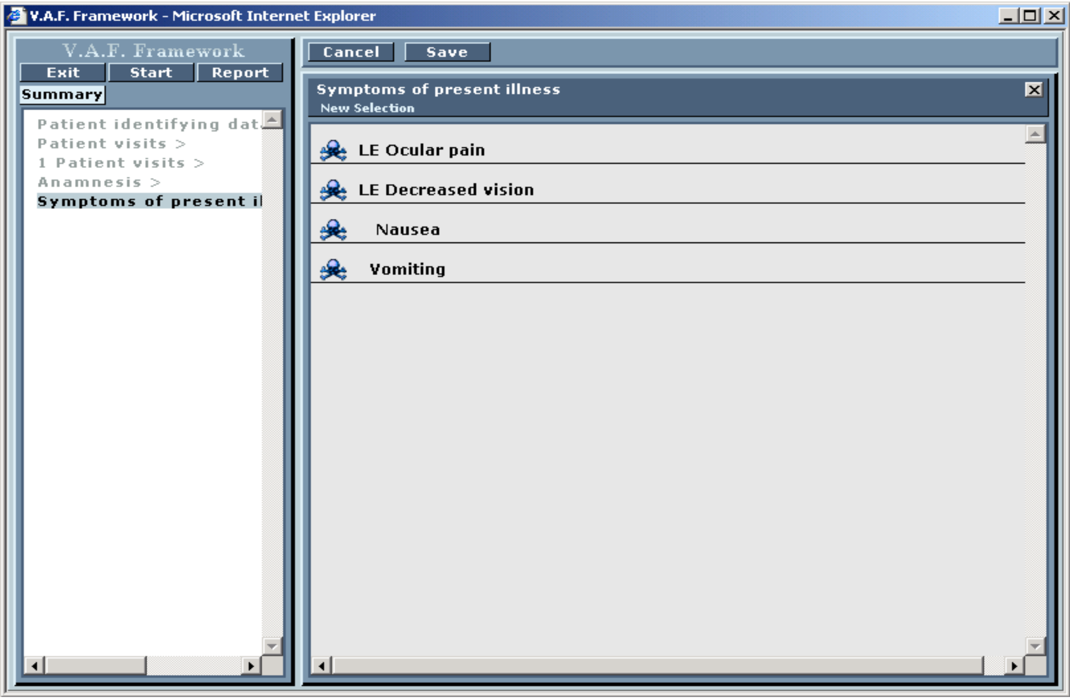


Figure 14. Selection of symptoms for *history of present illness* (English): *simulation* for 'Acute Red Eye'

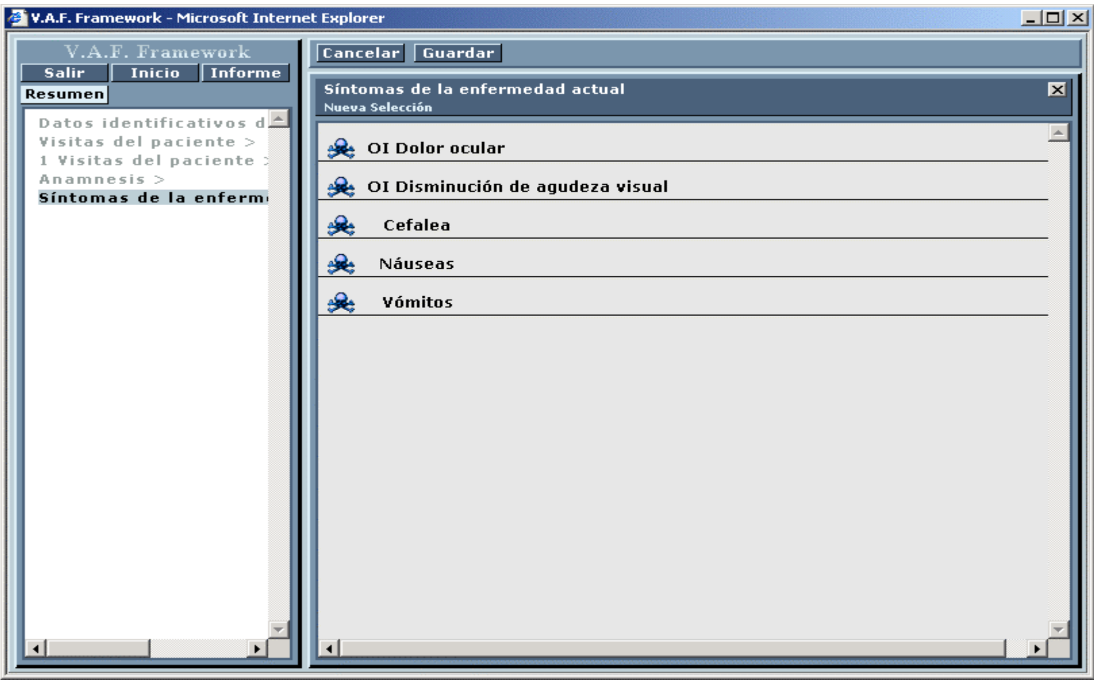


Figure 15. Selection of symptoms for *history of present illness* (Spanish): *simulation* for 'Acute Red Eye'

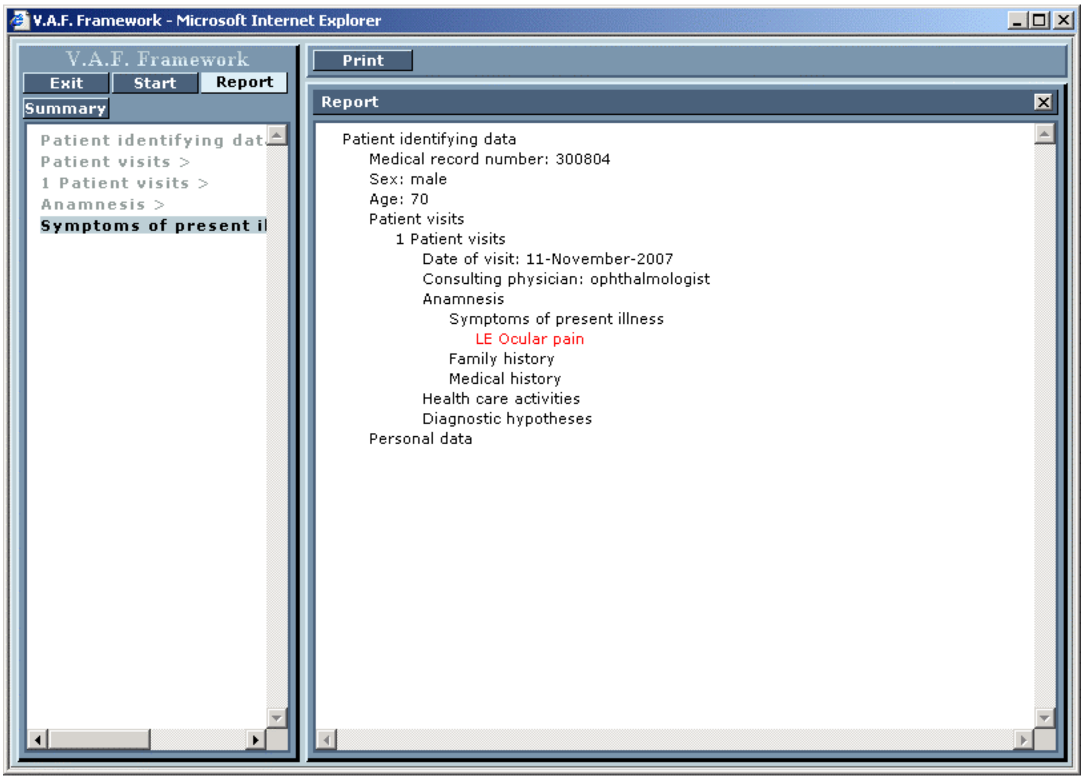


Figure 16. *Summary report* (English): *simulation* for 'Acute Red Eye'

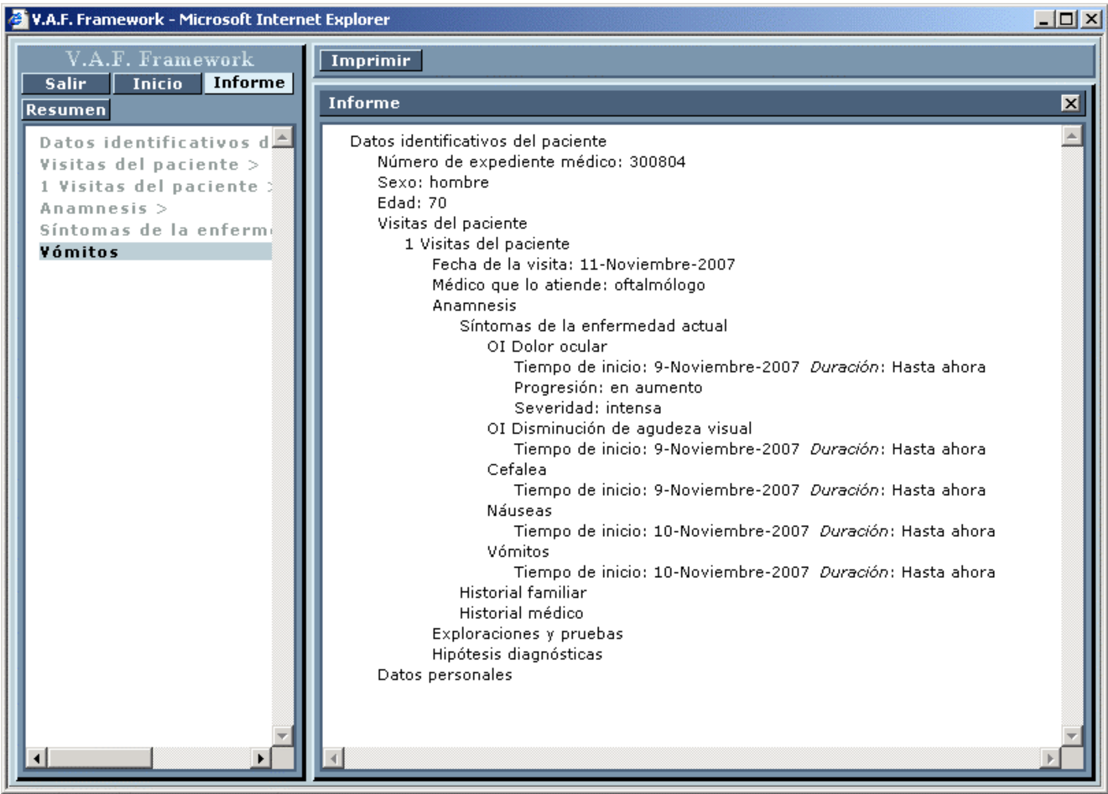


Figure 17. *Summary report* (Spanish): *simulation* for 'Acute Red Eye'

- **Phase 2** – this phase takes into account inference and task knowledge. As it was previously mentioned, the 'Acute Red Eye' may need urgent or immediate referral to ophthalmology. Thus, in the medical referral guideline [43], 'Red Eye' appears as an adult condition where the presence of one or more findings listed lead to the recommendation of patient referral to specialist (ophthalmology). In section 4 appeared the details of how the medical knowledge embedded in the textual form for 'Red Eye' can be re-written as a set of *Recommendation Rules* in SWRL.

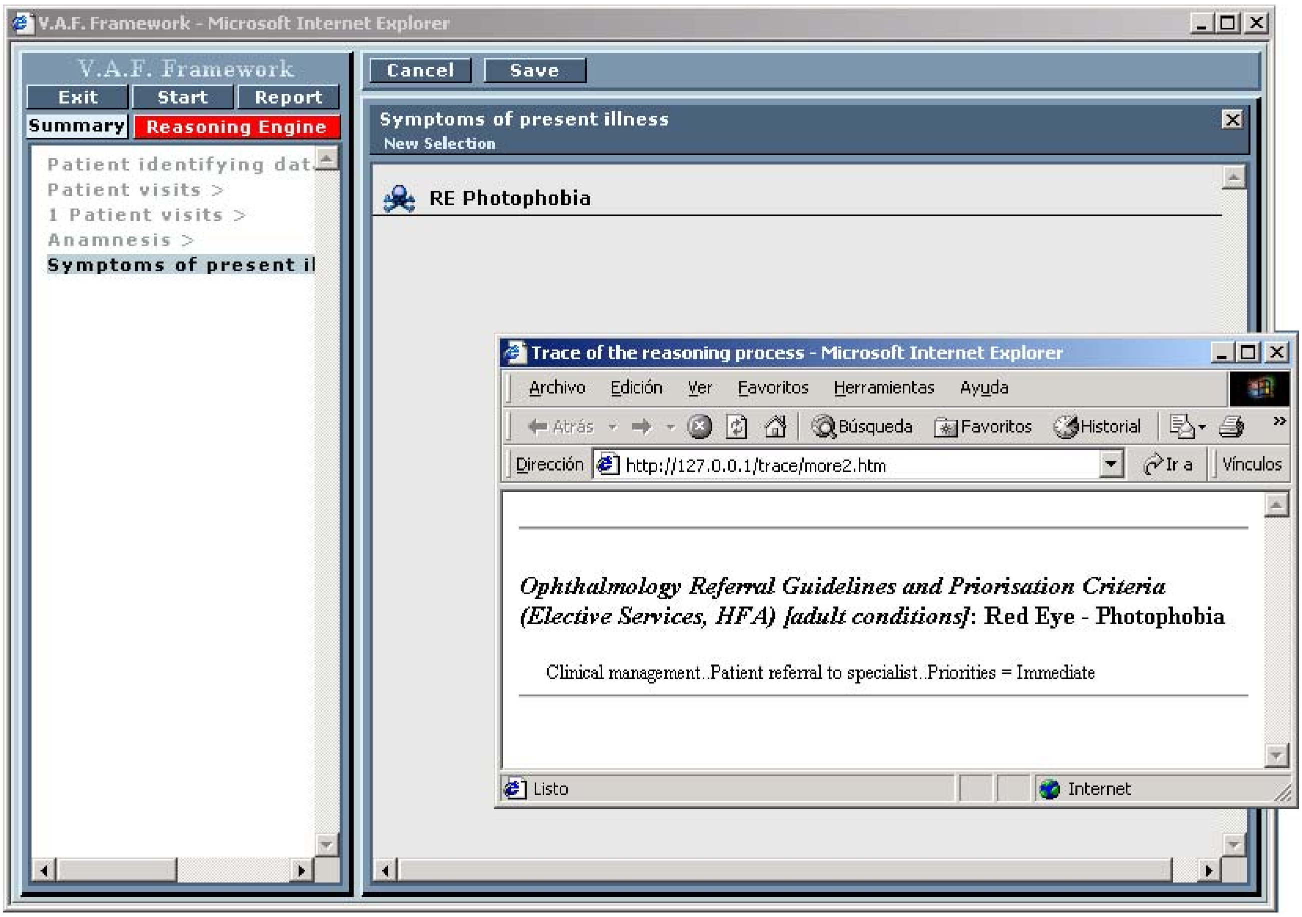


Figure 18. 'New' symptom for *history of present illness*: *simulation* for 'Acute Red Eye'

Figure 18 captures a *simulation* generated by the V.A.F. Framework to test if the medical knowledge modelled from the medical referral guideline [43], about the adult condition 'Red Eye', captures appropriately what should be done in a clinical situation (patient's health conditions) where "*a patient requested to be attended by his family's GP and relates that he presents photophobia and severe redness*".

The current approach tries to not replicate knowledge or functionality unnecessary, and thereby, it was necessary to formulate a strategy to deal with the symmetries of the human body (i.e. left eye; left knee; left lung; etc and right eye; right knee; right lung; etc).

In SNOMED-CT [27], for example, it is possible to find different ways to handle the symmetries of the human body, such as:

- **To replicate knowledge** – It is possible to find concepts that only differ by their body location. For example, there are: '*structure of pupil of left eye*' and '*on examination - Left eye rubeosis iridis*'; as well as: '*structure of pupil of right eye*' and '*on examination - Right eye rubeosis iridis*'.
- **To use specific terms** – There are terms that avoid indicating the precise location. For example: 'unilateral' and 'bilateral' (*unilateral visual field constriction*; *bilateral visual field constriction*); or 'monocular' and 'binocular' (*monocular indirect ophthalmoscopy*; *binocular indirect ophthalmoscopy*).

The current research makes use of the SNOMED-CT concept 'finding site' to specify (locate) the part of the body. For example: '*indirect ophthalmoscopy*' (has) '*finding site*' '*right eye structure*' or '*left eye structure*'. Therefore, the amount of SNOMED-CT concepts can be substantially reduced as the replication of medical concepts that contain terms related to the body symmetries can be easily avoid. To illustrate this: there are 146 SNOMED CT concepts related to 'visual acuity', 4 of them contain the term 'binocular', 54 of them contain the term 'right' or 'R-', and another 54 contain the term 'left' or 'L-'.

Typically, the button *Reasoning Engine* (figure 18 on the left) appears when the V.A.F. Framework has a recommendation. In the *simulation* captured by figure 18, the conditions specified in the antecedent of the *Recommendation Rule* that appears in figure 7 holds, then the conditions specified in the consequent (see figure 7) must also hold, and therefore, when the button *Reasoning Engine* is pressed, the V.A.F. Framework shows the trace of the reasoning process (in this case a primitive reasoning step) in another Web browser (right part of figure 18).

The V.A.F. Framework envisions being useful for GPs as well as for specialised clinicians. Thus, the *inference structure* showed in figure 12 is used to aid the differential diagnosis for 'Acute Red Eye' [46] [48]. Figure 19 captures a *simulation* generated by the V.A.F. Framework to test if the medical knowledge modelled between knowledge engineers and clinical experts (see figure 2) captures appropriately what should be done in the *clinical situation* outlined in the *summary report* that appears in figure 17.

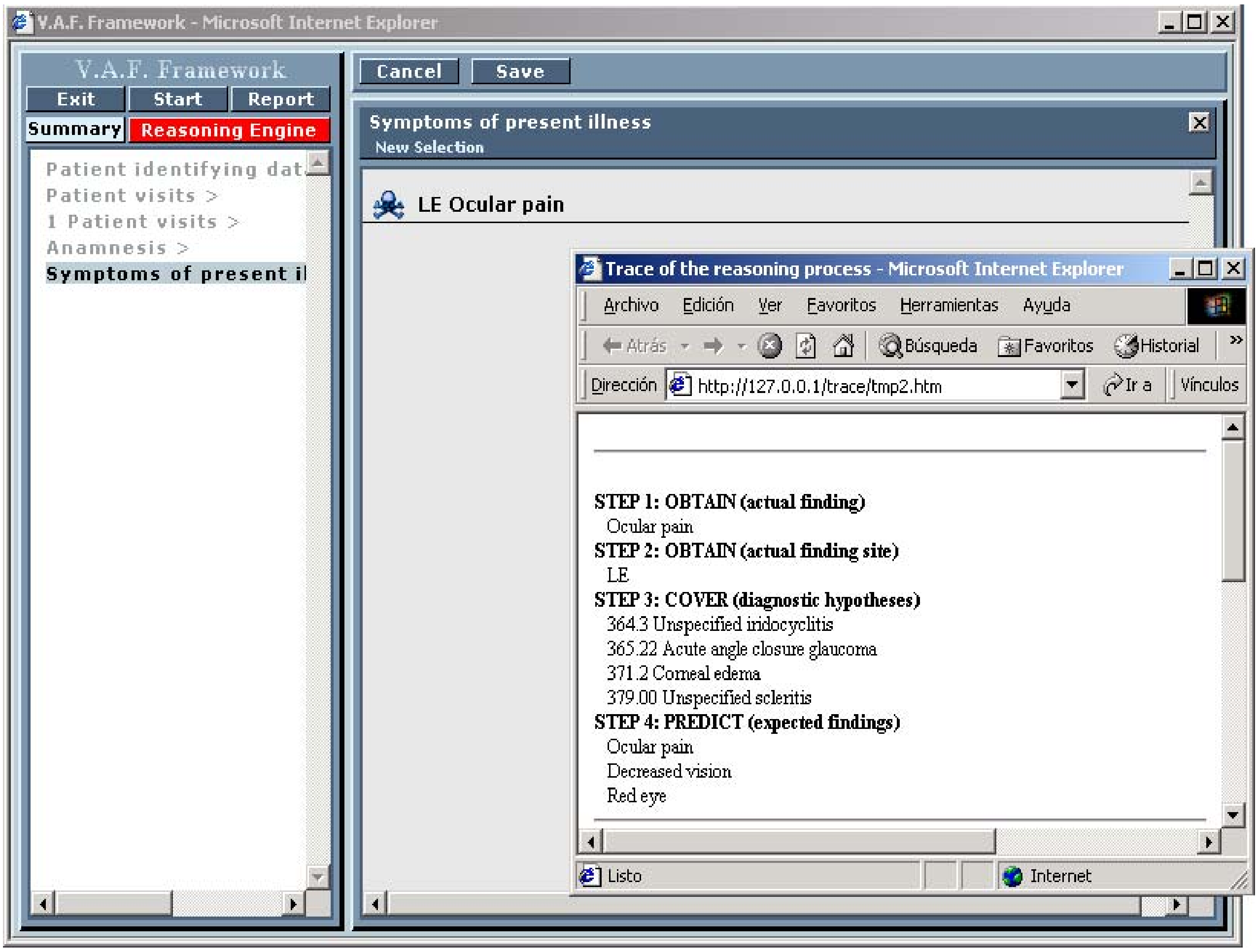


Figure 19. 'New' symptom for *history of present illness*: *simulation* for 'Acute Red Eye'

Due to the possibility of replicating Web services that deal with a certain type of inference (basic reasoning process) as much as needed, it is possible to balance the workload generated by an increasing number of medical guidelines and patients. Thus, the results obtained so far are promising and the V.A.F. Framework is expected to be scalable for ongoing larger evaluation studies.

To increase the clinicians' confidence into the V.A.F. Framework's recommendations, the button *Trace* shows the trace of the reasoning process in another Web browser. However, the initial *inference structure* used (grey area of figure 12) was extended to make clearer the medical background knowledge used for the reasoning services. Therefore, the *transfer function* (Present) and the *inference* (*Select*) were added to the OWL-S process model that appears in figure 13, which corresponds to the grey area highlighted in the *inference structure* of figure 12.

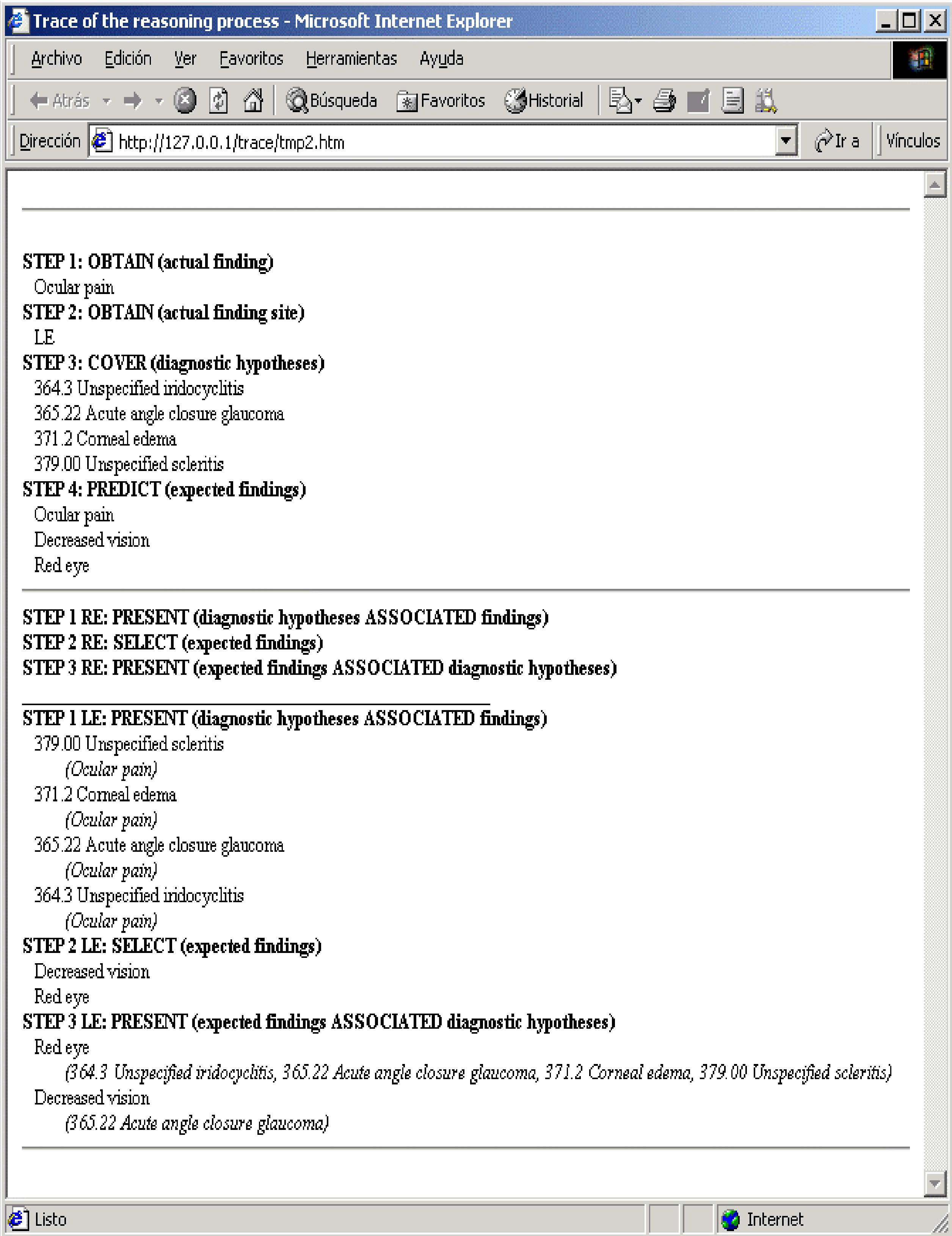


Figure 20. Reasoning trace: simulation for 'Acute Red Eye'

Another advantage of the use of the SNOMED-CT concept ‘finding site’ is to avoid replicating Web services unnecessary. To further illustrate this: when the initial *inference structure* showed in figure 13 (which appears highlighted in a grey area in figure 12) was extended to make clearer the reasoning process, the *inference* and *transfer function* added can be seen as a composite process, which is constructed from 3 atomic subprocesses (*Present-Select-Present*). This composite process can be reused for the ‘right eye structure’ or ‘left eye structure’. Thus, the reasoning trace that is shown in figure 20 reflects that the composite process *Present-Select-Present* is executed twice: once for the ‘right eye structure’ (RE for short) and once for the ‘left eye structure’ (LE for short).

The acceptability of the V.A.F. Framework is assessed by evaluation sessions with clinical professionals who are not familiarise with Semantic Web technologies. During those sessions, the Think-Aloud Protocol [49] was frequently used to gain an outline about the efficacy of V.A.F. Framework. So far the evaluations preformed comprise a small number of test users, although, larger evaluation studies are expected in the near future to promote massive testing and obtain a more detailed feedback. Two highlighted benefits that appear repeatedly in the evaluation sessions performed are: 1) the automatic provision of decision support while introducing clinical information in an EMR; and 2) provide recommendations quietly (the red button Reasoning Engine appears), so clinicians decide if they want to look up the recommendation made or not.

It should be noted that the end-users are kept are un-aware of the use of Semantic Web technologies as the Graphical User Interface remains simple (see figures 14 to 19) and much of the complexity is hidden to the end-user to increase usability and promote the uptake.

One indicator of the success and potential acceptability of the V.A.F. Framework is that the clinicians involved in the evaluation sessions sometimes forget that what they are seeing and interacting with it is just a *simulation* (an execution of the knowledge model) and no real system is behind. In other words, the look and feel of the *simulations* (see figures 14 to 19) could achieve a level of fidelity that is close to create the illusion of a health information system that is up and running.

## VII. Concluding Remarks

The research line followed considers that medical guidelines based on sound scientific evidence can contribute to improve the quality of healthcare, increase patients’ safety, and even reduce the costs of care. However, the transformation of a document-based medical guideline into a computer-based decision support is a time-consuming and error-prone activity.

The current research study is aligned with other research studies that aim to take advantage of both the CommonKADS methodology and Semantic Web technologies (OWL, SWRL, and OWL-S).

The V.A.F. Framework presented here promotes re-usability of functionality; follows a service-oriented approach; and outlines the use of OWL’s XML presentation syntax to obtain Web services that provide reasoning support and easily deal with facts and rules.

The current research study considers the existing gap between modelling ontologies and implementing ontology-based applications, and proposes *simulations* as a way to validate the *knowledge model* and reduce the overwhelming cost of deploying and developing pilot health information systems where a wide spectrum of medical guidelines (including the ones related to common non-life-threatening conditions) and Electronic Medical Records (EMRs) are fully integrated.

Based on the experiments conducted so far, the major benefits of the V.A.F. framework are:

1. To provide the necessary support to transform a document-based medical guideline into a computer-interpretable guideline; and
2. To enable experiments (*simulations of clinical situations*) that allow overcoming the main barriers mentioned in [26] to successfully expressing medical guidelines in an executable form compatible with EMRs.

The acceptability of the V.A.F. Framework is assessed by evaluation sessions with clinical professionals who are not familiarise with Semantic Web technologies. Although these sessions involved a small number of clinicians, the feed-back received is positive and two main advantages have been highlighted:

1. The automatic provision of decision support while introducing clinical information in an EMR; and
2. Provide recommendations quietly (the red button Reasoning Engine appears), so clinicians decide if they want to look up the recommendation made or not.

During the evaluation sessions with clinicians, it was observed that the look and feel of the *simulations* (executions of the *knowledge model*) could achieve a level of fidelity that is close to create the illusion of a health information system that is up and running. This is very promising, as the V.A.F. Framework was conceived to reduce the overwhelming cost of deploying and developing pilot health information systems.

## Acknowledgment

The development and deployment of the *V.A.F. Framework* presented here was inspired by Maria Victoria Alvariño Figuerola and Evaristo Casteleiro López.

## References

[1] C. Gordon, I. Herbert, and P. Johnson, “Knowledge representation and Clinical Practice Guidelines: the DILEMMA and PRESTIGE Projects”. In Medical Informatics Europe (MIE’96), 1996.

[2] S. W. Tu and M. A. Musen, “A Flexible Approach to Guideline Modeling”, Proceedings of the AMIA Annual Symposium, pp. 420–424, 1999.

[3] A. Hommersom, P. Groot, P. Lucas, M. Balser, and J. Schmitt, “Verification of Medical Guidelines Using Background Knowledge in Task Networks”, IEEE Transactions on Knowledge and Data Engineering, vol. 19, pp. 832-846, 2007.

[4] I. N. Purves, B. Sugden, N. Booth, and M. Sowerby, “The PRODIGY Project – the iterative development of the release one model”, Proceedings of the AMIA Annual Symposium, pages 359–363, Washington DC, 1999.

[5] J. Fox, N. Johns, A. Rahmanzadeh, and R. Thomson, “PROforma: A General Technology for Clinical Decision Support Systems”, Computer Methods and Programs in Biomedicine, vol. 54, pp. 59-67, 1997.

[6] A. Seyfang, R. Kosara, and S. Misch, “Asbru’s Reference Manual, Asbru Version 7.3”, Technical Report Asgaard-TR-20002-1, Inst. of Software Technology, Vienna Univ. of Technology, 2002.

[7] M. A. Musen, S. W. Tu, A. K. Das, and Y. Shahar, “EON: A Component-Based Approach to Automation of Protocol-Directed Therapy”, Journal of the American Medical Information Association, vol. 3, pp. 367-88, 1996.

[8] M. Peleg, A. A. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P.Mork, L. Ohno-Machado, E. H. Shortliffe, and R. A. Greenes, “GLIF3: The Evolution of a Guideline Representation Format”, Proceedings of the AMIA Annual Symposium, pp. 645-649, 2000.

[9] NHS Connecting for Health, http://www.connectingforhealth.nhs.uk [May 15, 2008]

[10] NICE, http://www.nice.org.uk/ [May 15, 2008]

[11] EMIS, http://www.emis-online.com/ [May 15, 2008]

[12] Mentor on the web, http://webmentorlibrary.com [May 15, 2008]

[13] S. Park, J. Kang, and W. Kim, “A Framework for Ontology Based Rule Acquisition from Web Documents”, RR 2007, LNCS, vol. 4524, pp. 229-238, Springer, Heidelberg, 2007.

[14] B. Chandrasekaran, J.R. Josephson, and R. Benjamins, “What are ontologies, and why do we need them?”, IEEE Intelligent Systems and their applications, vol. 14, pp. 20-26, 1999.

[15] R. Benjamins, and D. Fensel, “Editorial: Problem-Solving Methods”, International Journal of Human-Computer Studies, vol. 49, pp. 305-313, 1998.

[16] Schreiber, A., H. Akkermans, A.A. Anjewierden, R.Hoog, N.R. Shadbolt, W. Val de Velde, and B. Wielinga, Engineering and managing knowledge. The CommonKADS methodology, The MIT Press, 1999.

[17] D. Fensel, R. Benjamins, E. Motta, and B. Wielinga, “UPML: A Framework for knowledge system reuse”, Proceedings of the International Joint Conference on AI (IJCAI-99), 1999.

[18] H. Eriksson, Y. Shahar, S.W. Tu, A.R. Puerta, and M.A. Musen, “Task modeling with reusable problem solving methods”, Artificial Intelligence, vol. 79, pp. 293-326, 1996.

[19] D. Yang, L. Tong, Y. Ye, and H. Wu, “Applying CommonKADS and Semantic Web Technologies to Ontololgy-Based E-Government Knowledge Systems” ASWC’06, LNCS, vol. 4185, pp. 336-342, 2006.

[20] OWL, http://www.w3.org/2004/OWL/ [May 15, 2008]

[21] SWRL, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/ [May 15, 2008]

[22] OWL-S, http://www.w3.org/Submission/OWL-S/ [May 15, 2008]

[23] Jena, http://jena.sourceforge.net/inference/ [[May 15, 2008]]

[24] P. Parvatikar and K. Portwin, ”Scaling Jena in a Commercial Environment: the Ingenta Metastore Project”, Jena User Conference, 2006.

[25] GPRD, http://www.gprd.com [Jan. 25, 2009]

[26] A. Rector, P. Johnson, S. W. Tu, C. Wroe, and J. Rogers, “Interface of Inference Models with Concept and Medical Record Models”, AIME 2001, LNAI, vol. 2101, pp. 314-323, 2001.

[27] SNOMED CT, http://www.ihtsdo.org/our-standards/ [May 15, 2008]

[28] Protégé, http://protege.stanford.edu/ [May 15, 2008]

[29] L. Li, "Ontological modeling for software application development", Advances in Engineering Software, vol 36, pp. 147-157, 2005.

[30] P. Wongthongtham, E. Chang, T.S. Dillon, and I. Sommerville, "Ontology-based multi-site software development methodology and tools", Journal of Systems Architecture, vol. 52, pp. 640-653, 2006.

[31] S.H. Hsieh and M.D. Lu, "Collaborative Engineering Software Development: Ontology-Based Approach", Intelligent Computing in Engineering and Architecture, LNCS, vol. 4200, pp. 328-342, 2006.

[32] A. Gómez-Pérez and R. Benjamins, "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods", Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), 1999.

[33] R. Mizoguchi, J. Vanwelkenhuysen, and M. Ikeda, "Task ontology for reuse of problem solving knowledge", In N.J.I. Mars, editor, Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing, IOS Press, pp. 46-57, 1995.

[34] G. van Heijst, A.T. Schreiber, and B.J. Wielinga, "Using explicit ontologies in KBS development", International Journal of Human-Computer Studies, vol 46, no. 2/3, pp. 183-292, 1997.

[35] J. Rumbaugh, I. Jacobson, and G. Booch, "The Unified Modelling Language Reference Manual", Addison-Wesley, 1999.

[36] S. Falasconi and M. Stefanelli, “A library of implemented ontologies”, Proceedings of the ECAI Workshop on comparison of implemented ontologies, 1994.

[37] UMLS, http://umlsinfo.nlm.nih.gov [May 15, 2008]

[38] GALEN ontology, http://www.co-ode.org/galen/full-galen.owl [May 15, 2008]

[39] ICD-9-CM, http://icd9cm.chrisendres.com/ [May 15, 2008]

[40] EON, http://www.smi.stanford.edu/projects/eon/ [May 15, 2008]

[41] Clinical Practice Guidelines: Directions for a New Program, M. Field & K. Lohr eds., Inst. of Medicine, National Academy Press, 1990.

[42] I. Horrocks, P. F. Patel-Schneider, S. Bechhofer, and D. Tsarkov, “OWL rules: A proposal and prototype implementation”, J of Web Semantics, vol. 3, pp. 23-40, 2005.

[43] National Referral Guidelines, http://www.electiveservices.govt.nz/guidelines/adult-red-eye.html [May 15, 2008]

[44] B. Medjahed, and A. Bouguettaya, “A multilevel composability model for semantic Web services”, IEEE Transactions on Knowledge and Data Engineering, vol. 17, pp. 954-968, 2005.

[45] C.C. Davey. The red eye. Br J Hosp Med, vol. 55, no. 3, pp. 89-94, 1996.

[46] S. Cartwright and C. Godlee, "General Practice", Churchill Livingstone ed, Elsevier, 3rd edition, 2008.

[47] M. Hori, J. Euzenat, and P. F. Patel-Schneider, "OWL web ontology language XML presentation syntax", W3C Note, 2003. Available at: http://www.w3.org/TR/owl-xmlsyntax/ [May 15, 2008]

[48] J. Bertolini and M. Pelucio. The red eye. Emerg Med Glin North Am, vol. 13, no. 3, pp.561-579, 1995.

[49] Ericsson, K.A., and H.A. Simon, Protocol analysis: verbal reports as data, MIT Press, Cambridge, 1984