Deliverable No. 4.5
Ontology Aggregator Prototype

Grant Agreement No.:	270089
Deliverable No.:	D4.5
Deliverable Name:	Ontology Aggregator Prototype
Contractual Submission Date:	31/01/2014
Actual Submission Date:	31/01/2014

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	***p-medicine***
Project Full Name:	From data sharing and integration via VPH models to personalized medicine
Deliverable No.:	D 4.5
Document name:	Ontology Aggregator Prototype
Nature (R, P, D, O)[1]	P
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	1
Actual Submission Date:	31.01.14
Editor: Institution: E-Mail:	Ulf Schwarz USAAR-IFOMIS Ulf.Schwarz@ifomis.uni-saarland.de

1

R=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

2

PU=Public, **PP**=Restricted to other program participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

ABSTRACT:

This deliverable describes the strategy behind and the implementation of the Ontology Aggregator Tool (OAT) up to the release and testing of a prototype version. We develop this web-based tool in a highly modularized manner to keep it flexible and easily adjustable. It is developed for users who need quick, yet high-quality tailor-made semantic or terminological solutions in cases in which p-medicine's semantic framework is not expressive enough, for instance for providing metadata, data annotations (in conjunction with the Ontology Annotator Tool developed by UPM in task 4.3) or other semantic standardisation purposes. The OAT enables the user to extend p-medicine's semantic framework and compile her own dedicated semantic module. For this a large pool of pre-existing external semantic resources is re-arranged and categorized. Parts of the external resources are taken under the overall semantic structure of p-medicine's ontological framework that is defined by the Health Data Ontology Trunk (HDOT). The OAT can semi-automatically enrich HDOT and its modules by adding concepts, classes or terms from other semantic resources under appropriate HDOT classes on-the-fly. The developers opt for a relatively high degree of automation in this process to make this tool useful for users with very little or no experience in the area of semantic standardisation and ontology. The tool displays necessary information to the user in a concise and understandable manner so that she can make a well-founded ultimate decision on the appropriateness of the generated recommended solutions. We further describe the development cycles of the tool, analyse initial testing results and show how the tool can generate data for its own improvement in use. In addition, the OAT features a user-profile-specific rights management system and an update service to keep HDOT and its modules always aware of the latest developments of important semantic resources and new versions or releases of those.

KEYWORD LIST: ontology aggregation, re-use of semantic resources, ontology evaluation, testing, ontology enrichment, data annotation

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 270089.

MODIFICATION CONTROL			
Version	Date	Status	Author
1.0	15.09.13	Draft	Ulf Schwarz
2.0	19.10.13	Draft	Nikolina Koleva and Luc Schneider
3.0	23.12.13	Draft	Alberto Anguita and Ulf Schwarz
4.0	13.01.14	Final	Ulf Schwarz

List of contributors

- Ulf Schwarz, USAAR-IFOMIS
- Nikolina Koleva, USAAR-IFOMIS
- Luc Schneider, USAAR-IFOMIS
- Alberto Anguita, UPM
- Miguel García-Remesal, UPM
- Andoni Lloret, UPM
- Holger Stenzhorn, USAAR-HOM

Content

Executive Summary

This deliverable describes the strategy behind and the implementation of the Ontology Aggregator Tool (OAT) up to the release and testing of a prototype version. We develop this web-based tool as a fast problem solving service for users who need quick, yet high-quality tailor-made semantic or terminological solutions, for instance for providing semantic metadata, data annotations (provided by the Ontology Annotator Tool developed by UPM in WP4) or other semantic standardisation purposes (such as needed in WP10 by the Biobank Access Framework or within ObTIMA for creating case report form annotations). The OAT enables the user to extend p-medicine's semantic framework and compile her own dedicated semantic module for it out of parts of a large pool of pre-existing external semantic resources by re-arranging and categorizing these parts under the overall semantic structure of p-medicine's ontological framework defined by the Health Data Ontology Trunk (HDOT). The OAT can semi-automatically enrich HDOT and its modules by adding concepts, classes or terms from other semantic resources under appropriate HDOT classes on-the-fly. In this way, a user can continue working with her semantic extension of HDOT straight away. The developers opt for a relatively high degree of automation in this process to make this tool useful for users with very little or no experience in the area of semantic standardisations and ontology. However, the developers explicitly acknowledge that the user should always be able to make the ultimate decision on the correctness and appropriateness of any proposed semantic solution generated by the tool and that the tool should present necessary and concise information to the user to make this an easy and well-founded judgement. We further describe the development cycles of the tool, analyse initial testing results and show how the tool can generate data for its own improvement in use. This is realized by collecting problematic cases, sorting them and defining a workflow for solving them. This is particularly relevant for those cases for which the tool fails to provide good solutions because of a lack in the expressiveness of HDOT. The developers are thus able to curate HDOT according to user needs. In addition, the OAT features a user-profile-specific rights management system and an update service to keep HDOT and its modules always aware of the latest developments of important semantic resources and new versions or releases of those.

Besides the fact that the OAT can generate information itself with its update service, we have designed the OAT in a highly modularized way with many independent components so that it is very flexible and highly adjustable. We believe this to be necessary as new requirements and challenges in semantic standardisation are developing fast and any tool developed in this area needs to be able to be kept up-to-date rather easily.

1. Introduction

Semantic resources (SR) are used in many areas and for many purposes. We have nowadays a plethora of SRs at our disposal such as ontologies, taxonomies, controlled vocabularies, coding systems, thesauri, etc. Most SRs are designed and developed to meet more or less specific challenges and requirements offering sometimes better or occasionally worse solutions depending on the particular task at hand. For the p-medicine project we have decided to use a middle-layer ontology approach with the Health Data Ontology Trunk (HDOT)[3] to provide an overall semantic framework within which we lay out the semantic axiomatisation of key concepts and expressions on quite a general level. We have opted to aim for a strategy and procedure with which we can exploit the plethora of pre-existing SRs in order to enrich HDOT and add more specific concepts to this overall umbrella in such a way that this can be accomplished not only by ontological experts but by anyone in need of tailor-made semantic standardisations. Generally, we consider all users of the p-medicine platform as potential users for the Ontology Aggregator Tool (OAT), but we have certain groups of users in mind such as clinical data managers or clinical trial chairmen we consider more likely to use this tool than others.[4]

We describe a use case for the OAT below in which its connection to the p-medicine annotator tool is explored. It should be noted that this use case is generic in the sense that it can be equally applied to all other scenarios in which HDOT has to be extended so that a tool or service based on it can offer appropriate conceptual representations to its users. Such cases could emerge, for instance in the case report form builder offered with ObTIMA or within the Biobank Access Framework.

The use case we have based the development of the OAT on begins with a user experiencing a lack of a semantic representation in HDOT. Such a case can occur when a user attempts to annotate a data set with p-medicine's Ontology Annotator Tool. The Annotator Tool can only offer terms available in HDOT. If a term is missing in HDOT, the OAT is invoked and leads the user through a simple interactive process at the end of which the missing term can be added to the ontology.

However, we attempt to develop the OAT in such a way that it is easy to use and effective for users with very little or no experience in SRs. That is why we decided to avoid confronting the users with specific data formats such as ***OWL***, ***OBO*** or ***SKOS*** and only present such information to the users, which can easily be grasped and is primarily stated in natural language, for instance in labels.

We only display more technical information such as concept identifiers, e.g. *D018858* is the identifier for Germinal Center in Medical Subject Headings (MeSH), or unique resource identifiers (URIs), e.g. *http://purl.obolibrary.org/obo/UBERON_0010754* is the URI for Germinal Center in Uber Anatomy Ontology (UBERON). The user can choose to display the URIs or to hide them; the labels of concepts or classes are always displayed.

In general, we attempted to keep the interaction between the tool and its users to the bare minimum so that the tool does not ask too much of the user. It thus only requires the user to enter a term or expression she wants to search and in case the search is successful the tool prompts the user to confirm if the term found is actually represented having the semantic content the user had in mind for it. All other steps are processed fully automatically and in the end, an HDOT extension containing the search term (and possible sub-classes, if desired) is generated automatically and at the user's disposal straight away. The tool is supposed to be invoked only in those cases in which a user cannot find an adequate semantic representation in HDOT or one of its modules.

There are some services such as BioPortal[5] and Ontobee[6] that provide central access points for semantic resources, but we believe that the access is just the starting point in a sometimes quite arduous journey

[3] See deliverable 4.1 for an extensive introduction to HDOT

[4] See section 3.1 on use case for more details of expected use

[5] http://bioportal.bioontology.org/

[6] http://www.ontobee.org/

towards making good use of existing semantic resources. When users need to work with these resources in order to find solutions for semantic standardisation and semantic interoperability (for instance when annotating a data schema or designing a case report form) their demands can be stated in two simple questions:

A) Where and how can I find a suitable semantic standardisation for the term or expression I need?

B) How can I easily integrate this suitable standardised term or expression into my information or data management system?

In order to provide a solution for these questions within the p-medicine project, we attempted to translate these simple questions into more detailed challenges for the developers of the OAT:

1) Where and how can the OAT access a large number of existing SRs and analyse their content according to a set of machine-processable criteria in order to assess and structure their content with respect to its appropriateness for the user and for p-medicine's data integration requirements and semantic standardisation?

   The method for a solution needs to be flexible and adjustable so that yet unknown user needs can be accommodated as they develop. This means that the development and implementation of our strategy to analyse SRs (both on the level of analysing metadata about SRs and on the level of analysing the content of SRs) have to be realized in dedicated components so that the OAT can be tuned and refined in many ways and ideally incorporates a pre-defined method for generating required information for its own continuous improvement. This last aspect is very important for our strategy since we assume that we can only discover which improvements become necessary when users actually work with the tool.

2) The re-use of existing SRs is often difficult and arduous. Problems range from where to find them over very heterogeneous construction and design principles and different data formats up to the problem of extracting relevant parts and integrating them in user specific data systems.

3) How can terms be disambiguated? Some terms have more than one meaning. When a tool allows the re-use of existing SRs to standardise a term it must be able to disambiguate those term in a comprehensive, user-friendly way.

4) Some SRs are overlapping, i.e. they partly represent the same domain and partly offer the same terms, concepts or classes, but in different representational models. When a requested term is found in more than one SR that one should be used? There are some terms and concepts that are represented in only one SR, but this fact alone does not ascertain that this SR could or should be used as a solution for the required standardisation of this term or concept.

   These issues are by no means easy to handle even for someone who is familiar with representational systems for semantics and most of the following considerations must be addressed to for finding a good answer to the key question of which (if any) available resource is suitable for a particular representational task.

5) Most domain-specific SRs are not complete in the sense that users cannot be expected to find all necessary standardised terms or expressions in a single SR, if they are able to find them at all. If she needs terms and concepts from different domains it is very likely that she needs to check in different SRs. Thus, we consider it not unusual to find several necessary terms in several different SRs that are often not related and linked to each other (Euzenat et al., 2011). When several terms are found in different resources possible connections between these terms are often lost because these resources are usually not inter-related to each other. This lack of inter-connections is particularly challenging for high-level data integration tasks.

6) How can the quality and correctness of a SR be assessed in the process of re-use? This question is very difficult to address in general, but at least for the context of this project we attempt to provide a framework for the (semi-automatic) evaluation of SRs, which can be exploited, to generate meaningful recommendations for their re-use.

7) Is a SR kept up-to-date and continuously curated?

When we designed the OAT we had these considerations in mind and attempted to make progress towards finding solutions in systems which require only as much interaction with the users as is absolutely essential. The ultimate measure for the required interaction is set by the fact that only the user knows what semantic content she has in mind but is not able to find in HDOT, yet.

OAT's task is to semi-automatically aggregate single parts or whole sub-trees of pre-existing SRs under p-medicine's semantic frame HDOT and thus allow for its user-driven extension in modules.

It is important to bear in mind that the OAT generates recommendations of SRs for users only for the purpose of extending HDOT with one or more specific concepts that are to be re-used from that resource. This requires to apply a bunch of criteria (more than ten) spread over two distinct levels of analysis used to evaluate the appropriateness of a SR always in respect to its suitability for the purpose of enriching HDOT. This sets the OAT clearly apart from other ontology recommendation services like *Magpie* (Sabou et al., 2006) design, *Reflec* (Pafilis et al., 2009) or the *Biomedical Ontology Recommender web service* (Jonquet et al., 2010) that aim at completely distinct recommendation purposes and thus generate recommendations quite differently.

2. The middle-layer ontology strategy and the Ontology Aggregator Tool

In order to understand the overall strategy behind the OAT, we need to briefly recapitulate the principles of the middle-layer ontology design of p-medicine's semantic framework HDOT. This is necessary because the OAT exploits the structural features of HDOT as will be explained later on (cf. Chapter 3).

2.1 Design principles of the Health Data Ontology Trunk

The main structure of HDOT is designed as a biomedical middle-layer ontology in the sense that it specifies upper-level domain independent classes down to the biomedical domain. Meanwhile, it maintains a very general semantic and axiomatic structure, so that it can be further developed and specialized in different interrelated modules for different purposes and applications. We rely on the use of the ***owl:imports*** mechanism in Protégé, following Rector et al. (2012), to design HDOT as a modular ontology.

HDOT's development as a middle-layer ontology is governed by three main related structural considerations in order to achieve the highest level of semantic interoperability between heterogeneous data sources, maintain a high level of ontological soundness and ensure a high degree of expandability:

1. The HDOT level of generality is designed in such a way that the HDOT classes and relations are intended to cover all areas of the health-care domain. Thus, there is a meaningful ontologically well-defined HDOT super-class under which the essential parts and pieces of semantic data descriptions (annotations, metadata) can be directly subsumed or otherwise represented (with the provision that HDOT is still under development);

2. The core ontological structure integrates different modular ontologies at different levels of granularity. Each class is provided with an axiomatisation, which guarantees to the users' workflow high degrees of semantic representation and syntactic reasoning, together with the ability to construct defined classes and composite terms;
3. Ideally, HDOT's modules for specific applications can be obtained by stating further specifications of HDOT classes, i.e. by inserting sub-classes under existing HDOT slots (super-classes).

The hierarchy of the ontology is built on the logical subsumption relation between classes. In the same spirit of Rector (Rector, 2003), multiple-inheritance is not encouraged within the HDOT hierarchy, so that every class is a subclass of at most one upper-level class. Different axioms - mainly inherited from the Basic Formal Ontology 2.0 - are included in the ontology and originate from the following four considerations: (1) To provide machine readable and computable class constructions; (2) To provide ontologically sound relations between classes and the corresponding labels to enable the desired reasoning and inference capabilities; (3) To provide the basis for the composition of biomedical complex classes (e.g. 'blood pressure') by axiomatising the necessary relations between HDOT's constituents; (4) To include relations which bridge different levels of granularity.

2.2 HDOT and the re-use of existing semantic resources

In the development of HDOT we pay particular attention to the re-use of already existing well-founded ontologies. For the time being we include single classes with their URIs manually in HDOT, and we are utilizing key aspects of *MIREOTing*[7] techniques. Ontologies that are re-used in HDOT as a whole are already directly imported. Thus, we mostly resort to OBO Foundry (http://obofoundry.org), ontologies because they are widely used among the scientific community, are continuously under the vigilance of domain experts and ontology designers, and are developed in a modular fashion. Our purpose thus is not the release of new reference ontology, but the development of an ontological framework in which ontologies about different portions of the same domain can be integrated in a multi-granular perspective. This is the reason why as many classes as possible are actually re-used from other, well-established ontologies thus maximizing re-use and compatibility.

Only a few proper HDOT-classes are added in order to ensure a better integration of different components and in order to provide necessary classes for applications, which we are not able to draw from pre-existing resources. At the same time, we pay attention not to alter the intended content of a class or concept (or more precisely the attached term or expression as label) in its integration into a different overall semantic structure. The fact that HDOT is designed such as to maximally re-use existing ontologies sets it apart from other middle-layer ontologies for the biomedical domain such as BioTop (Beißwanger et al., 2008).

Another important factor in our approach is the integration of higher-level concepts or classes from other resources like PATO or *ChEBI* so that by the provision of those interfacing classes an extension of HDOT by further more specific classes from these resources becomes more easily manageable. In fact it is an essential feature of HDOT's design that it is driven by considerations about the requirements for the integration of further concepts and classes under its pre-defined structure. In this manner, it can serve as a resource for tools like the OAT that integrates (semi-automatically) less general classes from a very wide range other semantic resources subsumed under it. This can be achieved only if HDOT is expressive enough to provide semantic representations and logical axioms needed to allow a computation of a recommendation for suitable and ontologically sound HDOT upper class for parts (concepts, classes or terms) of other external resources.

[7] A set of minimal information to reference an external ontology term used by the Open Biological and Biomedical Ontologies organization (OBO Foundry, http://www.obofoundry.org/)

It is not hard to see that this process is easier realized for those parts of SRs from which we already re-used parts in the construction of HDOT itself because matches for appropriate upper-classes for subsumption are readily available in these cases, but the effectiveness of our approach is by no means limited to these resources. In those cases, in which parts from a SR, which is not already re-used in the construction of HDOT, is to be integrated we have implemented several requirements needed to ensure that representational mistakes can be avoided as far as possible.

The problem of keeping the representational correctness and semantic soundness at a very high level is very difficult to address in general, but at least in the context of this project we have set out several evaluation criteria for SRs. The criteria are used to sort and recommend other SRs and are reflected in the design principles of HDOT itself. From this perspective we attempt to use our middle-layer ontology as a semantic ground truth against which we can check design principles of other SRs. Additionally, we incorporate some obvious quality criteria such as the availability of natural language definitions of terms and the character of the represented concept hierarchies. We thus analyse the appropriateness of a candidate-term for integration under HDOT on two levels. First we exploit evaluation criteria we have defined for SRs in general and are able to rank them with these and secondly we analyse characteristics of candidate-terms and the corresponding classes within a specific resource itself by comparing them with HDOT. For ontology evaluation and HDOT design criteria we build on the results of our work described in p-medicine's deliverable D.4.1.

2.3 The main structure of HDOT

HDOT is split it in two main modules:

- The HDOT_CORE, a minimal specification of clinically relevant classes that immediately specify BFO leaf nodes[8]. It currently has 268 classes and DL expressivity SRIQ;
- HDOT_PM (for: HDOT p-medicine module), the direct extension of the HDOT_CORE aimed at providing a large enough basis for the development of further modules. It currently has 358 classes and has DL expressivity SRIQ.

HDOT_CORE extends BFO 2.0 towards the biomedical domain integrating all of OGMS and IAO, plus small segments of PATO, ChEBI, MF, GO, and the OMRSE. Moreover, it includes classes from REX and OBI. Although we commit to the FMA for the ontological representation of anatomy, we have not imported its class URI, because the FMA does not use the format suggested by the OBO Foundry (cf. OBO Foundry principle: FP 003 URIs,). HDOT_PM extends this core by including further classes from the aforementioned ontologies, plus some classes from RID, DOID and MFOEM (references to all mentioned ontologies here can be found in the NCBO (Musen et al., 2012) BioPortal (Whetzel et al. 2011))

When it comes to extending HDOT and its modules we decided to constrain the possible classes which can be specialised by adding sub-classes from other resources to the level of either leaf-node of HDOT_CORE or to classes within an existing module (all classes in these can be extended). On the one hand, we want to keep HDOT_CORE stable as the semantic input resource for the OAT and p-medicine in general. On the other hand, we are convinced that an integration of a class become only meaningful if we can add it in a rather specific category.

Additionally, we can realise a higher level of ontological cleanliness if we pre-define the level of generality. Under this level of generality, we allow extensions as we are able to put enough higher-level constrains on classes (in the form of axioms). These constrains are then inherited by all other sub-classes which become thus even more meaningfully defined and those axioms can be used to formalise the semantic content they are associated with through their labels and definitions, too.

---

[8] For the benefits of using top-level ontologies cf. (Keet, 2011)

3. Ontology Aggregator Tool

In this chapter the Ontology Aggregator Tool (OAT) is described and its workflow set out in detail. The OAT is an implementation of a semi-automatic strategy for extending a middle-layer ontology. The tool offers a modular architecture such that it can be reused with other middle-layer ontologies or different SR repositories and also different interfaces can be plugged in as a front end.

First, we discuss the most important use case of the tool. Next we explain why we need to sort external semantic resources, in which suitable terms/classes will be searched for integration and describe how this is realised. Then the workflow of the OAT and its components are presented. After that we report empirical evaluation, testing and refinements and further analysis of problem solving requirements. Additionally, the user interface is presented and the interaction with the user is explained. Afterwards, we discuss the need to keep the tool and its products, i.e. new user specific modules of HDOT, sensitive to the latest developments in SRs. Thus we present methods to keep the semantic input resource used by the tool up-to-date with respect to changes in external semantic resources. As last point, we describe the interaction of the OAT with other tools in the context of p-medicine.

3.1 Use case

This section describes the use case of the OAT and answers the questions when and how it is going to be used.

OAT is invoked whenever the Ontology Annotator Tool cannot find a suitable class in HDOT or its modules. It suggests at least one candidate concept to be integrated under HDOT so that the annotation can be completed. These candidate concepts are found in ontologies that fulfill quality criteria needed for the meaningful expansion of HDOT. The OAT exclusively re-uses pre-existing concepts or classes already standardised somewhere else.

The life sciences develop, change and improve daily. There are many portals and a huge amount of biomedical ontologies. For the first cycle of the development we decide to use BioPortal for searching appropriate candidate. BioPortal is an open repository, where many biomedical ontologies can be browsed, searched and visualized. Further more one can comment on the ontologies or even map one ontology to another. BioPortal provides access via both Web browsers and Web services to the ontologies it contains. Various formats like OBO, OWL, RDF, RRF, Protégé frames, and LexGrid XML are supported. Anybody who is registered as user can submit an ontology. Offering high quality semantic solutions to p-medicine users is a key desideratum. Therefore we specify a list of formal quality criteria that are suited to our purposes with respect to SRs' ontological soundness, user friendliness and relevance. This is a way to ensure that the quality of p-medicine's semantic framework will not be jeopardised.

In the process of extending HDOT in a module there is a sequence of actions and events that follow each other. As soon as OAT is triggered the search term is entered and a search in BioPortal starts. The search primarily, but not exclusively, uses the term given by the user and delivers a list with hits. After that the search results have to be restricted according to the list of formal quality criteria mentioned above and other information we can use to filter the search results and rank them so that the tool can generate a recommendation of a term for integration. To make the recommendation easily understandable to the user the OAT provides natural language definitions (where available) and also displays full concept hierarchies. The third step requires interaction with the user. The user is asked to accept or reject a recommendation so that she can always make the ultimate decision on the adequateness of the suggested integration. In the simple case of immediate validation a new HDOT-module is created and the task of the OAT is fulfilled. The other situation is that the user disagrees with the recommendation and then another recommendation is displayed if available.

The desired output of the OAT is an extension of HDOT in the form of a modified OWL file. In principle each new module is assigned to a specific user and is tailored to her particular needs.

Figure 1 depicts an UML diagram that describes the use case of the OAT. The included actions are obligatory part of the semi-automatic process and the extending actions are carried out if only if triggered. The triggers are visualised as yellow rectangles and are only available on extend edges.

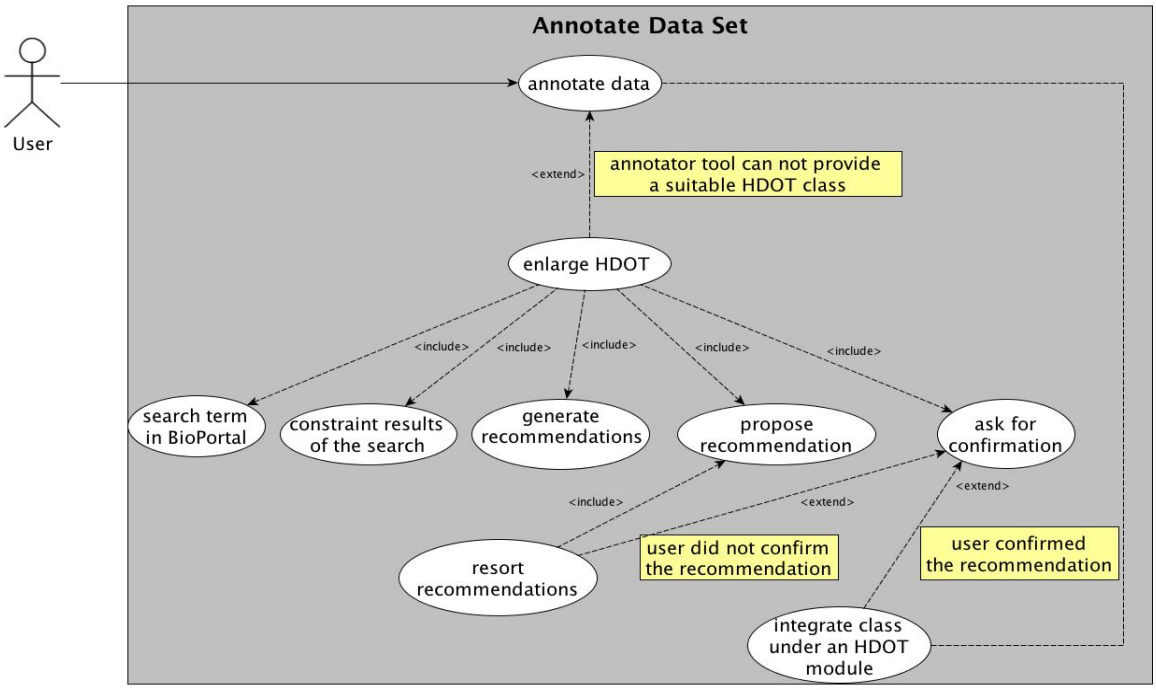

Figure 1: The use case of OAT

3.2 Ontologies Sorting

There are many portals for biomedical ontologies and a huge amount of semantic resources are nowadays available. The quality of the external semantic resources is crucial for the performance of OAT and the nature of the class suggested for integration. Thus it is very important to be able to distinguish *suitable* and *not suitable* ontologies the tool will search in. We decide to use NCBO BioPortal for searching an appropriate candidate that can be integrated under HDOT. The main reasons for choosing NCBO BioPortal ontologies is that most of them are based on Basic Formal Ontology (BFO) just like HDOT which makes integration easier and that HDOT already contains upper-level concepts of some major BioPortal ontologies like ChEBI (see Chapter 2). Furthermore, as we explained in Chapter 2, many NCBO BioPortal ontologies are candidates for inclusion in the OBO Foundry and thus conform to the OBO quality criteria, including the already mentioned use of BFO as a common top-level ontology. Finally, the NCBO BioPortal ontologies have been developed with the support of domain experts, which also guarantees the validity of their content beyond their formal validity.

At the time of writing this report there are 373 ontologies available in BioPortal. If one searches for the term "**kidney**" there are matches found in 45 of the available ontologies. The question that arises here is how to decide, which ontologies are better, i.e. more suitable such that a term that is found is them should be preferred.

We specify a list of *prima facie* suitable ontologies for the integration under HDOT classes, i.e. those whose parts are already used in HDOT. Since these ontologies are NCBO BioPortal ontologies, we rely on the expertise of the domain experts involved in their development for the validity of their content (as mentioned above). In order to be able to consider new ontologies that are contained in the BioPortal repository, we define formal criteria for sorting all available resources in BioPortal. The formal quality criteria are designed by an ontology engineer. Note that the material or content validity of the ontologies is already assumed to be guaranteed by the expertise of the domain experts involved in their development; also, formal quality criteria pertain to the expertise of ontology designers. These formal quality criteria provide constraints for the automatic proposal of classes that originate from resources that satisfy these criteria. This sorting on the ontology level implies that the search results are pre-sorted with respect to the semantic resource they are retrieved from.

Apart from the pre-defined list of acceptable ontologies we apply eight more criteria for comparison on the ontology level in order to be able to position a given ontology on the right position in the pre-sorted list. The criteria that were provisionally selected are the following:

- C1: contained in pre-defined list of acceptable ontologies;
- C2: contained in OBO Foundry;
- C3: date release after 2008-12-31;
- C4: the resource is not flat;
- C5: the resource is not only metadata;
- C6: author of classes is specified;
- C7: classes are documented;
- C8: depth of the hierarchy;
- C9: no classes with one subclass.

Let us briefly explain the justification of this list. First of all, we are of course limited by the metadata on ontologies that are provided by the repository we select the ontologies from, namely NCBO BioPortal; so the above criteria are all NCBO BioPortal metadata. C1 pertains to the fact that we have already included some major NCBO BioPortla ontologies in the design of HDOT and which to have the corresponding HDOT branches extended before adding classes from other ontologies. C2 reflects our adoption of the OBO Foundry criteria for inclusion, in particular the use of BFO. C3 is meant to ensure that the versions of the ontologies that are used are relatively new. C4 as well as C8 corresponds to our concern that the selected resource should provide a rich structure of hierarchies, which is a measure of the degree of organisation *per se*. Note that the more super-classes a class has, the easier it is to identify branches in HDOT with best matches under which to integrate this class. C5 is insofar obvious, as we want to exclude mere sets of metadata. C6, the identification of the author, is important as it means that in principle there (still) is somebody responsible for the ontology's further development, which may increase confidence in this ontology. C7 reflects the intuition that a good documentation is a quality criterion *per se*; furthermore, the presence of labels and definitions provides us information which we can exploit for generating matches. C8 has been discussed under C4. Finally, C9 also corresponds to the idea that rich hierarchies are not only a plausible measure for structure and organisation, but also sources of information to be used in the generation of matches for the automatic proposal of classes to be included to the user.

By using these criteria, we hope to best ensure that the overall quality of HDOT is preserved in the process of its automatic enrichment. Of course, the list of formal criteria provided above is not the final word on the matter. In further stages of the development, we would like to examine if those criteria are

sufficient or if there is need to add more criteria in order to refine the sorting of the external semantic resources. It is important to mention that all criteria can be applied on arbitrary portals that provide access to biomedical ontologies and the respective metadata.

3.3 Workflow

Having the sorting of the available external resources, we can continue with the actual processing towards the automation of the process of recommendation of a good term. Figure 1 shows an overview of the OAT workflow.

In the process of extending HDOT several steps are performed. For each step a dedicated component is implemented.

As soon as the OAT is called, the inputs (search term and list of sorted ontologies) are pre-processed (**Pre-processing**) and then the search in BioPortal starts (**Search term in BioPortal**). The search primarily, but not exclusively, uses the term given by the user and delivers a list with hits. By *hit* we mean here a search result, i.e. a class, with a label or URI that matches the searched term given by the user. The received results are then restricted with respect to the similarity of the query and the searched term. The similarity measure is the string metric "Levenshtein distance"[9] that measures how different are two character sequences in terms of how many insert, delete or replace operations are needed to get the one sequence from the other. The threshold for the similarity is set to 90% since we observed good filtering with it.

Once the hits are filtered according to the similarity we extract the path from a hit to its root in the original ontology. A path to root comprises all super-classes joined to a term or expression and consists of more general concepts or classes that are later used for comparison with HDOT terms or classes (for examples see Figure 3 and Figure 7). We select the best (top) ten hits for further processing. For each of the ten hits an appropriate position for integration in the HDOT hierarchy is searched.

To be able to generate possible recommendation(s), we extract the root paths of the top ten terms and in this way we can compare the hierarchy of the source semantic resource with the HDOT hierarchy. In this manner, we are able to detect the right position for the integration of the new term (module **Locate the right position for an integration, i.e. detect possible recommendation** in Figure 2). Thereby, we compare the classes that are members of the path to root for a hit class with HDOT classes with respect to their URIs and labels as well. Depending on the number of possible recommendations different actions are performed.

1. If there are no recommendations found then a **"Sorry"** message is displayed to the user and a notification is sent to the curators of HDOT. The user is informed that this term will be processed and soon added to the ontology provided that the user searched a meaningful term.

2. If there was exactly one recommendation, then it is shown to the user and she is asked if the found hit shall be integrated.

3. If there were more than one possible recommendation, then further constraints for ranking the recommendations are applied. So the recommendations are sorted and the user is suggested the top/best one.

[9] http://en.wikipedia.org/wiki/Levenshtein_distance

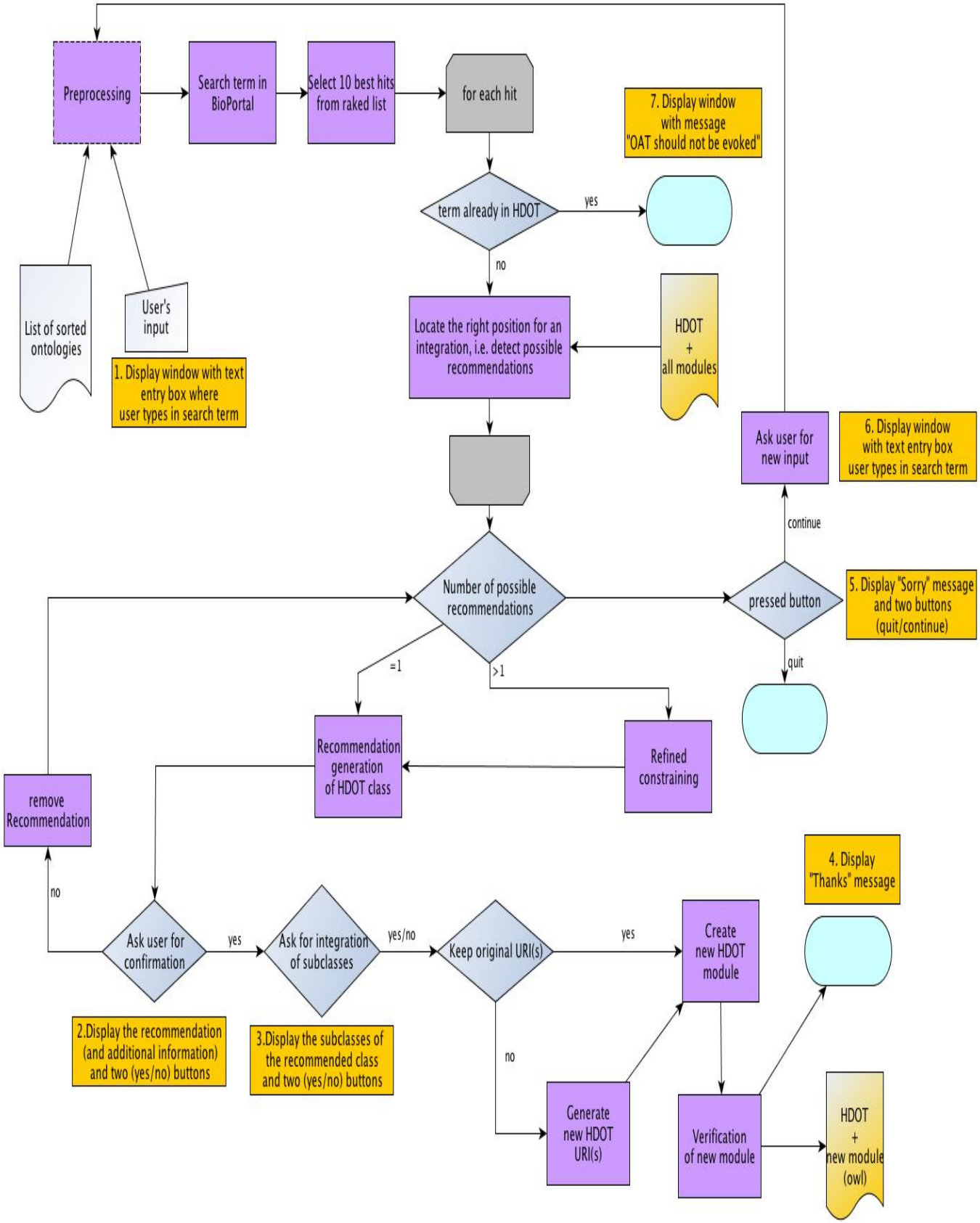

Figure 2: The workflow of the OAT

In the case that at least one recommendation has been generated the user is asked for confirmation. The next step depends on user's choice. Here two situations are possible:

1. If the user rejects the generated recommendation, then the recommendation is removed from the list of generated recommendations and until the list is not empty or the user confirms a recommendation on this list, the next recommendation is displayed.

2. If the user accepts the recommendation, the tool asks the user if she wants to integrate only this class or add also all its subclasses. The classes that are between the hit and the matched parent are automatically included as the recommendation is accepted.

In both cases the tool proceeds with taking the next decision, namely if the original URI(s) is(are) going to be kept or not. If the source ontology of the hit is not on the predefined list then new URI(s) is(are) generated.

After that the new HDOT module is automatically created and verified with the OWL API (Horridge and Bechhofer, 2011). After the integration of each single class on the path a consistency check on the ontology is applied. The module is then saved and **"Thank you!"** message is displayed to the user. Now the user can apply the new HDOT term to carry out the annotation.

During the processing, notifications are sent to the curators of HDOT, whenever:

- new module has been created
- there was no suitable recommendation detected
- there were potential recommendations generated in the core of HDOT (the core of HDOT stays always the same, so no extension in there is allowed)

3.4 Components of OAT

3.4.1 Ontology Sorter

The ontology sorter realises the ontology sorting (cf. Section 3.2) by implementing a chain comparator. The chain comparator combines different simple comparators. Each of these simple comparators implements one quality criterion from the predefined list. In general, a comparator is an operator used to order two objects of the same type at a time. In this case the compared objects are the external SRs. The chain comparator is applied on the list of all ontologies retrieved from BioPortal. In this way, a bubble sort[10] is implemented. With a single comparator for each condition, we offer a flexible construction that allows easy reordering, switching on or off different comparators. This feature facilitates the process of (further) development and exploring different experimental settings. Moreover, the modularity is particularly advantageous for the reusability of the different comparators.

The sorting of the ontologies is done offline, i.e. not while the user is using the OAT but in a previous step. This step is repeated each 6 months because we assume this to be a good time interval for an update of the sorted list because the SRs do not change more rapidly. The output of this component is the fixed order of the SRs hosted in BioPortal and it is given as input for the search engine that queries BioPortal for candidate classes.

[10] http://en.wikipedia.org/wiki/Bubble_sort

3.4.2 Search engine

As a backbone of our tool we use OntoCat (Adamusiak et al., 2011). OntoCat provides a programming interface with high level of abstraction. The project can be used to query public ontology repositories via REST web services. This is exactly what we need for fetching candidate-classes for integration. We provide a wrapper module that searches in the portal according to the fixed order of the ontologies that is automatically created by the previous component. This means that the received candidate classes are already sorted by the ontology they are found in. We apply additional filter to all retrieved candidates to select those that have similarity higher than 90% with user's query. Now the list contains only similar enough candidates and the next step is to extract the paths to the root for the top ten hits in this list.

3.4.3 Root path Extractor

This component carries out the extraction of root path(s), i.e. the path(s) from the found class (hit) to the root class in a given ontology. We extract the root path(s) of the ten best hits contained in the restricted list of results as mentioned above.

For the root path(s) extraction we use the BioPortal's SPARQL endpoint (Salvadores et al. 2012) because the REST services often threw server errors.

There is no direct way for getting the entire path with a single SPARQL query, as it was offered by the REST services. Thus the path from the matched class to its root is collected iteratively by querying for the direct parent and its label. We query not only for the ***rdfs:label*** but also for the ***skos:prefLabel***. The reason for that is that different ontologies use different namespaces and we noticed that those that use ***rdfs*** do not have ***skos*** attributes and vice versa. Though, we put the label as optional attribute in the query. In this way we potentially receive paths with terms lacking a label but the URIs are still compared in such cases later for the generation of recommendations.

According to our observations, the reliability of the SPARQL is better than the one of the REST services. SPARQL directly queries an ontology and do not access database entries that might not be updated. The paths are then further processed when generating a recommendation for the integration of a found hit.

A candidate class may have more than one path to root and this is the case in ontologies that have multiple inheritances. In order to explore all available paths, we implement a depth first search that traverses the semantic graph (the source ontology). It is worth mentioning that the component responsible for the root paths extraction and the one for performing the search are closely coupled. We process only hits for which we receive root path(s). If there is a candidate class for which we cannot get a path to root, i.e. the SPARQL response was empty, then we consider this hit as not suitable and proceed with the next one.

3.4.4 Recommendation Generator

The core component of the OAT is the ***RecommendationGenerator***. In order to generate a recommendation that suggests which of the hits to be integrated under HDOT, at least one class in HDOT that matches a parent class of the hit is required. All paths to root for a candidate class are analysed and compared with the HDOT hierarchy. The search for a match is applied bottom up with respect to the module hierarchy of HDOT. This means that fist the most specific modules are browsed then and it is successively moved to more general modules. This idea is illustrated in Figure 3., where the HDOT module hierarchy is depicted.

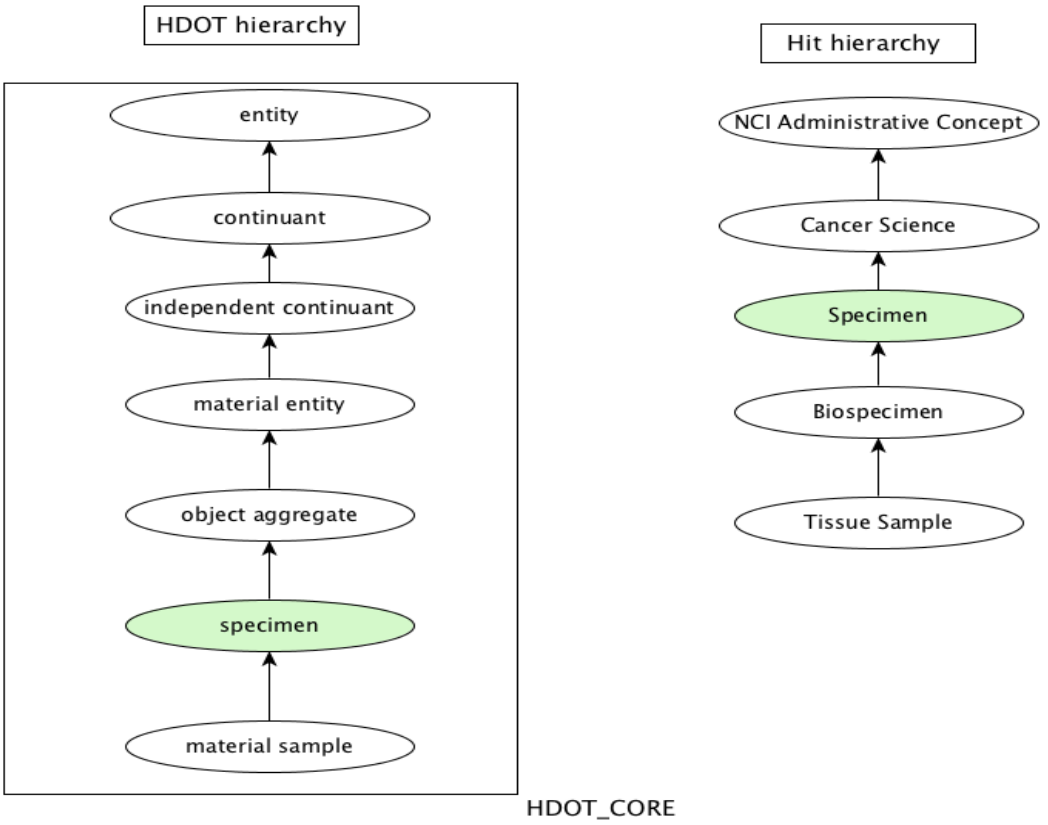

Figure 3: OAT is able to generate only a potential recommendation. The matched class is found in HDOT_CORE but because the core of HDOT is never modified no recommendation for the user is generated.

The core of HDOT is never modified and thus when looking for a match we consider certain conditions that have to be fulfilled, e.g. whether the HDOT-module where a match is found is in the core of HDOT and if so whether the matched class is a leaf node. If yes we are able to generate valid recommendation.

We compare both the URIs and the labels of the classes. In case the labels match but the URIs do not we generate a new HDOT URI but integrate the label of the original class because the recommendation was generated on its bases. There is a dedicated ***URI manger*** for this task. It essentially avoids the unnecessary generation of new URIs. We generate new URIs because we do not trust all available sources.

It is worth mentioning sometimes the label may be misspelled by a typo mistake but actually the class that is denoted with this label would be suitable for an integration given the underlying hierarchy. In order to make the system more robust against such cases, we could apply a spellchecker on the labels before checking for equality.

If a match (of a URI or label or both) is detected and the extension conditions are fulfilled, then a recommendation is generated. However, if any of the extension conditions is violated we generate potential recommendation that is only shown to the curators but not to the users. In this manner, the curators are supported by the analysis of the fails and can easily see why the automatic generation is not possible.

The core of the middle layer ontology HDOT is never modified, so if a match is found in the core then a class can be included if and only if the matched concept is a leaf node in the HDOT hierarchy, i.e. it does

not have sub-concepts. This means that the most specific possible class is integrated. Figure 3. illustrates this idea. The searched term is "*tissue sample*". The concepts highlighted in green "*Specimen*" and "*specimen*" are matched as they have the same label. The problem here is that this concept is contained in the module HDOT_CORE but is not a leaf node in the HDOT hierarchy. Consequently, the OAT cannot generate a recommendation that will be given to the user but a potential recommendation is sent to the curator. The curator can include the subclass "*Biospecimen*" and so there will be a leaf node that matches a parent of a term for integration.

By the search of matching concepts we consider a fixed sorting of the HDOT modules. We first check in the most specific modules and then proceed with more general modules. The motivation behind sorting the HDOT modules in advance is to first try to extend more specific branches existing in the ontology.

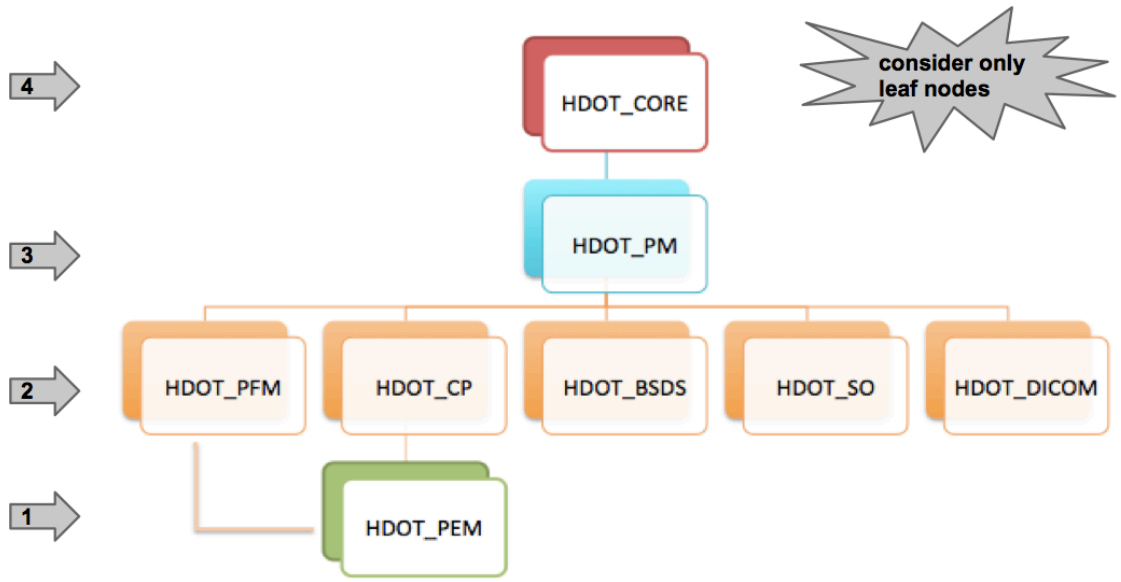

Figure 3: Bottom up search for match in the HDOT module hierarchy

3.4.5 Recommendations Filter

Since we generate not only one but several recommendations, we have to have a mechanism to decide which recommendation to present to the user first. Thus we apply different criteria for comparison to the list of recommendations similarly to the list of semantic resources.

The following criteria are applied by the ranking of the recommendations:

1. the source ontology of the hit is in the pre-defined list of ontologies;
2. the hit provides a definition;
3. the number of the matching parents (higher number preferred);
4. the position of the matched parent in the HDOT hierarchy (higher number preferred);
5. the position of the matched parent in the hit hierarchy (higher number preferred);

The first criterion is the same as the first one by the ontology sorting and is applied here again since for the integration of new terms the compatibility of the searched semantic resources with HDOT is very important and we want to double check this criterion. The second criterion is essential for the user. If a definition is given for a particular term then a recommendation is preferred because the user can better understand the meaning of the proposed class and decide whether it is suitable or not. Criterion No 3 can be seen as similarity measure of the two hierarchies that are compared (the one of the hit and the HDOT hierarchy). In other words, we are not simply checking for a string match of labels and URIs but also consider the fraction of matching classes contained in a hit hierarchy. The larger the number of matching parents the higher a recommendation is ranked. The fourth criterion prefers classes that are deeper in the HDOT hierarchy because the deeper they are the more specific they are and consequently we are able to append the new class further down in a dedicated module. The last criterion for the comparison of the recommendations is the position of the matched parent relative to the searched concept in the hit hierarchy. This criterion prefers lower number and that means that there are not many classes between the matched parent and the class proposed for the integration. In Figure 4 this idea is illustrated. When searching for the term "*brain*" the OAT is able to generate more than one recommendation. The matched parent of the concept "*brain*" is highlighted in green. For the first hit the position of the matched parent relative to the searched concept is three (on the left hand side in the picture), while the position of the matched parent concept for the second hit is one (on the right hand side). Therefore the recommendation for the second hit will be ranked higher by the fifth criterion than the one for the second hit.

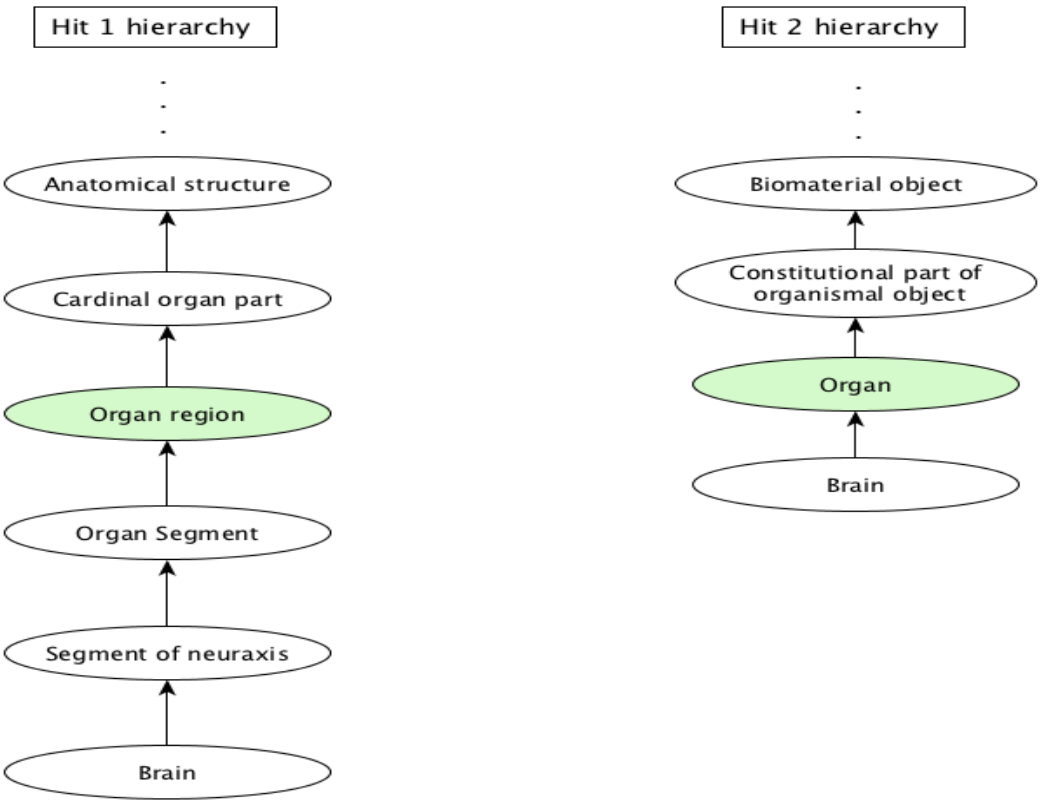

Figure 4: The position of the matched parent concept relative to the searched concept is considered by ranking of the generated recommendations

3.4.6 Super-classes Integrator

By the integration of an accepted term into HDOT the tool automatically includes the super-classes of the hit that are between it and the most specific matched parent. Thus, the HDOT hierarchy is ultimately enriched by exploiting the structure of the source ontology and not just integrating a single term if there are no sub-classes. Moreover, if the super-classes would not be integrated the in a next run a user may

search for one of the super-classes and it would be just added as a concept on the same level but not as parent, which would be rather suboptimal. This idea is illustrated in Figure 5. There the user searched for "*copd*" which is an abbreviation for "*chronic obstructive pulmonary disease*" but might also refer to a protein. OAT generates recommendation for both meanings so if the user has the chance to choose which of them suits her needs. Here we will consider the first meaning to illustrate the idea described above. The hierarchies of HDOT and the source ontology where the term "*chronic obstructive pulmonary disease*" is found are displayed.

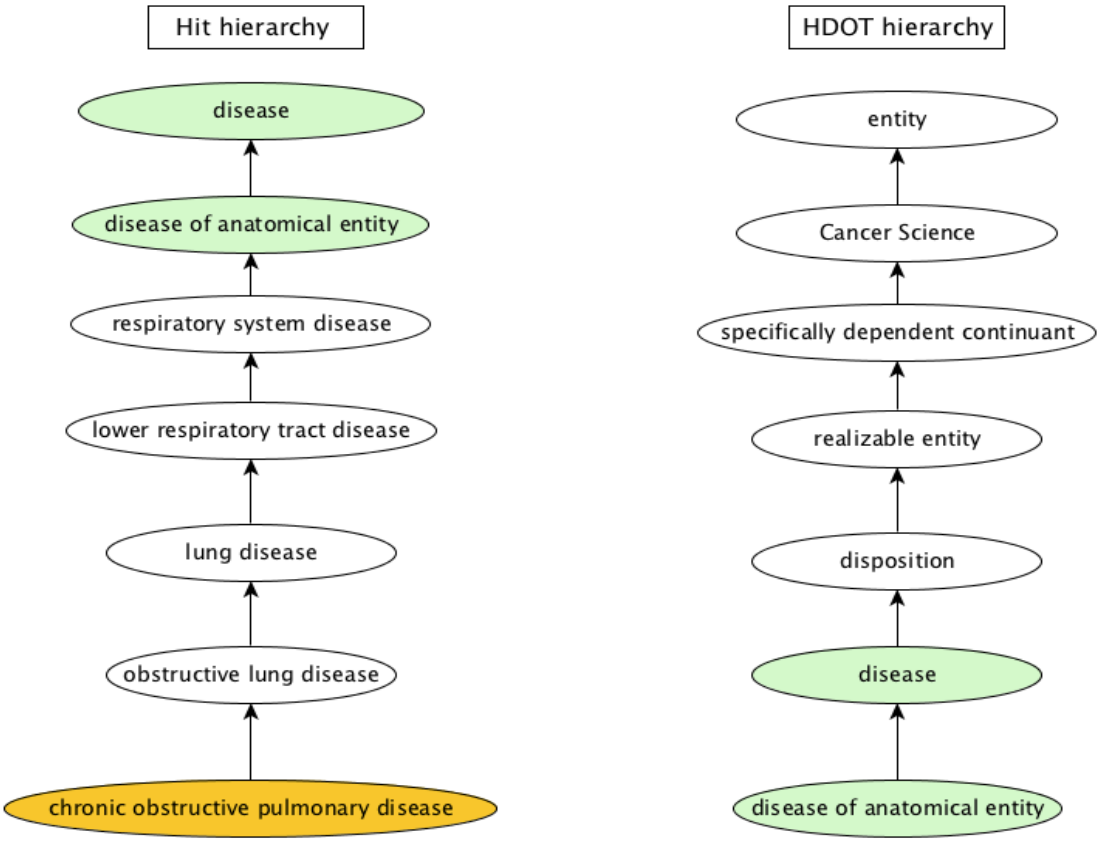

Figure 5: **Searching for term "copd" to the left the hierarchy of the source ontology where a hit is found and to the right the HDOT hierarchy. The matched concepts are highlighted in green and the searched term in orange.**

Two parent concepts matched in the two hierarchies, namely "*disease*" and "*disease of anatomical entity*". In case the user accepts this recommendation not only "*chronic obstructive pulmonary disease*" is included in HDOT but also all classes above it and under the most specific matched parent ("*disease of anatomical entity*") and these are "*respiratory system disease*", "*lower respiratory tract disease*", "*lung disease*" and "*obstructive lung disease*". In this way HDOT is enriched not only with single leafs but with whole branches that are taken from a reliable source semantic resource that fulfills the quality criteria of our semantic framework.

3.4.7. HDOT-module Generator

The OAT generates *user* specific modules. In other words, each user has a dedicated module, where the terms relevant for her specific needs are stored. However, a user is classified as an expert or a non-expert and depending on this status the terms that this specific user integrates are visible for all other users or not. So if a user is an expert then we can trust that the integrated terms are good enough to be used by all other users. If the user is not an expert, she can integrate the term in her dedicated module and perform the annotation with this term but the other user can use the term integrated by a non-expert user after a curation process.

By the integration of the new class in a user module a component of OAT called *URI manager* decides whether to keep the original URI of the external resource or to generate a new one. In favour of the semantic standardisation and reusability it would be better to keep the original URI of the external class. On the other hand, we want to ensure that the quality of the classes offered by HDOT is reliable and thus trust only the ontologies listed in our preferred list to keep the URI. For hits coming from different external resources, a new URI is generated. A new URI is composed by taking the base of the HDOT module where the matched parent was found and appending the counter for terms contained in this module, the counter for this module is then incremented.

After integrating the new class the user module is verified by applying an automatic consistency check. This is a further quality assurance step to the integration of new term(s).

The creation of the new HDOT module and its verification are realized with the OWL API.

3.5 Testing

We simulated user interactions with a list of potential search terms. In this manner, we tested the productivity of the tool. We compiled this list of 36 terms (see Appendix 4) that may be searched and needed for annotation from heterogeneous sources such as research literature, data schemata and case report forms. The complexity of the terms contained in the test list and their level of generality vary. There are simple and frequent terms consisting only of one word like: "*gene*", as well as terms consisting of more than one like "*tissue sample*" or even "*general health rating*". Moreover, we test the behaviour of the OAT on abbreviations such as "*snp*" (*single-nucleotide polymorphism*). The different complexity of the test term gives a notion for the robustness of the OAT.

Table 1. summarises the results of the first run of the OAT on the test terms.

	# of searched terms	**# terms for which OAT generated recommendation(s)**	**# of term already contained in HDOT**	**# terms for which OAT generated potential recommendation(s)**	**productivity**
Cycle 1	36	16	7	10	0.64
Cycle 2	36	26	9	1	0.97

Table 1. Experimental results for the first and second cycle

The total number of the terms we searched amounts to 36. Seven of these terms were detected in HDOT and so there is no need to generate any recommendations. The total number of the terms for which the OAT generated valid recommendation(s) that can be presented to the user is 16. For ten of the terms on the list, OAT creates potential recommendations since some of the extension conditions are violated and the recommendation cannot be directly given to the user but is sent to a curator. This means that only for three terms the OAT could not do anything and this is because no results were retrieved from BioPortal. We report a measure called productivity that is calculated as follows:

$$productivity = \frac{number\ of\ available\ terms}{total\ number\ of\ searched\ terms}$$

Thereby *number of available terms* is defined as the sum of the number of terms with valid recommendations and the number of already contained terms.

The productivity in the fist development cycle is already encouraging namely 64%. The generated potential recommendations are then used for the adaptation and extension of HDOT. After these modifications, which are described in the next section, the productivity measure is improved and reaches even 97%.

3.6 Development Cycles and Refinement

The test described above not only allows evaluation of the OAT effectiveness, but it also provides hints as to the improvement of HDOT, namely through the output of the OAT for those terms, in which the OAT failed to give valid recommendations. Appendix 2 contains a detailed description of the changes made in HDOT that are inspired by the OAT test results of the first cycle.

Remember that in 16 cases recommendations could be generated and in 6 cases the OAT correctly identified the respective query term as already included within HDOT. The latter result is relevant insofar we would like to avoid the reduplication of existing HDOT classes. It is also an important feedback for those tools which have initially evoked the OAT, e.g. the annotator tool, because it indicates that the term is already provided by HDOT. This will be the case even more often as we enrich the annotations of the HDOT classes with synonyms (using the OWL-tag "***hasSynonym***" in the class description), as for example "*neoplasm*" for "*tumour*".

The remaining problematic results can be classified into three categories (see Table 2).

No result from BioPortal	3 terms	general health rating tumour resection tumour grade
No suitable recommendation	1 term	fine needle aspiration biopsy
Recommendation possible, but matched class is not a leaf node	10 terms	pathologist psychological therapy risk factor clinical trial tissue sample

		gene mrt snp mass neoplasm

Table 2. Problematic search terms

We will briefly discuss each case below.

1) No result from BioPortal

No match was found within BioPortal. The only solution to this problem would be to enlarge the base of eligible ontologies, with the risk of importing parts of semantic resources of a lower quality. For example, "*resection*" is actually an HDOT class, such that we could provide a match with "*tumour resection*" in case a lower similarity measure is adopted for potential matches.

2) No suitable recommendation

This case is different from the previous one, since the OAT could find matches within BioPortal, but could not generate any recommendations. A possible reason why for a query such as "*fine needle aspiration biopsy*" no recommendation can be generated, is that "*fine needle aspiration biopsy*" is a complex term, thus that several partial matches are possible, as illustrated below by two possible recommendations generated by the OAT:

```
**************************hit Nr:1**************************
The length of the current path to root is: 7
http://www.w3.org/2002/07/owl#Thing
http://purl.bioontology.org/ontology/SNOMEDCT/71388002  Procedure
http://purl.bioontology.org/ontology/SNOMEDCT/128927009         Procedure by method
http://purl.bioontology.org/ontology/SNOMEDCT/118292001         Removal
http://purl.bioontology.org/ontology/SNOMEDCT/62972009  Extraction
http://purl.bioontology.org/ontology/SNOMEDCT/14766002  Aspiration
http://purl.bioontology.org/ontology/SNOMEDCT/48635004  Fine needle biopsy

************************************************************
**************************hit Nr:2**************************
The length of the current path to root is: 7
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C43431       Activity
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16203       Clinical or Research Activity
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C25218       Intervention or Procedure
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C18020       Diagnostic Procedure
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C15189       Biopsy
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C15190       Needle Biopsy
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C15361       Fine-Needle Aspiration

************************************************************
```

Figure 6. Two possible recommendations for "*fine needle aspiration biopsy*"

If confronted with such a choice, the curators can only make an educated guess and add "fine needle aspiration" for instance. If the user searches again for "fine needle aspiration biopsy" and gets the proposal "fine needle aspiration", she is free to accept or reject it, thus providing a feedback to the curators as to the adequacy of their choice of enriching HDOT.

3) Recommendation possible, but matched class is not a leaf node

The OAT can only propose to attach the imported class to a high-level class such as "specifically dependent continuant", if this is the only match between the hierarchy of the imported class and a branch within HDOT. In order to generate more sensible recommendations, we have to modify HDOT_CORE (and HDOT_PM) themselves, changes which only the curators are authorised to make. These changes are not made in an ad hoc manner if it can be avoided, but take clues from the class hierarchies suggested by the OAT wherever possible. In this way, even though it fails to produce a good recommendation, the OAT can give us hints to improve the domain coverage of HDOT und thus the quality of future hits.

We will briefly discuss the different scenarios below; the reader should refer to the Appendix 2 for further details on how we modified HDOT on the basis of the evidence provided by the OAT.

The first option, which can be applied to 5 terms (*pathologist, gene, mrt, clincial study* and *snp*), is to extend the HDOT_CORE hierarchy by adding the intermediate classes suggested by the OAT between the high-level HDOT_CORE class and the query class. In order to ensure exhaustiveness, the sister nodes of the subsuming terms should be added too. For example, as the first hit for "*pathologist*", the OAT suggests the following hierarchy from the Ontology of Biomedical Investigations (OBI):

http://purl.obolibrary.org/obo/BFO_0000001	entity
http://purl.obolibrary.org/obo/BFO_0000002	continuant
http://purl.obolibrary.org/obo/BFO_0000005	dependent continuant
http://purl.obolibrary.org/obo/BFO_0000020	specifically dependent continuant
http://purl.obolibrary.org/obo/BFO_0000017	realizable entity
http://purl.obolibrary.org/obo/BFO_0000023	role
http://purl.obolibrary.org/obo/OBI_0000202	investigation agent role
http://purl.obolibrary.org/obo/OBI_0000116	worker role
http://purl.obolibrary.org/obo/OBI_0000145	pathologist role

The HDOT hierarchy that matches this OBI branch is the following:

http://purl.obolibrary.org/obo/BFO_0000001	entity
http://purl.obolibrary.org/obo/BFO_0000002	continuant
http://purl.obolibrary.org/obo/BFO_0000005	dependent continuant
http://purl.obolibrary.org/obo/BFO_0000020	specifically dependent continuant
http://purl.obolibrary.org/obo/BFO_0000017	realizable entity
http://purl.obolibrary.org/obo/BFO_0000023	role

In order to ensure a better match in the future, we add the OBI classes investigation agent role and worker role to HDOT, as well as their sister classes of worker role in order to satisfy the exhaustiveness requirement for HDOT.

This is also perhaps the best place to mention the problem of acronyms such as "*MRT*" or "*COPD*". These acronyms are mostly ambiguous, as they may stand for distinct concepts. "*MRT*" may stand for "*magnetic resonance tomography*" or for a specific gene, while "*COPD*" may stand for "*chronic obstructive pulmonary disease*" or for a certain protein. The curators have no insight into the meaning of the original query and so have to make educated guesses, the best being either the most generic or the most frequent meaning. Once the user is proposed this new term, she may either adopt or reject the choice, which will trigger a notification to the curators who can amend HDOT accordingly, proposing other concepts or classes to the user, whose response can again be either positive or negative, and so on. As for other ambiguous terms, this cycle of "update-feedback-update" will hopefully eventually lead to a successful disambiguation.

The second option to generate suitable recommendation in cases there are yet none, is to add alternative labels to existing HDOT_CORE classes that are synonymous either with the query terms or with classes that immediately subsume the query terms. This solution could be applied to three terms (psychological therapy, tissue sample, neoplasm). In the case of psychological therapy, the matching subsuming class is labelled "*therapeutic act*", which is synonymous with "*therapy*" and the HDOT_CORE class "*treatment*" (to which we consequently add the labels "*therapy*" and "*therapeutic act*"). Similarly, tissue sample is subsumed in the matching hierarchy by a class labelled "*biospecimen*", which can be added as a label to the HDOT_CORE class "*sample from organism*". Finally, since there is already a class labelled "*tumour*" in HDOT, we can easily add "*neoplasm*" as an alternative label for that class. So the OAT may be able to suggest us new labels for existing HDOT classes, and thus also give us a measure of the quality of the middle-layer built so far.

The third option is the ad-hoc addition of the query class, where no (or at least no suitable) intermediary class hierarchies are suggested by the OAT. This may be necessary if the suggested class is a high-level class itself and can thus be sensibly added below a BFO term, as in the case of "*risk factor*" which we have to add under "*role*". An ad hoc addition is also necessary when the available resources within which matches could be found are of low quality and do not fully satisfy OBO design criteria. This is the case for "*mass*" in the sense of "*lump*". The matching class hierarchy from *CRP* could not retained as it mixes material entities such as lesions with information artifacts such as clinical findings. Since we have no way to figure out what was the exact meaning of the search term the user had in mind, we have to make educated guesses about the optimal interpretation of the term. In the case of "*lump*", we choose the most generic meaning, namely "*mass of tissue*", which would be an object aggregate in HDOT; as such "*lump*" is added to HDOT_PM, not to the HDOT Core itself in case we have to revise our change. In fact, feedback from future uses of the OAT may give us clues as to whether or not our guess was correct or not.

3.7 Updates

Updating the information encoded in a system is one of the main issues faced when dealing with knowledge management, especially for tools applied in the fast growing and ever changing field of life sciences. Thus we need a strategy that will allow us to keep the terms in HDOT up-to-date.

As already mentioned BioPortal has a large inventory of semantic resources and it is steadily developing. Since the new terms in HDOT that are integrated by a user are found in a semantic resource contained in BioPortal we ideally should ensure that the terms are (1) still available (2) have the same meaning and (3) are in the top 10 of the ranked hits list.

There are three ways to approach an update strategy. In order to automatically detect changes in a previously used semantic resource and to also consider new ontologies included to BioPortal, one can:

- parse the update messages that are generated by BioPortal and if any of the resources used for extending HDOT is mentioned then to check if the term that has been integrated is also affected or not.

- generate a logging with all searched terms within the last six months, redo the search and check for changes in the generated recommendations.

- detect changes in the number and position of the ontologies by the ontologies sorting

If we go for the first option, we will depend on the completeness of the generated update messages of BioPortal. As a matter of fact on BioPortal's homepage currently the number of ontologies is 373 but the web services retrieves 537. So the information seems not to be synchronized. Additionally, all messages should be parsed although they may not concern resources that have been used for the extension of HDOT.

The second option seems to be more appropriate for the updates we want to detect. The idea is to regenerate the recommendations that the user has seen by the time of integrating a term and check the following:

- is the number of generated recommendations the same

- is the content the same and in particular, is the integrated term available

To be able to do this, we need to know the state of HDOT at the time when the user performed the search.

The main issue here is how to keep track of all states of the HDOT ontology for all searched terms within the last six months. Since various searches and several terms can be integrated under HDOT for the period of six months, designing an effective version control of the ontology among with the terms searched with each version opens a new research area.

Therefore we believe that the third option (detecting changes by the ontologies sorting) is most appropriate and will serve as a good indication of changes. On the one hand side, we check which ontologies have been removed. On the other hand side, we check which ontologies are added in the last six months.

For the list of removed ontologies the update service examines if any of the preferred ontologies is contained. In this case a strong warning is produced. Terms originating from the preferred semantic resources are integrated with their original URI and so the integrated term is no more available. If the list of removed ontologies does not contain an ontology on our preferred list, even if a removed ontology has been used as a source for generating a recommendation, a new URI has been generated by the integration of a term.

We collect the information which ontologies are used for extending HDOT and in particular, which terms of these ontologies are reused.

If the position of any used ontology shifts in more than 5 positions down in the sorted list, the curators are informed and can decide whether the ontology needs to be updated or if the changes are rather minor and irrelevant to the use of HDOT. They can investigate the generated recommendations for the terms taken in out of the shifted ontology and integrated in HDOT and check if the recommendation of this resource is still in the top 10 recommendations.

We suspect that structural changes almost will not be found on the term level, i.e. the position of a term in the hierarchy is different in the same resource. Rather *simple* changes in the surface representation of a term like different label or including a definition will occur.

3.8 Interaction with other tools

We see possible interaction of the OAT with all other tools that exploit HDOT. For there is always the possibility that a semantic representation a user needs is not included in HDOT or one of its modules yet. In this case the tool can invoke the OAT to allow the user to find a solution for the missing concept. We have designed the OAT in such a way that this solution becomes available to the user straight away (provided she has the appropriate rights) so that she can continue working with the other tool seamlessly. Such tools are for instance the Ontology Annotator Tool, the case report form builder in ObTIMA or the Biobank access tool.

Apart from this the OAT can be invoked directly from p-medicine's portal (to be fully implemented in month M38) in case a user would like to access it outside the context of using another tool for instance because she is interested in ontology development and wants to extend and refine HDOT for its sake and independent of any other task.

3.9 The Ontology Aggregator user interaction

3.9.1 Stand-alone web interface

The Ontology Aggregator functionality is accessible through a dedicated web interface that allows end users to access and execute the aggregator workflow described previously. This interface (OAT Interface from now on) provides a simple access point for users lacking technical background who wish to extend HDOT with additional resources from BioPortal.

The OAT Interface is built on a series of technologies, which include Java, Javascript, JSP and CSS. The focus has been put on flexibility upon future changes in the way the information is presented to the users. The resulting OAT Interface application architecture follows the data-driven philosophy, by moving part of its logic to external files, outside of the programming and scripting code. In this sense, the use of CSS allows defining the *look* of the interface in external files, facilitating its edition and replacement upon new requests. Furthermore, the content of the information presented to the user, i.e. text messages, recommendation layouts, is also defined in external *properties* files which can be edited to fit different needs. Hence, the OAT interface can be deployed using different configurations in order to adapt to different languages or different types of users, e.g. expert users and novice users. The OAT Interface architecture is depicted in Figure 7.

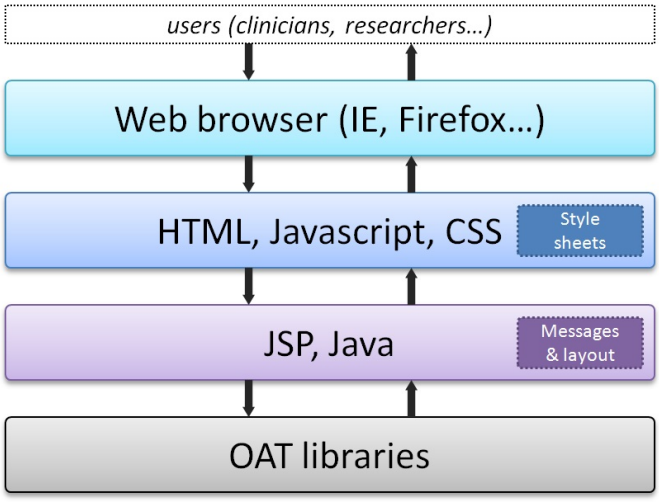

Figure 7. Architecture of the OAT Interface (blue and purple layers), including the programmatic libraries for extending HDOT (grey layer)

The last layer of the architecture, depicted as "OAT libraries", provides the access to the search engine on the BioPortal repository and the HDOT modification functionalities. These methods are sequentially accessed by the Java code contained in the third layer in order to compile the workflow of events necessary to realize the search of information and extension of HDOT. In addition, the Java code includes the logic to maintain the sessions that identify the users that are connected to the application at any given time. It provides the necessary concurrency features for enabling the asynchronous access to the Ontology Aggregator.

The OAT Interface has two main screens: the initial search screen and the recommendation presentation screen. At the beginning, the user is issued to input a search term in the initial search screen. This screen is shown in Figure 8.

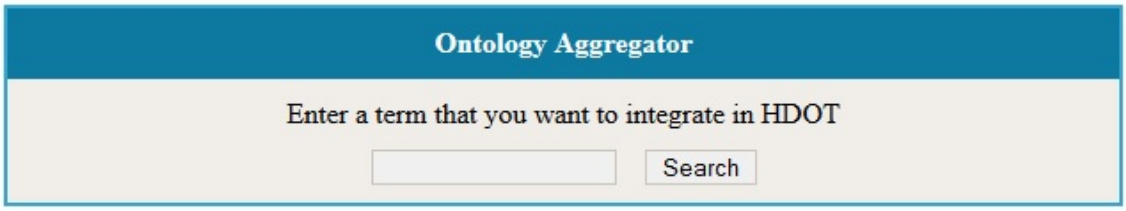

Figure 8. The initial search screen

When the user types a term in the text box and clicks on the "Search" button, the OAT Interface initiates a search process by accessing the OAT libraries. A search might take about a minute, during which the user must wait until a result is presented. When the OAT libraries have compiled enough results, they are presented to the user as a series of recommendations, one by one. Figure 9 shows the recommendation presentation screen, which presents all the information related to one of the recommendations generated after a search process.

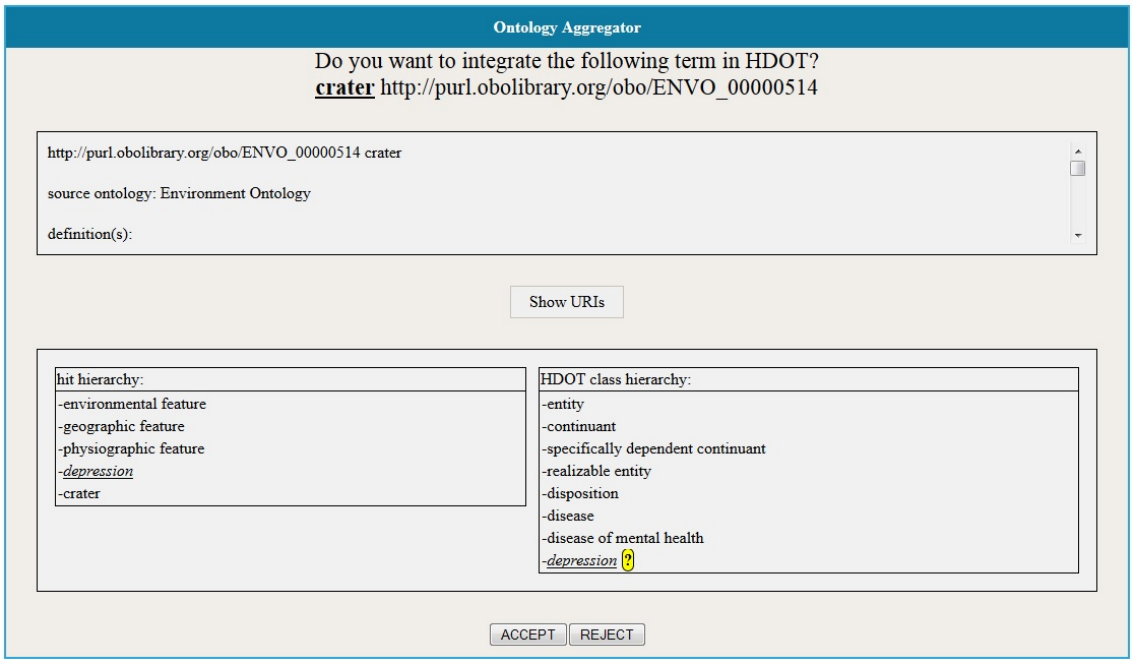

Figure 9. The recommendation presentation screen, including information about the proposed class to include and HDOT, and the location in HDOT where this class would be inserted

This screen presents a recommendation, which is composed of a class found in one of the ontologies of Bioportal, the information related to such class –full URI, label, source ontology, definition, synonyms and

subclasses–, the class hierarchy of the found class, and the HDOT class under which the found class would be integrated, if the user agrees with it.

For a full reference of the features contained in the OAT Interface and a detailed description of its components, please refer to Appendix 2.

Types of users

The OAT Interface distinguishes two types of users: normal users and users with extended rights. The two types of users can perform the same actions in the application. However, new classes added by the former always require the supervision of an HDOT curator, while the latter users are able to circumvent this supervision.

The fact that a user acts as a normal user or as a user with extended rights is actually transparent to the user herself. Normal users with immediately see the changes performed to HDOT by extending it with the OAT, and they will be able to access the new classes in the Ontology Annotator tool. However, internally, the OAT will hide those changes from the rest of users until they pass the required validation. Changes made by users with extended rights, on the contrary, will be immediately visible by the rest of users.

In the current version, users themselves must specify with they have extended rights or not at the time of login in. In next versions, the extended rights will be a property of the p-medicine credentials associated to the user, so it will not have to be specified at login time.

3.9.2 Upcoming features

The current state of the OA Interface must still undergo several enhancements in order to comply with the requirements defined in the p-medicine project. These are listed below:

- Deployment inside p-medicine infrastructure: The OAT Interface is an independent application deployed at a server hosted by UPM. Upcoming development will focus on enabling access to the OAT Interface through the p-medicine portal, thus fully integrating the application in the p-medicine infrastructure. The OAT Interface will be accessible as an independent tool from the available tools in the portal, and also by selecting it in the Ontology Annotator.
- Integration with p-medicine security infrastructure. Currently, the OAT Interface does not check the identity of users. This will be modified so that only registered users with proper permissions will be allowed to use the OAT Interface and extend HDOT. The credential schema in p-medicine must still define specific properties that define the rights of each user to extend HDOT. The OAT Interface will read this property and act accordingly.
- Full communication with the Ontology Annotator. The Ontology Annotator is not yet aware of the extensions of HDOT that the Ontology Aggregator produces. Future developments will enable this by communicating the Ontology Annotator with the OAT Interface. The latter will provide services for dynamically querying the state of the extensions of HDOT, both general extensions and per user extensions. This communication will by totally hidden from users, providing on-the-fly updates of the new modules generated by the Ontology Aggregator.

The expected deadline for these updates is month M38.

3.9.3 User interaction and useful tips

This section provides a kind of detailed reference manual for the OAT Interface. It is aimed at end users that wish to make use of this application in order to extend HDOT upon their needs.

Accessing the OAT

The OAT can be accessed at http://ifomis.org/oat. The OAT can be executed in any modern web browser and in any computer with Internet capability.

NOTE: the final version of the OAT will be accessible through the *tool* menu in the p-medicine portal. The current URL will then no longer be available

Login screen

The first screen that is shown after accessing the previous URL is the login screen. Here, the user must provide his login details. Figure 10 shows the OAT login screen.

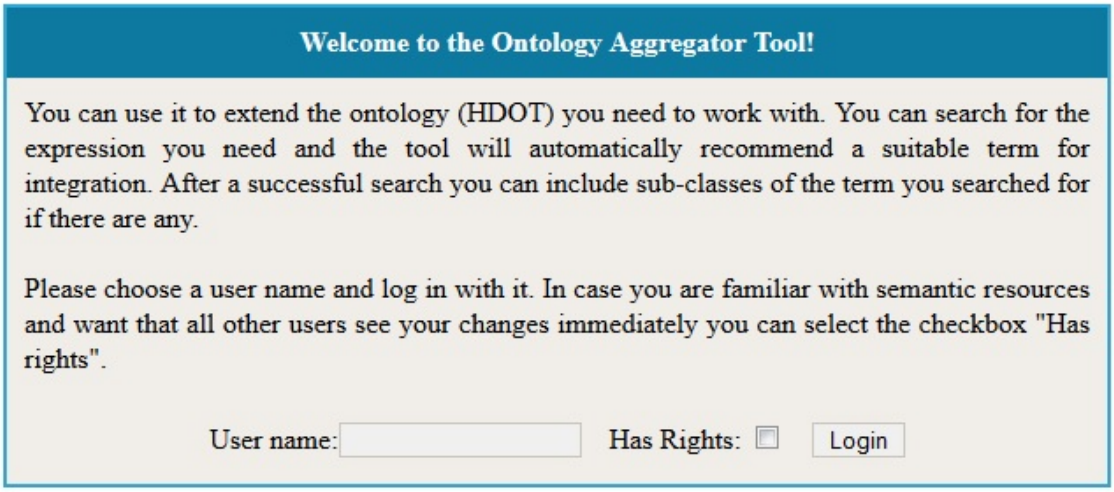

Figure 10. The OAT login screen. Users must provide a user name and specify whether they have rights to extend HDOT without curator supervision

The login screen provides a brief description of the OAT, allowing new users to get a glance of the purpose of the application. The login information that must be provided is a user name and whether or not this is a trusted user that has rights for autonomously extending HDOT with new classes or, on the contrary, any actions she performs require the supervision of an HDOT administrator. The user name information will allow the OAT to store any changes performed by the user in an appropriate location identified by that user name, so that those changes are available to the same user in subsequent sessions.

NOTE: the final version of the OAT will not include the login screen, since the application will take the login information from the p-medicine credential system

Search screen

After the user logs in the application, she is presented with the search screen, as shown in Figure 11.

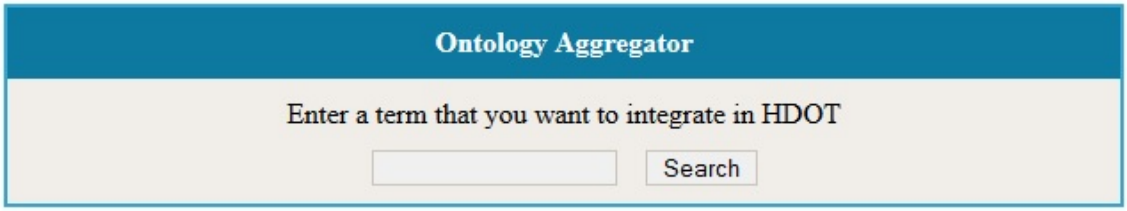

Figure 11. The OAT search screen. Users can type any term they wish to include in HDOT here

This screen allows users to search for terms which they wish to include in HDOT. The user can type the term of search in the available text box. Examples of searches are "brain", "biomarker", or "clinical trial". Clicking on the *Search* button will initiate the search of the provided term. This process can take a few minutes to complete, until the search results are presented to the user.

Search results

The search issued by the user can end in one of seven possible results, which are listed below:

1. ***No results found***: the OAT did not find any match for the given search term. The user will be issued to rephrase his search, or perform a new search.
2. ***Searched term is already in HDOT***: the OAT found the searched term in HDOT, meaning that the user does not need to use the OAT in this specific case. Detailed information about the existing term in HDOT, including its full hierarchy, is provided to the user.
3. ***Error***: there was an internal error which prevented the OAT to perform the required search. The user will be issued to try again after a few minutes.
4. ***Searched term will soon be included in HDOT***: the term searched by the user was found by the OAT, but it cannot be included in HDOT immediately, as it requires a deeper analysis by an authorized curator. The HDOT administrators are automatically notified of this situation to proceed with the manual inclusion of the term.
5. ***Searched term cannot be included in HDOT***: the searched term was found by the OAT, but the system was unable to find a suitable path in HDOT where the term should be inserted. The HDOT administrators are automatically notified of this situation so they can fix it.
6. ***Several results that match the search term were found***: the OAT found several possibilities that match the search provided by the user. The user will have to decide which the correct one is. This case is described in more detail in the *Recommendation screen* subsection.
7. ***HDOT was extended***: one match was found for the given search term and it was automatically included in HDOT. The system presents details about the included term (label and URI).

Cases 1, 2 and 3 end up with no changes in HDOT and automatically take the user back to the search screen. Cases 4 and 5 will eventually produce changes in HDOT, but after the administrators have checked the details of the situation. Cases 6 and 7 will produce immediate changes in HDOT (case 6 will require confirmation by the user). Figure 12. shows a diagram summarizing these situations.

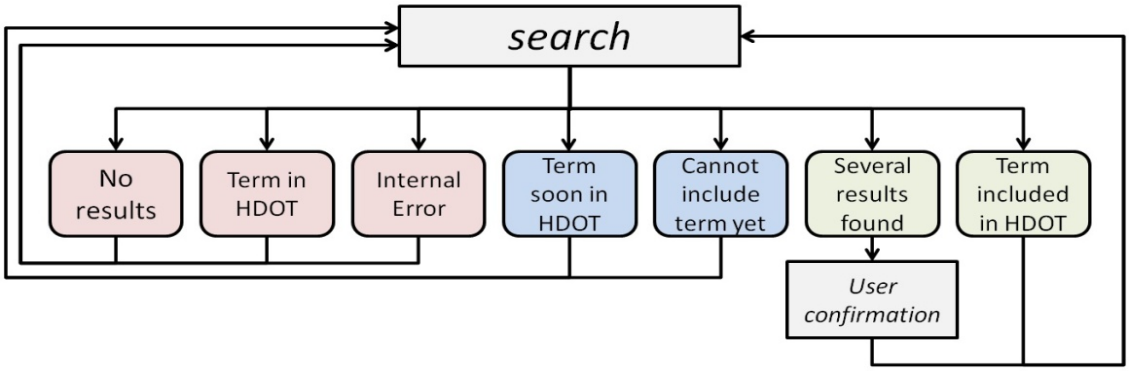

Figure 12. Diagram representing the different situations that the user can reach when searching for a term in the OAT

The screen corresponding to each of the possible situations provides intuitive messages that explain the situation to the user. In all cases, the screens include in their top left corners two buttons labeled as "Logout" and "New Search". Figure 13. shows these buttons.

Figure 13. Every OAT screen includes these two buttons, which allow the user to either log out of the application or perform a new search

These buttons allow the user to go back to the login screen and to the search screen, respectively.

NOTE: the final version of the OAT will remove the logout button in favor or the p-medicine portal logout control

Recommendation screen

This screen represents a special case in the OAT workflow, as it requires further interaction from the user upon selecting the most appropriate result from a series of matched terms from BioPortal. In this case, the user is sequentially presented with the information of the best recommendation, which she can either accept or discard. If the recommendation is accepted, then the term in it is included in HDOT. However, when a recommendation is discarded, the next one is presented to the user until the system runs out of recommendations for the current search.

One single recommendation contains several elements of information. Figure 14. depicts one recommendation obtained for the search "influenza".

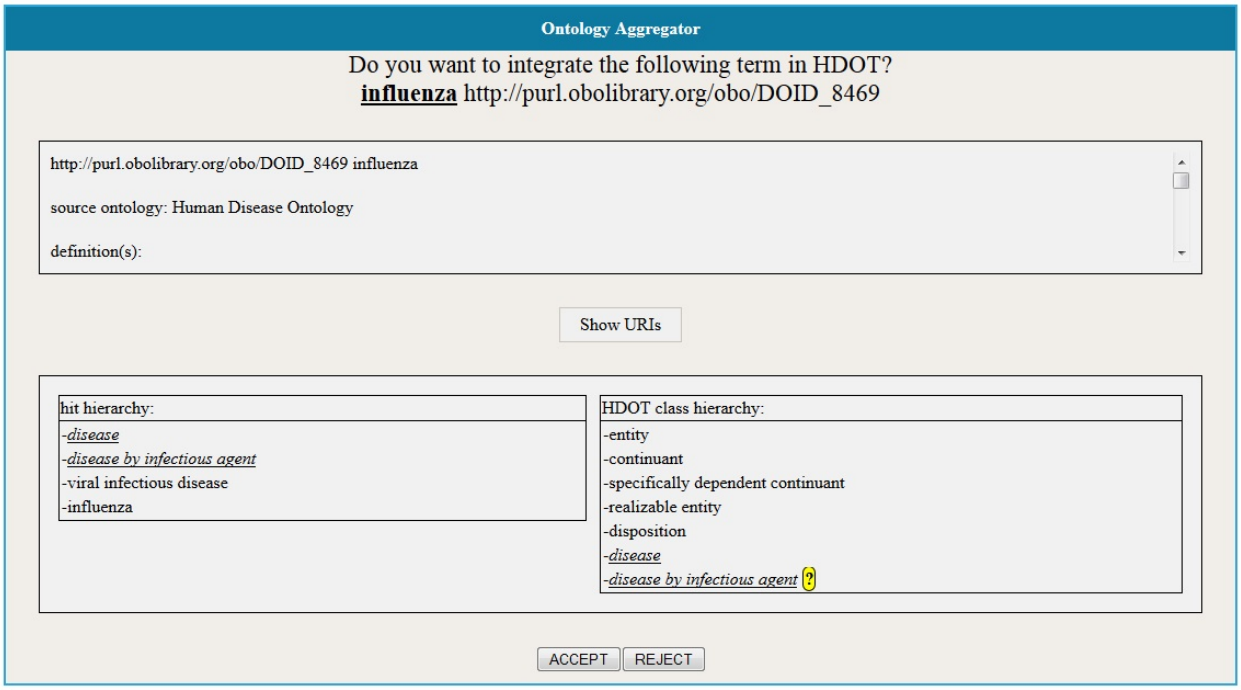

Figure 14. . Recommendation generated for the search of "influenza"

A recommendation is formed by one term from an ontology in BioPortal, called "hit", and an existing term in HDOT that, in case of inclusion, will become the parent of the hit. The hit label and URI are displayed on top of the screen. Below that, a scrollable window provides additional information about the hit. This includes:

- Hit label and URI: the hit label and URI are repeated at the top of the scrollable window
- Source ontology: the ontology from BioPortal where the hit has been found
- Definition(s): the available definitions for the hit. The user should put especial care by checking that this definition suits his original search.
- Synonyms: the available synonyms for the found term.
- Subclasses: the terms that inherit from the given hit.

Figure 15 provides detail for the scrollable window after searching for "influenza".

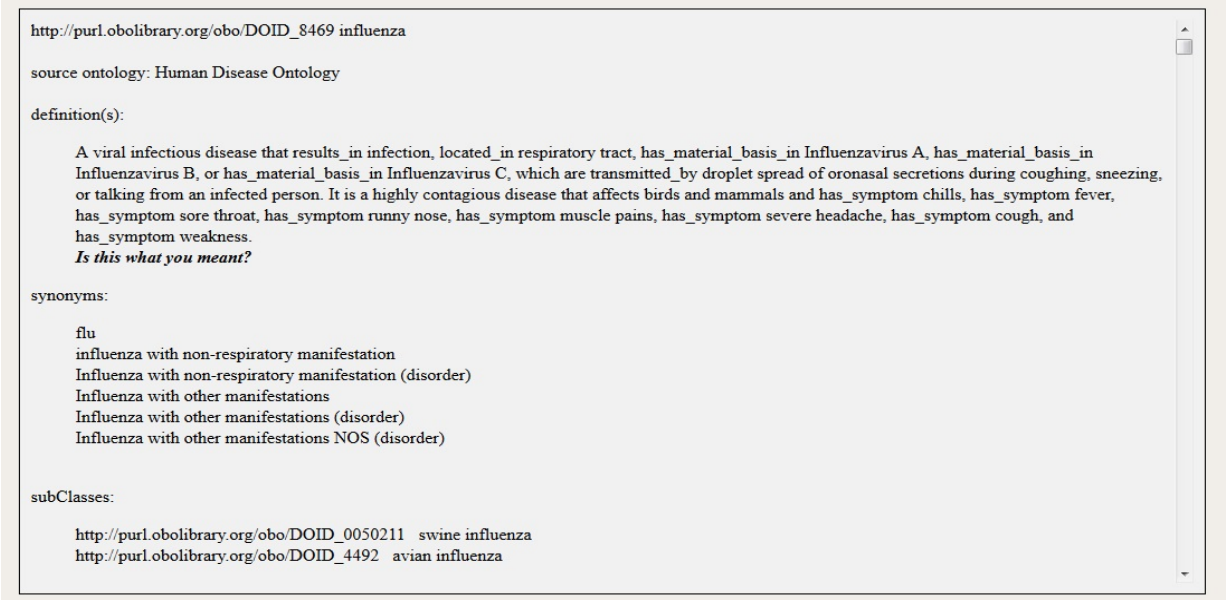

Figure 15. Example of information provided in the scrollable window inside the recommendation screen.

Below the detailed information for the recommendation hit, the recommendation screen displays both the class hierarchy for the hit and the class hierarchy for the HDOT class under which the hit could be integrated. Figure 16. focuses on this section of the recommendation screen for the search of the term "microarray".

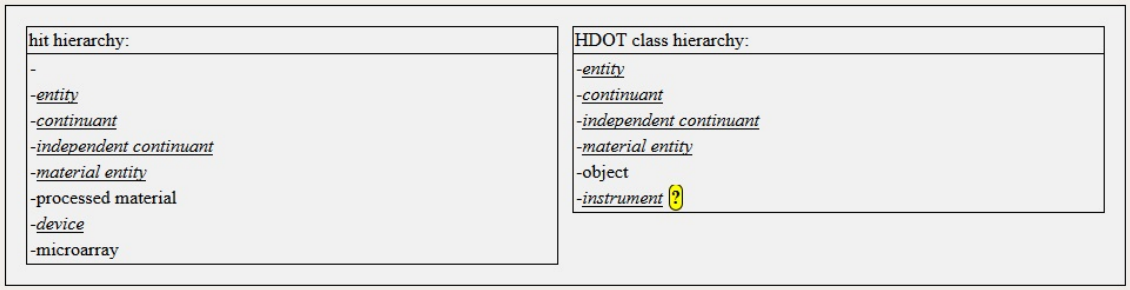

Figure 16. The class hierarchies for the hit in BioPortal and for the HDOT class under which such hit will be inserted. The hit (microarray) appears at the bottom of the hit hierarchy. The HDOT class (instrument) appears at the bottom of the HDOT hierarchy

When a pair of terms from both hierarchies match –i.e. have the same meaning–, they are underlined to help the user identifying these coincidences. In the previous example, the OAT identifies five pairs of coincident terms: i) "entity" with " entity ", ii) "continuant" with " continuant ", iii) "independent continuant" with "independent continuant", iv) "material entity" with "material entity", and v) "device" with "instrument".

When the user accepts the proposed recommendation, a pop-up window is displayed showing again the hit label and URI, together with the hit subclasses. The user is then enquired whether the hit subclasses should also be added to HDOT. Figure 17. depicts this situation.

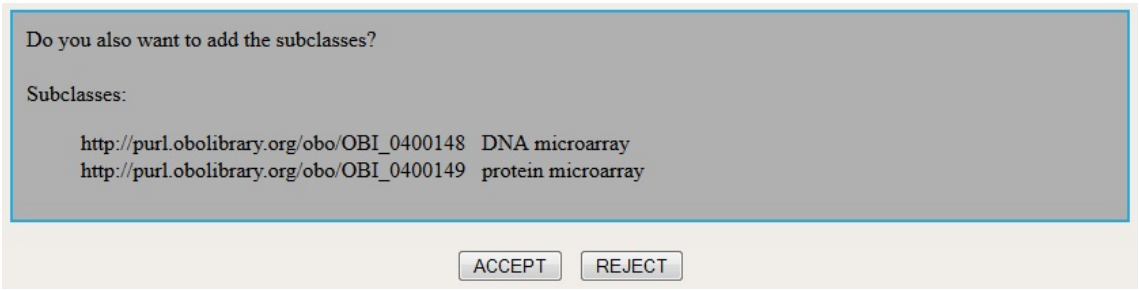

Figure 17. If an accepted recommendation hit has subclasses, the used must confirm whether the subclasses should be included as well

Useful tips

Invalid recommendations

The user should always check the consistency of a recommendation. There are cases in which the OAT will make meaningless suggestions, which must be detected by the user. It is important to read the hit definition and compare it with the definition of the HDOT class under which the hit will be integrated –this can be done by hovering the mouse over the '?' symbol next to this HDOT class, in the HDOT hierarchy. For example, if a user performs the search for the term "crater", the recommendation shown in Figure 18. will be displayed.

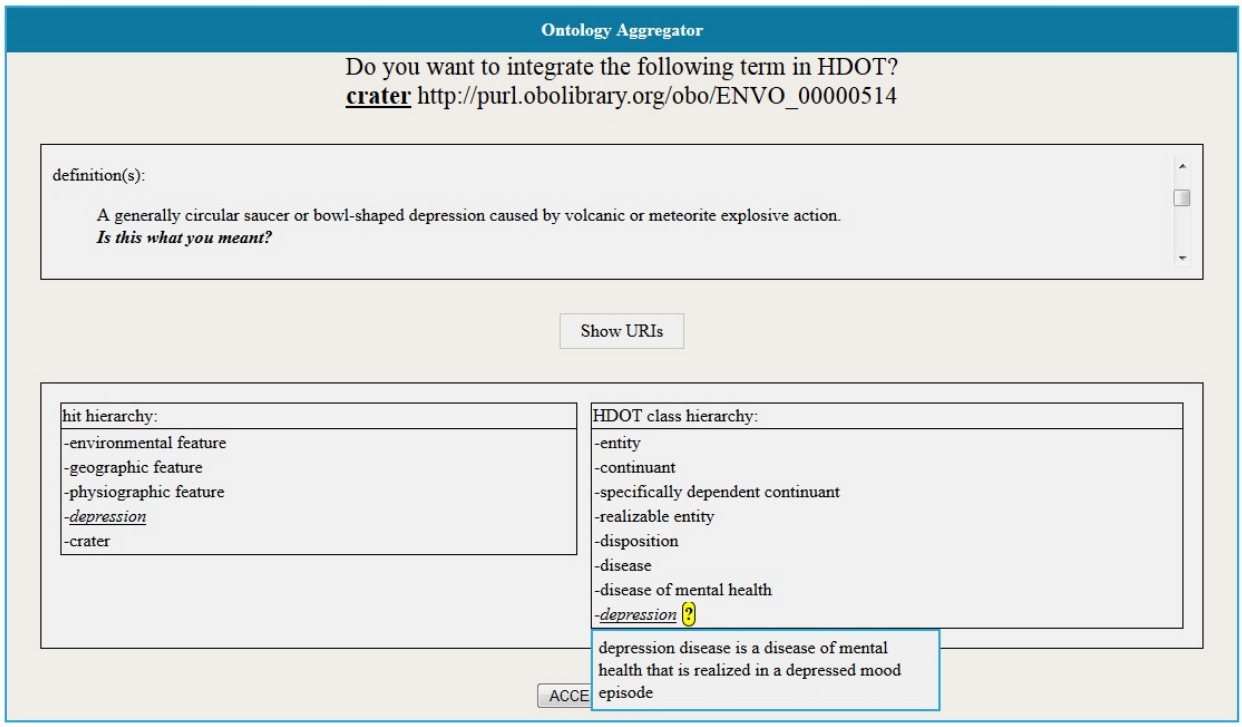

Figure 18. Searching for the term "crater" produces an invalid recommendation

However, by taking a look at the corresponding definitions, it is easy to see that the hit refers to a geological depression, which the HDOT class refers to the mood disorder. The user should therefore reject this recommendation.

Visualising the hierarchies classes URIs

The recommendation screen includes a button labelled "Show URIs". By clicking it, the screen includes the URIs of the terms in both hierarchies. Figure 19. shows a recommendation screen with the URIs activated.

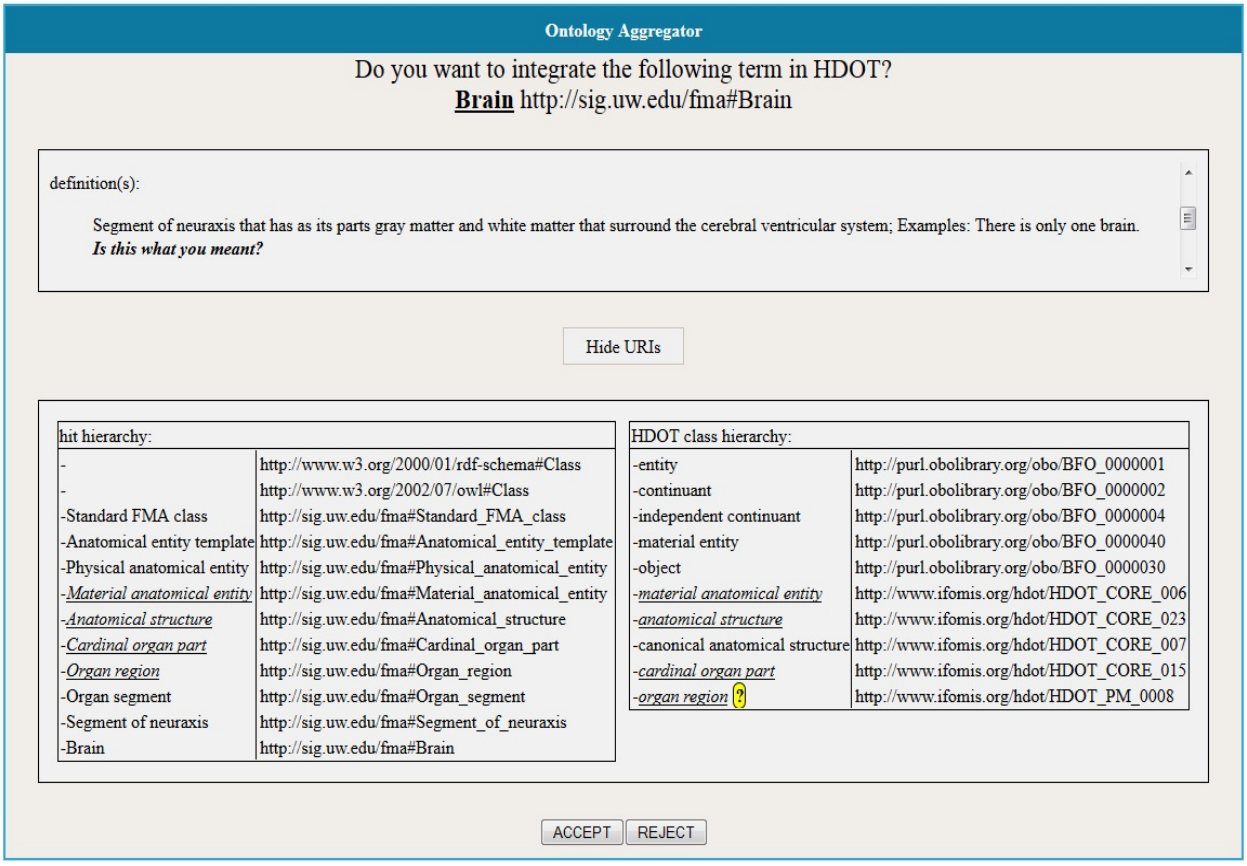

Figure 19. A recommendation displaying the URIs of both class hierarchies for the search of the term "brain"

This information might be useful for expert users, capable of identifying a class by its URI. For the rest of users, it is recommended to ignore the URIs.

Selecting the best recommendation

When presented with a list of recommendations in the recommendation screen, the system will try to put first the most adequate according to its own internal criteria. Nevertheless, the first one might not always be the one that the user expects, forcing him to discard recommendations until she finds a suitable one. Always consider carefully whether accepting or rejecting a recommendation, since the system does not allow undoing a step, and rejected recommendations can only be recovered by performing a similar search again.

4. Achievements and Challenges

From our point of view we succeeded in building a flexible and powerful tool for extending HDOT with pre-existing parts of other SRs. The design of the OAT is split into separate components (see chapter 3 for detailed descriptions) each of which can be adjusted and improved independently so that future developments and new requirements can be accommodated without having to re-design and change the whole system.

If for instance it becomes desirable to generate different or more recommendations to users we can easily adjust the list of preferred ontologies or play with the similarity measure for finding hits in external SRs. Additionally, the criteria for ranking hits could be weighed differently or new criteria added in such a way that only small changes in a specific component are necessary to tune the effectiveness of the tool in this respect.

This feature of its design is highly relevant for future work as the OAT generates a lot of valuable information especially in those cases in which a problem occurs and an extension of HDOT is not yet possible. The most interesting cases for future improvements are those in which the tool is able to find a hit for a candidate-class from an external SR, but is not able to subsume this class in HDOT because there are no appropriate matches of parent classes. In this case the tool sends a message to the curators describing the search, the parent-classes and the reason why possible matches are not available so that the curators are provided straight away with all the information they need to add to HDOT missing higher-level concepts so that better matches can be realized. We thus learn where problems occur with the tool and in most cases it is able to propose possible solutions automatically. We described how this works in detail in the section development cycles and show in the appendix how HDOT can be refined with this method. We are also looking into the possibility of processing this information fully automatically so that the interference of curators could be kept to a minimum in these cases.

Another aspect in which we could decrease the need for human intervention would be the exploitation of definitions available for concepts and classes. We do already retrieve this information from external SRs where available and provide it anyway in HDOT. By parsing the definitions we could add an additional criterion for assessing appropriate subsumption relations inside HDOT. This could be desirable particularly if we opt to reduce the necessary interaction with users even further.

However, we do not consider this option as one that should be pushed too far in the current context and state of p-medicine's data description requirements. Before we can think about reducing human interaction in data description scenarios we first have to make absolutely sure that the terms, concept and classes added to HDOT are represented in the sense and meaning the user working with these has in mind. For judging this matter we rely on the user to carefully check the subsumptions and definitions the OAT offers and displays to her before HDOT is extended with additional terms and classes. Only if a human user accepts the recommendation generated by the OAT a new class can be added because, ultimately, if with all the computations done by the OAT in the background, it is only through the user's acceptance that we can be sure to avoid false representations of intended meanings for terms or other conceptual mistakes. This is particularly true for ambiguous terms for which the tool can only offer necessary information to decide in which meaning the term should be represented, but it is the user who has to make that choice between multiple meanings as the machine simply cannot know what the user has in mind.

We are convinced that we have managed to display all relevant information for users to make a well-informed decision about the integration of terms, concepts and classes from other SRs into HDOT and we are pleased to find that the complexity and amount of that information seems to us to be such that we can expect users to comprehend and use it quite easily.

It should be noted that we opted to reduce the necessary interaction with the user to the ultimate choice between accepting or rejecting a recommendation, the input of a search term and the selection of possible subclasses for integration. We designed the OAT as a quick problem-solving service and so we

do not display any information we consider not necessary for this purpose, i.e. we try to reduce users' cognitive load as much as possible. We thus imagine that users only invoke the OAT to solve problems they face elsewhere and we would like to let them get on with this as fast and effortlessly as possible.

One major technical challenge is to reduce the processing time of the tool. There are several ways in which this could be realised. We could for instance reduce the number of BioPortal hits we subject to a comprehensive matching process or we could restrict the number of SRs we make available to users. We have already tested these options and the disadvantages of these solution seemed to us not be compensated by the achieved improvements in processing time. This is because we found that by far the most relevant factor for the processing speed is accessing the REST service of BioPortal. At the moment this lies outside our influence and can only be changed if we were able to install a virtual machine for BioPortals triple store locally. We have already explored this solution and prepared space on a service at PSNC for this, but unfortunate we incurred technical problems with the virtual machine provided by BioPortal for this purpose. We raised these problems with its developers, but are still waiting for a solution.

Another challenge we consider important is the handling of complex or composite terms and expression. Currently we are not able yet to decompose those into their building blocks, process these separately and allow for relating them again after analysis and form the desired composite expression. However, we are confident that we can use ontological relation between classes and concept to model at least some of the semantic relations between parts of composite terms and thus represent those by representing their parts and relating them in an appropriate way in HDOT. This is by no means trivial and will require a lot of more research, but the graspable benefits if this approach seem evident to us.

We see also acknowledge the need to find better solutions for those terms and expressions which are not included in BioPortal (or SRs in general). Naturally, reusing pre-existing standardised concepts in these cases cannot be a solution and up to now the OAT cannot offer the user any help in these cases. However, we are convinced that we could exploit the information contained in HDOT and the possibilities of extending it with the OAT in these cases to design a short "ontological dialog" with the user by through which we could be able to at least offer some well justified ontological representation of the term the user needs represented inside HDOT.

5. Conclusion

New IT strategies are currently explored to enhance the communication between patients and health care providers. Ontologies play a central role in this framework, because they help in many ways, such as making people's assumptions explicit and representing the complexity of the biomedical domain in a computable language, thus making data sharing more efficient. In this paper, we present the Health Data Ontology Trunk (HDOT). We describe crucial methods in its development in particular its modularity, the benefits of a middle-layer ontological approach and our method of re-using parts of pre-existing resources implemented in the OAT. Indeed, a domain middle-layer ontology like HDOT can be further extended and specialized in different but interrelated modules according to more specific biomedical and clinical needs and requirements while always maintaining one and the same axiomatic framework. One major advantage of this approach is that we can specify necessary top-class axioms in the middle-layer that are automatically inherited by appended subclasses. In this way we provide a minimal set of semantic constraints on the use of biomedical and clinical terms and expressions and are at the same time able to represent possible interrelations by sorting them into one coherent axiomatic framework. Systematic reuse of existing ontologies is a basic principle in the design not only of the modules that extend HDOT,

but also of HDOT itself, which sets our approach apart from other strategies in the development of middle-layer ontologies for the biomedical domain.

We are pleased with productivity and effectiveness of the OAT in first tests and we find the data generated by the tool in problematic cases very helpful so that we are able to define a problem solving workflow for those quite easily. It should be noted that the tool collects a lot more information than we choose to make available to the user. This allows the developers to get a profound insight into semantic standardisation problems within p-medicine's semantic framework. On the other hand we restricted the displayed information to a minimum to allow users to make informed decision without overburdening them with ontological detail not relevant for the solution of the problems they invoked the OAT for.

Lastly, we are convinced to have designed a highly modularized ontological tool, which is kept aware of recent developments in SRs and flexible enough to be adjusted to a very wide range of possible use scenarios.

6. Availability and requirements

HDOT and its modules are accessible at the Google Code page:

https://code.google.com/p/hdot/

OAT can be currently accessed and tested here:

http://ifomis.org/oat

The OAT will be fully integrated in p-medicine's portal and also accessible there from M38.

References

[Adamusiak et al., 2011] Adamusiak, T., Burdett, T., Kurbatova, N., van der Velde, K. J., Abeygunawardena, N., Antonakaki, D., Kapushesky, M., Parkinson, H., and Swertz, M. A. (2011). Ontocat - simple ontology search and integration in java, r and rest/javascript. BMC Bioinformatics, 12(1):218.

[Beißwanger et al., 2008] Beißwanger, E., Schulz, S., Stenzhorn, H., and Hahn, U. (2008). Biotop: An upper domain ontology for the life sciences: A description of its current structure, contents and interfaces to obo ontologies. Appl. Ontol., 3(4):205– 212.

[Euzenat et al., 2011] Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., and Trojahn, C. (2011). Journal on data semantics xv. chapter Ontology Alignment Evaluation Initiative: Six Years of Experience, pages 158–192. Springer-Verlag, Berlin, Heidelberg.

[Horridge and Bechhofer, 2011] Horridge, M. and Bechhofer, S. (2011). The owl api: A java api for owl ontologies. Semant. web, 2(1):11–21.

[Jonquet et al., 2010] Jonquet, C., Musen, M. A., and Shah, N. H. (2010). Building a biomedical ontology recommender web service. Biomedical Semantics, 1(S1). Selected in Pr. R. Altman's 2011 Year in Review at AMIA TBI.

[Keet, 2011] Keet, C. M. (2011). The use of foundational ontologies in ontology develop- ment: an empirical assessment. In Antoniou, G. et al., editors, 8th Extended Semantic Web Conference (ESWC'11), volume 6643 of LNCS, pages 321–335. Springer. Her- aklion, Crete, Greece, 29 May-2 June, 2011.

[Musen et al., 2012] Musen, M. A., Noy, N. F., Shah, N. H., Whetzel, P. L., Chute, C. G., Storey, M.-A. D., and Smith, B. (2012). The national center for biomedical ontology. JAMIA, 19(2):190–195.

[Pafilis et al., 2009] Pafilis, E., O'Donoghue, S. I., Jensen, L. J., Horn, H., Kuhn, M., Brown, N. P., and Schneider, R. (2009). Reflect: augmented browsing for the life scientist. Nature biotechnology, 27(6):508–510.

[Sabou et al., 2006] Sabou M, Lopez V, Motta E: Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner? 15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks, EKAW'06, Volume 4248 of Lecture Notes in Computer Science SpringerStaab S, Svátek V, Podebrady, Czech Republic 2006, 96-111.

[Salvadores et al., 2012] Salvadores, M., Horridge, M., Alexander, P. R., Fergerson, R. W., Musen, M. A., and Noy, N. F. (2012). Using sparql to query bioportal ontologies and metadata. In Proceedings of the 11th international conference on The Semantic Web - Volume Part II, ISWC'12, pages 180–195, Berlin, Heidelberg. Springer-Verlag.

[Whetzel et al., 2011] Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyu- las, C., Tudorache, T., and Musen, M. A. (2011). Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Research, 39(Web-Server- Issue):541–545.

Appendix 1 - Abbreviations and acronyms

API	Application Programming Interface
HDOT	Health Data Ontology Trunk
OAT	Ontology Aggregator Tool
OBO	Open Biological and Biomedical Ontologies
OWL	Ontology Web Language
RDF	Resource Description Framework
RDFS	RDF-Schema
REST	Representational State Transfer
RRF	Rich Release Format
SKOS	Simple Knowledge Organization System
SOA	Service Oriented Architecture
SPARQL	Simple Protocol and RDF Query Language
SR	Semantic Resource
URI	Unique Resource Identifier

Appendix 2 - Changes in HDOT suggested by OAT test results

a) Extensions of HDOT hierarchy

a1) "pathologist role"

We add the following class hierarchy from OBI under the HDOT class "role in human social processes":

investigation agent role

A role borne by an entity and that is realized in a process that is part of an investigation in which an objective is achieved. These processes include, among others: planning, overseeing, funding, reviewing.

http://purl.obolibrary.org/obo/IAO_0000122

reporting party role

A study personnel role played by a party who reports the outcome of a study component.

http://purl.obolibrary.org/obo/OBI_0000068

responsible party role

A study personnel role played by a party who is accountable for the execution of a study component and can make decisions about the conduct of the study

http://purl.obolibrary.org/obo/OBI_0000102

specimen collector role

An Investigation agent role borne by a person or organization which is realized in a specimen collection process.

http://purl.obolibrary.org/obo/IAO_0000120

worker role

a personnel role played by a party who executes a component of the study plan; this can occur before, during, after or outside the study timeline

http://purl.obolibrary.org/obo/OBI_0000116

The class "pathologist role" is subsumed by "worker role".

a2) "gene" and "mrt"

We add the following GRO-hierarchy under the HDOT_PM class "information biomacromolecule" next to "protein".

nucleic acid

A macromolecule made up of nucleotide units and hydrolysable into certain pyrimidine or purine bases (usually adenine, cytosine, guanine, thymine, uracil), D-ribose or 2-deoxy-D-ribose and phosphoric acid.

http://purl.obolibrary.org/obo/CHEBI_33696

DNA

A chain of deoxyribonucleotides.

http://www.bootstrep.eu/ontology/GRO#DNA

DNA molecule

A chain of deoxyribonucleotides as it occures at a whole.

http://www.bootstrep.eu/ontology/GRO#DNAMolecule

DNA region

A chain of deoxyribonucleotides that is part of a longer chain of deoxyribunucleotides.

http://www.bootstrep.eu/ontology/GRO#DNARegion

double strand DNA

Double stranded DNA molecules

http://www.bootstrep.eu/ontology/GRO#DoubleStrandDNA

nucleic acid molecule

A complex, high-molecular-weight biochemical macromolecule composed of nucleotide chains that convey genetic information. The most common nucleic acids are deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). Nucleic acids are found in all living cells and viruses.

http://www.bootstrep.eu/ontology/GRO#NucleicAcidMolecule

nucleic acid region

A sequence feature of nucleic acids

http://www.bootstrep.eu/ontology/GRO#NucleicAcidRegion

RNA

A chain of ribonucleotides.

http://www.bootstrep.eu/ontology/GRO#RNA

peptide

http://www.bootstrep.eu/ontology/GRO#Peptide

protein domain

A part of protein sequence and structure that can evolve, function, and exist independently of the rest of the protein chain. Each domain forms a compact three-dimensional structure and often can be independently stable and folded. Many proteins consist of several structural domains.

http://www.bootstrep.eu/ontology/GRO#ProteinDomain

The queried class "gene" is subsumed by "DNA region"; note that "gene" subsumes "mrt", such that this extension is sufficient. Since the URIs for GRO-classes do not conform to OBO conventions, we have to create own HDOT_PM URIs.

a3) "clinical study"

We add the following ERO-hierarchy under the HDOT class "planned process" (the ERO hit hierarchy is not exhaustive, except the branches below human study).

human study

Research project that uses or collects measurements or assessments about humans.

http://purl.obolibrary.org/obo/ERO_0000015

epidemiological study

A human study of diseases in populations of humans or other animals, specifically how, when and where they occur. Epidemiological studies can never prove causation, epidemiological evidence can only show that this risk factor is correlated with a higher incidence of disease in the population exposed to that risk factor. The higher the correlation the more certain the association, but it cannot prove the causation.

http://purl.obolibrary.org/obo/ERO_0000017

feasibility study

A preliminary study to determine the practicability of a proposed health program or procedure or of a larger study and to appraise the factors that may influence

its practicability. A feasibility study aims to discover those things which may affect successful study conduct on a larger scale.
http://purl.obolibrary.org/obo/ERO_0001402

qualitative human study
A qualitative study is an individual-human study whose primary mode of inquiry is qualitative.
http://purl.org/net/OCRe/research.owl#Qualitative_human_study

quantitative human study
A quantitative study is an individual-human study whose primary mode of inquiry is quantitative.
http://purl.org/net/OCRe/research.owl#Quantitative_human_study

interventional study
An interventional study is a quantitative study that prospectively assigns human participants or groups of humans to one or more health-related interventions to evaluate the effects on health outcomes. Interventions include but are not restricted to drugs, cells and other biological products, surgical procedures, radiologic procedures, devices, behavioural treatments, process-of-care changes, preventive care, etc.
http://purl.org/net/OCRe/research.owl#Interventional_study

observational study
An observational study is a quantitative study in which the investigators do not seek to intervene, and simply observe the course of events.
http://purl.org/net/OCRe/research.owl#Observational_study

The class "clinical trial" is subsumed by the ERO-class "interventional study". Again, non-standard URIs have to be replaced by HDOT-URI's

a4) "snp"

We add the following class hierarchy from OGI under the HDOT node "organismal quality".

genetic variation
Genetic variationis a variation in alleles of genes, occurs both within and among populations.
http://purl.obolibrary.org/obo/OGI_0000012

chromosome abnormality

A chromosome anomaly, abnormality or aberration reflects an atypical number of chromosomes or a structural abnormality in one or more chromosomes.

http://purl.obolibrary.org/obo/OGI_0000086

genetic polymorphism

strictly, the existense of two or more variants(alleles, phenotypes, sequence variants, chromosomal structural variants) at significant frequencies (i.e. more than 1%) in the population. Looser usages among molecular genetics include (1) any sequence variant present at a frequency >1% in a population (2) any non-pathogenic sequence variant, regardless of frequency.

http://purl.obolibrary.org/obo/OGI_0000089

genomic aberration

http://purl.obolibrary.org/obo/OGI_0000087

mutation

In genetics, a mutation is the result of the change of the nucleotide sequence of the genome of an organism, virus, or extrachromosomal genetic element.

http://purl.obolibrary.org/obo/OGI_0000088

The class "snp" is subsumed by the OGI class "genetic polymorphism". Note that we had to change the definition of "mutation", in order to distinguish the process of mutation from its result.

b) New labels for existing HDOT classes to ensure hits

a1) In the hit hierarchy from CPR, "psychological therapy" is subsumed under "Therapeutic act". Now we have a class "treatment" that is synonymous with "therapy". Hence we add the labels "therapeutic act" and "therapy" to "treatment".

a2) "tissue sample" is subsumed under OBI:Biospecimen. As it happens, we have the HDOT class "sample from organism" (from OBI) to which we add the label "biospecimen".

a3) "neoplasm" is added as an alternative label to the HDOT class "tumour".

c) Ad hoc addition to HDOT

The class

http://www.ebi.ac.uk/efo/EFO_0003919 "risk factor"

should be added under "role" (not, as the hit suggests, under "disposition").

As for "mass", the suggested hit hierarchy is of rather low quality (as material entities like "lump" are lumped together with information artifacts such as "clinical findings"). In this case, we have to make an educated guess. We interpret "lump" as generically as possible, namely as "mass of tissue" and add it under the HDOT class "object aggregate".

Appendix 3 – List of compiled terms for testing OAT

CRP

agonist

biomarker

bone marrow

brain

breast cancer

cancer

cell

cell proliferation

clinical trial

diagnosis

diet

disease

dna microarray

drug

fine needle aspiration biopsy

fine needle biopsy

gene

general health rating

head

hospitalization

lymph node

mass

mrt

neoplasm

pathologist

patient

psychological therapy

risk factor

sensitivity

snp

stem cell

tissue sample

tumor

tumor grade

tumor resection