D5.2 – An ontological framework for the capture and representation of stakeholder perspectives to augment collaboration and decision making

Due date of deliverable: 28 February 2011  
Actual submission date: 14 March 2011  
Responsible Partner: UOL  
Contributing Partners: UOL, UPM, FHG, CTI

Nature:  
■ Report  □ Prototype  □ Demonstrator  □ Other

Dissemination Level:  
■ PU : Public
□ 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)

Keyword List: Ontology Modelling, Ontology Engineering, Multi-layered Ontologies, Multiple Perspectives, Service Integration

The Dicode project (dicode-project.eu) is funded by the European Commission, Information Society and Media Directorate General, under the FP7 Cooperation programme (ICT/SO 4.3: Intelligent Information Management).
The Dicode Consortium

Research Academic Computer Technology Institute (CTI) (coordinator), Greece

University of Leeds (UOL), UK

Fraunhofer-Gesellschaft zur Foerderung der angewandten Forschung e.V. (FHG), Germany

Universidad Politecnica De Madrid (UPM), Spain

Neofonie GmbH (NEO), Germany

Image Analysis Limited (IMA), UK

Biomedical Research Foundation, Academy of Athens (BRF), Greece

Publicis Frankfurt GmbH (PUB), Germany
**Document history**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Modifications made by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>03-12-2010</td>
<td>Initial outline</td>
<td>Vania Dimitrova(UOL) &amp; Guillermo de la Calle Velasco (UPM)</td>
</tr>
<tr>
<td>2</td>
<td>01-02-2011</td>
<td>First Draft</td>
<td>Guillermo de la Calle Velasco (UPM), Axel Poigné (FHG), Vania Dimitrova (UOL), Ronald Denaux (UOL)</td>
</tr>
<tr>
<td>3</td>
<td>28-02-2011</td>
<td>Second draft – sent to reviewers</td>
<td>Vania Dimitrova (UOL), Manolis Tzagarakis (CTI), Ahmad Ammari (UOL), Fan Yang-Turner (UOL), Lydia Lau (UOL)</td>
</tr>
<tr>
<td>4</td>
<td>11-03-2011</td>
<td>Reviewers’ comments incorporated (version sent to SC)</td>
<td>Dhaval Thakker (UOL), Vania Dimitrova (UOL)</td>
</tr>
<tr>
<td>5</td>
<td>14-03-2011</td>
<td>Final (approved by SC, sent to the Project Officer)</td>
<td>Vania Dimitrova (UOL)</td>
</tr>
</tbody>
</table>

**Deliverable managers**
- Vania Dimitrova (UOL)
- Dhaval Thakker (UOL)

**List of Contributors**
- Ahmad Ammari, UOL (section 6.2.3)
- Axel Poigné, FHG (section 2.3)
- Dhaval Thakker, UOL (sections 1.2.3 and 3.3, 7)
- Fan Yang-Turner, UOL (sections 6.2.1 and 6.2.2)
- Guillermo de la Calle Velasco, UPM (sections 2.1 and 3.1)
- Lydia Lau, UOL (section 4.3)
- Manolis Tzagarakis, CTI (sections 4.1 and 4.2)
- Ronald Denaux, UOL (section 2.2)
- Spyros Christodoulou, CTI (sections 4.1 and 4.2)
- Vania Dimitrova, UOL (sections 1, 3.2, 4.3, 5 and 6)

**List of Evaluators**
- Stefan Rueping, FHG
- Manolis Tzagarakis, CTI
Summary

This deliverable outlines the Dicode ontological framework for the capture and representation of stakeholder perspectives to augment collaboration and decision making. Specifically, the ontological framework includes the following key issues:

- Identify the role of the Dicode Ontology (DON) in the Dicode project and specify the key research challenges that have to be addressed.
- Review the foundations for the design of the Dicode Ontology Framework, including languages and standards, ontology engineering methodologies, and services and semantics.
- Identify available ontologies that can be re-used for deriving the Dicode Ontology.
- Outline an agile Dicode Ontology Framework including the main iterations in creating and using the ontology, as well as the key ontology layers.
- Provide a description of the first iteration of the Dicode Ontology Framework, focusing on the modelling of generic and use-case specific layers.
Table of Contents

1 Introduction ................................................................................................................................. 7
  1.1 Why does Dicode need an Ontology? ...................................................................................... 7
  1.2 Key Research Challenges ......................................................................................................... 8
    1.2.1 Data, Service, and Stakeholder Heterogeneity ................................................................. 8
    1.2.2 Specificity and Generality of Dicode Use Cases and Services ...................................... 9
  1.3 Relevant European Research Projects ................................................................................... 11
  1.4 Conclusions ............................................................................................................................ 17

2 DON Framework Foundations .................................................................................................. 18
  2.1 Introduction ............................................................................................................................. 18
  2.2 Languages and standards ......................................................................................................... 18
    2.2.1 Extensible Markup Language (XML) ................................................................................ 18
    2.2.2 Simple HTML Ontology Extensions (SHOE) ................................................................. 18
    2.2.3 XML-Based Ontology Exchange Language (XOL) ...................................................... 20
    2.2.4 Resource Description Framework (Schema) - (RDF(S)) ............................................ 21
    2.2.5 Ontology Web Language (OWL) ................................................................................... 22
    2.2.6 Ontology languages comparison .................................................................................... 22
  2.3 Ontology Engineering Methodologies .................................................................................... 23
    2.3.1 Traditional Methodologies ............................................................................................. 23
    2.3.2 Methodologies to Actively Involve Domain Experts ................................................... 25
    2.3.3 Tools for Intuitive Ontology Engineering ..................................................................... 26
  2.4 Services and Semantics ........................................................................................................... 28
    2.4.1 Introduction ...................................................................................................................... 28
    2.4.2 Service Oriented Architectures ...................................................................................... 28
    2.4.3 Service Description Languages ..................................................................................... 29
    2.4.4 Semantic Technologies for Services ............................................................................ 31
    2.4.5 Semantic-enhanced Service Coordination ................................................................... 34
    2.4.6 Conclusions .................................................................................................................... 35

3 Available Ontologies for Use-case Specific Layers in DON .................................................. 36
  3.1 Medical Ontologies Relevant to Use Cases 1 and 2 ............................................................. 36
    3.1.1 Biomedical ontologies .................................................................................................... 37
    3.1.2 Biomedical ontologies search engines ......................................................................... 42
  3.2 Ontologies Relevant to Use Case 3 ........................................................................................ 43
  3.3 Open Datasets relevant to the use cases ............................................................................... 44

4 Available Ontologies for the DON Generic Layer .................................................................. 45
  4.1 Introduction ............................................................................................................................. 45
  4.2 Collaboration Ontologies ....................................................................................................... 45
    4.2.1 3C-based Collaboration Ontology .................................................................................. 45
    4.2.2 Collaboration Networks Ontology .................................................................................. 45
    4.2.3 Collaboration Ontology based on Collaboration Engineering ..................................... 46
  4.3 Decision Making Ontologies ................................................................................................ 47
    4.3.1 Decision Support Ontology ............................................................................................ 47
    4.3.2 Ontology for Information Systems Engineering ............................................................ 48
    4.3.3 Ontology for Decision Support in Health Informatics .................................................... 49
1 Introduction

In recent years, the use of ontologies has become very popular in scientific areas to share information at different levels, from people to software applications or databases. Ontologies constitute the key factor for the emerging Semantic Web to represent the semantics of web documents. Ontologies have become critical to integrate information coming from different sources. Applications use ontologies to carry out different tasks such as searches, transformations or merges of information\(^1\). Another key feature is the accessibility of the ontologies for the applications or agents involved in the communication. Ontologies do not only have to be physically available, but must be encoded using a language legible by all parties, both human and computers. Different ontology languages have arisen during the last few years to facilitate codification. In parallel, several methodologies and development frameworks have been defined. Along this deliverable, we analyze the most relevant providing a short description of each one. Based on such frameworks, methodologies and languages, we will create our own environment (ontological framework) for the Dicode project, which will be presented in this deliverable.

Section 1 of the deliverable highlights the role for an ontological framework in the Dicode project and key research challenges in forming such framework. Section 2 lays the foundation technologies and methodologies on which the ontology framework will be built. The state of art survey is covered in the sections 3 and 4 with respect to the existing research efforts relevant to the Dicode ontology framework. Section 5 and 6 outlines the development iterations expected to take place in building such framework.

1.1 Why does Dicode need an Ontology?

“An ontology is an explicit specification of a conceptualization” (Gruber, 1993). Specifically, an ontology is a formal, shared and consensuated representation of the knowledge within a specific domain. Ontologies include a set of concepts, organised in a hierarchy, and relationships among such concepts. In practice, ontologies have been used to establish a common vocabulary shared by different agents to ensure the communication and understanding among them.

Ontologies have been designed, developed and used for different purposes (Denaux et al. 2009), such as:

- Enabling knowledge sharing and reuse between different applications and services;
- Enabling interoperability between different applications and services;
- Gaining a better understanding of a specific domain;
- Facilitating comparison between domains;
- Influencing data processes, e.g. creation, editing, navigation, annotation and publishing (this mainly refers to using ontologies to enable linked data);
- Providing a knowledge base to study a specific system;
- Facilitating collaboration processes (when an ontology provides a common vocabulary that enables social processes between practitioners in a specific domain);
- Enhancing existing applications (e.g. improve information extraction techniques);

In Dicode, an ontology (called **Dicode Ontology** and referred hereafter as **DON**) will be implemented and used for some of the purposes listed above, namely:

\(^1\) [http://www.w3.org/TR/webont-req/]
• Gaining an **understanding** of data intensive collaboration and decision making;
• Enabling **interoperability** between different **Dicode services**, as well as interoperability between Dicode services and external services which facilitate data-intensive processes;
• Facilitating the **application and reuse** of **Dicode services** in a given use case which requires data intensive collaboration and decision making;
• Enhancing the **functionality** of specific **Dicode services** (e.g. social media mining or community modelling);
• Providing a **common vocabulary** to facilitate collaboration between **Dicode service developers**.

### 1.2 Key Research Challenges

The specificity of the Dicode project brings forth several research challenges to the DON-Framework which underpins the development and utilisation of DON.

#### 1.2.1 Data, Service, and Stakeholder Heterogeneity

Dicode is a highly heterogeneous project aimed at facilitating data-intensive collaboration and decision making. The heterogeneity comes from several aspects.

**Data heterogeneity.** Dicode deals with data coming from **diverse disciplines**, such as biological, medical, or web data. Furthermore, the data to be processed comes in **different formats**: structured (e.g. database tables or spreadsheet data), semi-structured (e.g. decision provenance data, log data from guided argumentation and negotiation), up to unstructured (e.g. data coming from social media, such as blogs, forums, tweets). Such diversity constitutes a great challenge regarding the semantic management models and processes in Dicode. DON has to cater for a diversity of data sources, and to provide a flexible mechanism for expanding and refining the semantic description of data used in Dicode.

**Service heterogeneity.** Dicode services will support a **broad range of data-intensive tasks**, e.g. data discovery, data cleaning, data analysis, and data interpretation. Although, there are generally common goals in data-intensive tasks (e.g. discovering patterns of data, inferring data content relationships and rules from them, discovering trends, comparing alternatives, forming opinions), their realisation in intelligent services varies significantly. Furthermore, Dicode considers the synergy of technology and people, and aims at developing services to facilitate this synergy. This imposes a key challenge to the semantic modelling mechanism that has to accommodate the involvements of **both machines and humans** in data-intensive collaboration and decision making. DON has to enable the interoperability among the services created and used by the different use cases, and to assure the integration of various heterogeneous services.

**Stakeholder heterogeneity.** Dicode services will be developed and utilised by a range of stakeholders, including: **service developers** (coming from different technical disciplines, e.g. data mining, text mining, argumentation, collaboration support, distributed systems); **service customers** (system administrators and developers of applications which utilise Dicode services, and may also use other services developed outside the Dicode project); **service users** (end users of the applications that utilise Dicode services in their collaboration and decision making practice). These groups have different background, use different vocabularies, and have different expectations from the Dicode services. It is a major challenge to provide a flexible semantic model that accommodates the stakeholder diversity, and engages the various stakeholders in ontology composition and use.
1.2.2 Specificity and Generality of Dicode Use Cases and Services

Dicode considers three use cases with a significant variation of their data intensive collaboration and decision making practice.

Use Case 1: Dicode in a Scientific Research Team

This use case illustrates a typical use of data intensive services for collaborative decision making in a scientific research team. A Breast Cancer research group is embarking on an analysis to discover any common characteristics or trends that could be deducted from recent studies which used high-throughput technologies such as microarrays and next-generation sequencing. This use case provides scientific research community perspective on data intensive collaboration and decision making.

Generic aspects. There appear to be some generic decision tasks, such as:
- data discovery;
- making judgement about data sets;
- discover common characteristics;
- identify trends.

There are also some generic collaboration tasks, such as:
- group meeting (discussions);
- collective planning;
- cooperation (split work between people depending on their roles).

Specific aspects. There is a specific way of conducting the tasks, which depends on:
- the organisational practice of this research team;
- the roles undertaken by team members (lead researcher, research assistant);
- the specific domain (Biological and Gene analysis);
- the availability of data sources (structured data in matrix format, data base tables);
- the possibility to use external collaboration (e.g. other researchers’ opinions on data sets).

Tacit/intuitive aspects. Many of the tasks involve tacit knowledge of researchers and their intuitive decision making (e.g. an expert researcher is capable of pointing at leading authors to check, make intuitive judgement on the suitability of the sources, perceive gaps in the analysis). This human intelligence is not properly captured and capitalised upon to augment the development and use of data intensive services.

Use Case 2: Dicode in Collaborative Medical Diagnosis

This use case illustrates a typical use of data intensive services for collaborative medical diagnosis. A clinical trial of Rheumatoid Arthritis treatment is conducted to evaluate the effectiveness of the treatment, by analysing the condition in wrists and possibly other joints over the period of treatment. A team of medical practitioners has to examine the data sets and analyse the effectiveness of the treatment on each patient. This use case provides collaborative medical diagnosis perspective on data intensive collaboration and decision making.

Generic aspects. There appear to be some generic decision tasks, such as:
- identify change;
- identify tendency;
• compare data sets (e.g. for different patients);
• combine several data sets.

There are also some generic collaboration tasks, such as:
• discuss;
• give opinion;
• provide justification;
• cooperation (split work between people depending on their roles).

Specific aspects. There is a specific way of conducting the tasks, which depends on:
• the organisational practice of this medical diagnosis team;
• the member roles (radiologist, clinician, patient);
• the specific domain (Medicine, arthritis);
• the specific data sources being used (images, meta data, provenance data);
• limitations regarding security and access control;
• need for quality monitoring and reviewing of the decision making process.

Tacit/intuitive aspects. Some tasks involve tacit knowledge and intuitive judgement by experts (e.g. a radiologist identifies regions of interest in an image, a clinician makes a judgement on the arthritis condition). This human intelligence can be captured and capitalised upon to augment the development and use of data intensive services.

Use Case 3: Dicode in Product Monitoring and Marketing
This use case illustrates a typical use of data intensive services for a product online image analysis, in order to compile the product’s marketing strategy plan. A Brand Consultancy company is to embark on an analysis of a Client’s current product image before recommending an action plan for the Client’s marketing strategy. To build up a strategy, it is crucial to have a valid foundation (i.e. an evidence based approach). For this, a team from the company conducts analysis of the brand image in online social media.

Generic aspects. There appear to be some generic decision tasks, such as:
• discover data sets;
• identify certain characteristics (e.g. positive and negative sentiments);
• analyse changes over time (e.g. tracking brand image over time);
• identify patterns (e.g. to find key influencers);
• conduct comparison (e.g. compare brands or products).

Some generic collaboration tasks may also be associated with this use case:
• discuss;
• assess alternatives;
• provide argumentative judgements;
• create a shared outcome (marketing plan).

Specific aspects. There is a specific way of conducting the tasks, which depends on:
• the organisational practice of this marketing consultancy company;
• the member roles (team leader, marketing researcher);
• the specific domain (sentiment analysis);
• the specific data sources being used (unstructured social data).
Tacit/intuitive aspects. Some tasks involve tacit knowledge and intuitive judgement by experts (e.g. judging how influential certain opinions could be or discovering emerging themes). This human intelligence can be captured and capitalised upon to augment the development and use of data intensive services.

To sum up, by capturing the generic and specific aspects of data intensive collaboration and decision making in the three Dicode use cases, DON will enable gaining a better understanding of the needed support from data-intensive services. It will also enable a systematic comparison between the use cases, and will facilitate the adoption of Dicode in use cases beyond the project. Finally, as seen above, each of the use cases has intuitive decision making aspects, related to tacit knowledge expert human decision makers apply. These aspects, if captured, can point at some heuristics how to further augment the decision making processes (Salas et al, 2010). One way of eliciting tacit knowledge is through an ontology (Chin & Xin, 2008) which can be derived by involving experts or by mining team discussions. We envisage that DON can capture some tacit aspects in an explicit way.

1.2.3 Intuitive Ontology Authoring and Use

Active involvement of stakeholders. A key challenge in ontology authoring is the active involvement of domain experts (which in the case of Dicode includes the various stakeholders). The domain experts are the actual holders of expertise to be encoded in the ontology and provide the prime sources for formulating knowledge constructs and validating the ontology. Most domain experts lack knowledge engineering skills and are unfamiliar with logical formalism for coding and querying ontologies (Dimitrova et al., 2008). This is certainly the case in Dicode where most stakeholders have not been engaged in ontology engineering. Furthermore, due to the high diversity and heterogeneity in the Dicode use cases and the broad scope of data intensive collaboration and decision making, the knowledge engineers are unlikely to be able to grasp the full complexity of the domain expertise. Therefore, the DON Framework should consider appropriate ways to facilitate stakeholders' engagement in ontology construction and use.

Semantic Enrichment of Service Descriptions. Dicode envisages the use of a range of web services, some of which will be developed by Dicode partners, and others can come from relevant external services. To facilitate the automatic service discovery, execution, composition, and interoperation for specific needs, semantic-enriched descriptions of services should be provided (McIlraith et al, 2001). A semantic layer to service description typically includes (Dietze et al, 2011):

- **functional semantics** (e.g. input and output, pre and post conditions);
- **data semantics** (description of the data that is being passed between services);
- **quality-of-services semantics** (description of the quality of services);
- **execution semantics** (the execution pattern of these services, the pattern of the entire process – which could be in the form of workflows).

In the case of Dicode, the semantic layer will be meta-data describing the aspects of data intensive collaboration and decision making each service addresses, considering the four dimensions above and linking to appropriate DON classes. To enable this, the DON Framework should include intuitive ways to compose semantic layer of Dicode services (e.g. by providing intuitive ways to compose semantic service descriptions or by exploiting automatic annotation techniques to derive semantic layer from service descriptions in natural language).
Semantic enriched data and text mining. There are ways semantics can augment various tasks in Data mining systems (Kuo et al, 2007).

- For example, one of the challenges data mining developers faces is the meaning of the attributes/metrics is not easily understood by them as in most cases they are not familiar with the domain they are working with. Semantic can provide domain knowledge in such cases to aid the data mining developers’ understanding of the domain.
- Interpreting data mining results is dependent to some extent on domain knowledge.
- Semantics can help with the understanding of the dataset, relationships among variables, known casual relations.
- In tasks such as, association rules mining, semantics can be used at the data preparation stage by categorizing variables in semantic groups.
- Semantics are helpful to address differences within the definition of variables (such as dod -> date of death)

In the case of text mining systems (Stavrianou et al, 2007),

- Semantics can provide a bootstrapping model for text mining to find new entities and detect patterns.
- Semantics guides the text mining process in terms of what to extract (entities) and what attributes of entities to look for (Thakker et al, 2010).
- Ontologies ease the burden of metadata management which sometime can be overwhelming with the development of the Text Mining systems.

The DON framework will investigate the ways semantics can play a role in data and text mining tasks.

1.3 Relevant European Research Projects

The DON Framework will build upon, and further contribute to, existing European research in intelligent content and semantics. Table 1 positions the Dicode ontological approach within the context of relevant European projects. Table 2 compares the DON Framework to ontological frameworks applicable to use cases relevant to Dicode use cases.
Table 1. DON Research Context: Comparison with Relevant European Research Projects.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>Relevance for Dicode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIP: Data, Information, and Process Integration with Semantic Web Services</td>
<td>DIP has defined and implemented additional layers of functionality on top of the current Web Service technology stack. The DIP framework was utilised in case studies for e-work, e-commerce, and e-government.</td>
<td>Dicode follows the DIP’s approach for using ontologies and semantic descriptions of services. The DIP semantic layer for service description will be followed. DON will enable further application of the DIP framework in a new domain.</td>
</tr>
<tr>
<td>ASPIC: Argumentation Service Platform with Integrated Components</td>
<td>ASPIC has developed a common framework to underpin the services that are emerging as core functions of the argumentation paradigm. These include reasoning, decision-making, learning and communication.</td>
<td>Dicode is directly relevant ASPIC, especially because both projects provide generic semantic model. APSIC considers only one kind of data-intensive collaboration and decision making services, while DON will offer a broader description. Also, DON and the semantic services for ontology engineering and service annotation will be provided as open source.</td>
</tr>
<tr>
<td>@neurIST: Integrated Biomedical Informatics for the Management of Cerebral Aneurysms:</td>
<td>neurIST provides an integrated decision support system to assess the risk of aneurysm rupture in patients and to optimize their treatments. An IT infrastructure has been developed for the management and processing of a vast amount of heterogeneous data acquired during diagnosis.</td>
<td>Similarly to neurIST, Dicode will consider a broad range of stakeholders involved in the decision making process, and will describe how different services will be used by stakeholders. Compared to the ontology developed and used in neurIST, DON will have a broader scope, and will enable not only intuitive ontology inspection (as in neurIST) but also intuitive construction of the ontology based on a range of sources and involving people without knowledge engineering skills (domain experts did not have active involvement in the actual creation of the neurIST ontology).</td>
</tr>
<tr>
<td>Knowledge Web:</td>
<td>A European Network of Excellence that developed the foundations of Semantic Web technologies.</td>
<td>Dicode will follow the standards and guidance developed by Knowledge Web, specifically: semantic web services, ontology engineering methodologies, and evaluation of ontology frameworks.</td>
</tr>
</tbody>
</table>
Table 1(cont.). DON Research Context: Comparison with Relevant European Research Projects.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>DON Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NEON: Networked Ontologies</strong></td>
<td><a href="http://www.neon-project.org/">http://www.neon-project.org/</a></td>
<td>NEON developed a toolkit of services for producing ontologies in larger numbers and exhibiting greater complexity. Some challenges addressed by NEON are: using context for developing, sharing, adapting and maintaining networked ontologies; and making it easier for users with different levels of expertise and experience to browse and make sense of ontologies. DON will follow the Ontology Lifecycle model defined by NEON, and will provide validation of this model in a complex domain (data intensive collaboration and decision making).</td>
</tr>
<tr>
<td><strong>TAO: Transitioning applications to ontologies</strong></td>
<td><a href="http://www.tao-project.eu/">http://www.tao-project.eu/</a></td>
<td>TAO created an open source infrastructure to aid transitioning of legacy applications to ontologies, through automatic ontology bootstrapping, semantic content augmentation, and generation of semantic web service descriptions. Dicode builds on state-of-the-art reviews in semantic web services and ontology authoring by TAO. Dicode will use some of the tools from TAO (e.g. the Gate information extraction suite and the automatic service annotation tools).</td>
</tr>
<tr>
<td><strong>Service Finder: Realizing Web Service Discovery at Web Scale</strong></td>
<td><a href="http://www.service-finder.eu/">http://www.service-finder.eu/</a></td>
<td>Service-Finder aims at developing a platform for service discovery in which Web Services are embedded in a Web 2.0 environment. It includes: Semantic Search Engine for Web Services; Web Service Crawler; Automatic generation of Semantic Service Descriptions. Dicode will follow the Service Finder approach for automatic semantic annotation of services.</td>
</tr>
</tbody>
</table>

The DON Framework will exploit these frameworks and conceptually extend these:

- DON will enable application of the DIP framework in a new domain.
- While APSIC considers only one kind of data-intensive collaboration and decision making services, DON will offer a broader description. Also, DON and the semantic services for ontology engineering and service annotation will be provided as open source.
- Compared to the ontology developed and used in neurIST, DON will have a broader scope, and will enable not only intuitive ontology inspection (as in neurIST) but also intuitive construction of the ontology based on a range of sources and involving people without knowledge engineering skills (domain experts did not have active involvement in the actual creation of the neurIST ontology).
Taking NEON’s definition of context and perspective, Dicode will extend this by considering the heterogeneity and use case specificity/generality. The Dicode semantic services will integrate services from the NEON toolkit and will provide further services for intuitive and iterative ontology authoring by people without knowledge engineering skills (the NEON tools require logical background during ontology creation, and consider domain experts involved in ontology inspection only).

- Dicode will extend TAO’s tools to provide intuitive ontology engineering and service annotation tools in a highly heterogeneous domain
- The lightweight ontological approach by Service Finder will be extended to deal with heavy weight, multi-layered ontology. Furthermore, DON will offer a holistic model that describes the domain, the data, and the services (Service Finder instead developed separate ontologies).
Table 2. Comparison of the DON Framework and semantic frameworks in relevant use cases.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>DON Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALBC: Collaborative Annotation of a Large Biomedical Corpus: <a href="http://www.calbc.eu/">http://www.calbc.eu/</a></td>
<td>CALBC will provide a web based system that allows any developer of biomedical text-mining applications to submit their annotation of the corpus, determine how this submission diverges from all the others and exploit this information to improve its performance. The final corpus, together with its final annotation will be made publicly available in June 2011.</td>
<td>CALBC is particularly relevant to defining the semantic model for Dicode use case 1. Currently, there is no publicly available semantic corpus. When it is available, it may provide a useful source for research data descriptions. The notion of perspectives, considered as different annotations in CALBC, will be further extended in Dicode by considering wider dimensions linked to use cases and stakeholders.</td>
</tr>
<tr>
<td>e-LICO: e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Sciences <a href="http://www.e-lico.eu/">http://www.e-lico.eu/</a></td>
<td>e-LICO aims to link data miners (in quest of data to feed their sophisticated tools) and domain scientists (who must confront massive data). The project aims at developing efficient technologies for intelligent knowledge extraction from globally growing loads of images, text and other structured data and also at fostering the use of these data mining methodologies in data-intensive sciences. e-LICO will be demonstrated on a systems biology approach to disease studies, with focus on diseases of the kidney and urinary pathways.</td>
<td>e-LICO has a very similar goal to Dicode. Hence, the DON Framework will make a use of the ontological models defined for each of e-LICO’s layers: e-science; data mining; and application layer.</td>
</tr>
<tr>
<td>WeKnowIt: Emerging, Collective Intelligence for personal, organisational and social use <a href="http://www.weknowit.eu/">http://www.weknowit.eu/</a></td>
<td>WeKnowIt develops tools for exploiting multiple layers of intelligence from user-contributed content, which together constitute Collective Intelligence.</td>
<td>WeKnowIt is relevant to use case 3. Specifically, DON will make use of the semantic description of social media data developed by WeKnowIt. The ontological framework in Dicode is much broader that the ontological framework in WeKnowIt.</td>
</tr>
</tbody>
</table>
The DON Framework will exploit these frameworks and conceptually extend these:

- The notion of perspectives, considered as different annotations in CALBC, will be extended in Dicode by considering wider dimensions linked to use cases and stakeholders.

- While e-LICO considers a layered approach of the services and separate ontologies for each layer, Dicode will apply a more holistic approach delivering one ontology but distinguishing several layers and perspectives in it. Also, while e-LICO focuses on a specific domain, Dicode considers a broader diversity, and in that respect, the ontological framework would be more generic. The ontology authoring in e-LICO is conducted in the ‘classical’ way by knowledge engineers, while the creation of DON will involve various Dicode stakeholders.

- Dicode intends to provide a much broader ontological framework than that of WeKnowIt.

1.4 Conclusions

Dicode will address the key research challenges by extending the work of relevant European research projects and available semantic frameworks guided by the demands of the use cases. The main contribution and novelty of the Dicode ontological framework will come from in-depth analysis and semantic description of data-intensive collaboration and decision making, providing a generic, holistic ontological approach applied to three diverse use cases. Unique aspects of the Dicode ontological approach are:

- domain – there is no ontological model based on a comprehensive description of decision making or collaboration tasks integrated with service description;

- generic versus specific layers – DON will identify what aspects are generic for use cases (i.e. will be close at the service level description) and which are specific (i.e. will be close at use case description);

- different perspectives – DON considers perspectives to apply to different stakeholder categories or the specific use cases;

- considering the role of both machines and humans in data processing services – current ontological frameworks tend to miss the people’s role in decision making, missing important tacit knowledge of intuitive decision making. DON will aim to explicitly capture this knowledge by involving various stakeholders in ontology authoring and analysing stakeholders’ feedback on service usage;

- intuitive and iterative ontology construction and use – the DON framework will integrate intuitive tools for constructing and validating an ontology by people who do not possess knowledge engineering skills and lack experience in ontology construction.
2 DON Framework Foundations

2.1 Introduction

This section will present the foundations of the DON framework, including ontology languages and standards, ontology methodologies, services and semantics. The aim is to identify the technologies to be used, the ontology engineering methodologies to be adopted, and the semantic integration approaches to be followed in the Dicode Ontology Framework.

2.2 Languages and standards

The evolution of the World Wide Web and the growing need of sharing information and working in a cooperative way, among heterogeneous agents, have driven experts to create knowledge representations about the concepts they are using together. Such knowledge representation is referred as ontology. Creating sufficiently expressive human-readable ontologies as well as making this knowledge accessible to machines, is one of the key points of ontology languages. During the last years, several ontology languages have emerged for implementing ontologies (Corcho & Gómez-Pérez, 2000). In this section, we review the most relevant standards and ontology languages used to develop ontologies.

2.2.1 Extensible Markup Language (XML)

XML, the Extensible Markup Language, is a standard format that allows users to define specific language grammars. Since the Web started to be the most important information repository, new ontology languages were defined based on XML. These ontology languages are known as web-based ontology languages or ontology markup languages (Gomez-Perez, 2004) and their syntax is mainly based on XML.

![Figure 1. Ontology markup languages (taken from (Gomez-Perez, 2004)).](image)

2.2.2 Simple HTML Ontology Extensions (SHOE)

SHOE (Heflin & Hendler, 2000) was the first ontology markup language. It extends HTML with a set of knowledge oriented tags that, unlike HTML tags, provide structure for knowledge acquisition as opposed to information presentation. Thus, its philosophy is to provide web-content semantic information to machines.

```html
<HTML>
<HEAD>
 <META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">
 <TITLE>My Page</TITLE>
 </HEAD>
<BODY>
<P>Hi, this is my web page.
```
I am a graduate student and a research assistant.

Also, I'm 52 years old.

My name is George Stephanopolous.

Here is a pointer to my graduate advisor:

And is a paper I recently wrote.

This example about how to use SHOE is taken from its Official Web Page (http://www.cs.umd.edu/projects/plus/SHOE/). It shows how to extend a simple HTML web page with a defined set of tags that make its content accessible to agents. It is instantiating concepts and attributes belonging to the cs-dept-ontology ontology. Below is expressed how to define an attribute. Note that it is possible to declare range and domain restrictions but cardinality, default values or another constraints can not be expressed. Moreover, in SHOE we can not define class attributes nor instance attributes whose value is inherited for the concept subclasses and instances (Gomez-Perez et al, 2004). For example, let suppose that the advisor is always the same for all the students: Mark Ericseen. This could not be expressed with SHOE neither in the category nor relation definitions and inherit in the instances, but it will have to be declare in each instance or as an axiom as it will be shown later.

Disjoint and Exhaustive knowledge could not be expressed with SHOE. Neither functions because the tag cannot declare which are input or output arguments. A function needs a mechanism for expressing a set of input arguments that output only one value.
Axioms in SHOE are expressed as Horn rules (H:-B), that is H, the head, happens if B, the body, holds. The head is inside the `<INF-THEN>` tag while the body is inside the `<INF-IF>` tag. In the example above is being stated that if an undergraduate student is member of CS department then he or she is advised by John Doe. As well as the specification of SHOE as a superset of HTML, the SHOE team defined a formal syntax of SHOE as an XML application.

**2.2.3 XML-Based Ontology Exchange Language (XOL)**

XOL (Karp et al, 1999) is a definition of some of the OKBC primitives (OKBC-Lite) in XML. It was thought for being used as an intermediate language for transferring ontologies among different tools. Unlike most of the ontology languages XOL does not support to import knowledge from other ontologies.

```xml
<class>
  <name>person</name>
</class>
<slot>
  <name>age</name>
  <domain>person</domain>
  <value-type>integer</value-type>
  <numeric-max>150</numeric-max>
</slot>
```

XOL supports classes, properties (slots), slot constrains (facets) and instances (individuals). In the XOL code concepts person and age are defined. Also, age is related to a person and it is being constrained with the following facets: Age must be an integer top-bounded to 150.

Slots, as in the predecessor OKBC protocol, could be template slots or own slots. A template slots is a slot in a class that is automatically attached to its subclasses and instances (inheritance) while an own slot is isolated and can be attached to any class (By default slots are own).

Moreover, is possible to express concept hierarchy thanks to the subclass-of primitive but disjointness and exhaustiveness is not possible to state using XOL. The following example shows how to create hierarchies.

```xml
<class>
  <name>man</name>
  <documentation>
    The class of all persons who define their gender as male.
  </documentation>
  <subclass-of>person</subclass-of>
</class>
```

Relations are also expressible in XOL using the primitive slot, domain and range are stated with domain and value-type primitives respectively. In XOL, in contrast to SHOE, all the relations are binary (Ternary or higher relations in XOL are defined by reification or, what is
the same, creating a class for the relation itself). Finally, in XOL is possible to express relational cardinality using the primitives slot-maximum-cardinality, slot-minimum-cardinality and slot-cardinality and inverse semantic relationship using slot-inverse (Karp et al., 1999).

As XOL is intended just for exchanging ontologies it does not define any kind of reasoning mechanism. Thus, it is not possible to express axioms or functions with XOL.

2.2.4 Resource Description Framework (Schema) - (RDF(S))

RDF(S), that is Resource Description Framework and Resource Description Framework Schema, is one of the knowledge representation mechanisms most used nowadays. It is the union of two different Knowledge Representation formalisms: Semantic Networks and Frames and is widely spread due to the Semantic Web and Linked Data boom. RDF is intended for representing meta-data about “Web Resources” understanding as resource a Web Page, items from an on-line shopping facilities, a description of a Web user's preferences, a concrete person, etc. RDF is based on the semantic network formalism where both entities and relations between them are Web Identifiers (Uniform Resource Identifiers or URIs). Therefore, a RDF resource could be seen as a set of triple statements as (subject, property, object), for instance, (Eric Miller, has_mail, em@w3.org) where the domains of subjects and properties are URIs and the domain of the objects could be both URIs and or another data type as Strings, Integers, etc.

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

In the last piece of RDF code, it can be seen an example where is being said that there is a person identified as http://www.w3.org/People/EM/contact#me whose name is Dr. Eric Miller and his mail address is em@w3.org. Thus, is seen how RDF allows users to link Web Resources with a semantic meaning.

RDF Schema extends RDF including the concept of class and adding the notion of hierarchy. RDFS is based on the Frames formalism and it defines the concepts of class, instance, subclass of, subproperty of and property range and domain among others.

```xml
<rdfs:Class rdf:about="http://www.ontologies.org/ontologies/people#Professor">
  <rdfs:subClassOf>
    <rdfs:Class rdf:about="http://www.ontologies.org/ontologies/people#Person"/>
  </rdfs:subClassOf>
</rdfs:Class>
```

In this example (RDF Vocabulary Description Language 1.0: RDF Schema, W3C, 2004) is being defined that a Professor is subclass of a Person. One of the most important weakness of RDFS emerges when is necessary to describe any resource in a more detailed way, i.e. heavy-weighted ontologies: Disjoint or Exhaustive knowledge can not be expressed with RDFS nor existence/cardinality constrains, etc. The lack of formal semantics to create
RDF(S) has caused several problems to define the semantics of other languages that extend RDF(S) like OIL, DAML+OIL, and OWL (Gomez-Perez et al., 2004).

2.2.5 Ontology Web Language (OWL)

The Ontology Web Language (OWL) in its second version (OWL 2) provides classes, properties, individuals, and data values. Ontologies in this language are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents (Grau et al, 2008).

OWL 2 ontologies could be located within three different layers depending on the ontology expressiveness and restrictiveness: OWL Lite (Small subset of primitives), OWL DL (Includes description logic) and OWL Full (relaxes some of the constraints on OWL DL to preserve some compatibility with RDFS). OWL 2 allows balancing between expressiveness and restrictiveness locating the ontology closer to be expressive or restrictive depending on how it will be used.

An OWL 2 ontology is a formal (and desirably reached by consensus) specification of a concrete domain of interest. OWL 2 ontologies are composed by the next three syntactic categories:

- **Entities**: such as classes, properties, and individuals or instances, are identified by Internationalized Resource Identifiers (IRIs) and compose the primitives of an ontology.
- **Expressions**: represent complex notions in the domain being described.
- **Axioms**: statements that are asserted to be true in the domain being described.

Another important characteristic of OWL 2 is that it allows to import already created ontologies. Thus, is possible to reuse knowledge already used satisfactorily and, what is more important, reached by experts to consensus.

Summarizing, OWL 2 extends RDFS in order to give a formal semantics and includes some of the mechanism that RDFS does not provide for creating heavy-weighted ontologies and reasoning in a more powerful way.

2.2.6 Ontology languages comparison

Table 3 presents a comparison among the different ontology languages.

<p>| Table 3. Ontology language comparison. |</p>
<table>
<thead>
<tr>
<th></th>
<th>SHOE HTML</th>
<th>XOL</th>
<th>RDF(S)</th>
<th>OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Import</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Concepts</strong></td>
<td>Yes (Categories)</td>
<td>Yes (Classes)</td>
<td>Yes (Classes)</td>
<td>Yes (Classes)</td>
</tr>
<tr>
<td><strong>Attributes</strong></td>
<td>Yes (Relations)</td>
<td>Yes (Slots)</td>
<td>Yes (Properties)</td>
<td>Yes (Properties)</td>
</tr>
<tr>
<td><strong>Default Values</strong></td>
<td>No</td>
<td>Yes (Own slots inside class definition)</td>
<td>No</td>
<td>Yes (owl:hasValue, not in OWL DL)</td>
</tr>
<tr>
<td><strong>Relations</strong></td>
<td>Yes (Relations)</td>
<td>Yes (Slots)</td>
<td>Yes (Properties)</td>
<td>Yes (Properties)</td>
</tr>
<tr>
<td><strong>Range and Domain</strong></td>
<td>Yes (&lt;arg type=&quot;&quot;&gt;)</td>
<td>Yes (&lt;domain&gt;and&lt;slot-value-type&gt;)</td>
<td>Yes (rdfs:domain and rdfs:range)</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Relational Arity</strong></td>
<td>All n-arity</td>
<td>Yes (&lt;slot-minimum-cardinality&gt;, &lt;slot-maximum-cardinality&gt; and &lt;slot-cardinality&gt;)</td>
<td>All n-arity</td>
<td>Yes (owl:minCardinality, owl:maxCardinality, and owl:cardinality)</td>
</tr>
<tr>
<td><strong>Disjointness and Exhaustiveness</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Functions</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes (Binary) (owl:Functional Property and owl:InverseFunctional Property)</td>
</tr>
<tr>
<td><strong>Axioms</strong></td>
<td>Yes (Horn rules)</td>
<td>No</td>
<td>No</td>
<td>Yes (Description logic)</td>
</tr>
<tr>
<td><strong>Reasoning</strong></td>
<td>Exposé (Heflin &amp; Hendler, 2000)</td>
<td>No</td>
<td>Graph Matching</td>
<td>Subsumption, Equivalence and Consistency</td>
</tr>
</tbody>
</table>

2.3 Ontology Engineering Methodologies

Ontology engineering methodologies deal with the formalisation process of building ontologies. Here these methodologies are differentiated based on the level of stack holders’ involvement in building ontologies.

2.3.1 Traditional Methodologies

**Gruninger and Fox’s methodology** (Gruninger & Fox, 1995) is based on experience in developing the ontology in the TOVE (Toronto Virtual Enterprise) project. The developed ontologies using this methodology, e.g. scheduling ontology and enterprise design ontology, employ first order logic for constituting their integrated model (set of ontologies to support enterprise modelling). This methodology consists of six proposed steps:
- capture of motivating scenarios;
- formulation of informal competency questions;
- specification of the terminology of the ontology within a formal language;
- formulation of formal competency questions using the terminology of the ontology;
• specification of axioms and definitions for the terms in the ontology within the formal language;
• establishing conditions for characterizing the completeness of the ontology.

The methodology emphasizes the importance of the competency questions during the whole development process. It uses the motivating scenario to define the scope corresponding to the first step of our methodology.

**Uschold and King’s methodology** (Uschold & King, 1995) is based on experience in developing the Enterprise Ontology (including terms and definitions relevant to business enterprise). This methodology consists of four steps:
• identifying purpose;
• building the ontology;
• evaluation;
• documentation.

Uschold and Kind’s methodology suggests using the potential users to define the scope, which is relevant to our case.

**METHONTOLOGY** (Fernandez-Lopez, 1999) is an ontological methodology based on the IEEE 1074-1995 standard describing software development processes. Several large scale ontologies have been developed following this methodology e.g. chemical ontology, environmental pollutant ontologies, reference-ontology. METHONTOLOGY is used in large ontology development with involvement of domain experts and ontologists for long time periods. Three categories of activities are distinguished:
• project management activities including planning, control and quality assurance;
• development-oriented activities including specification, conceptualization, formalization, implementation and maintenance;
• support activities including knowledge acquisition, evaluation, integration, documentation and configuration management.

**On-To-Knowledge** (Fensel et al, 2003) was developed for creating an ontology for knowledge management applications in enterprises. On-To-Knowledge includes the following stages:
• Kick-off - requirements capturing and collection of knowledge sources;
• Refinement – extraction and formalization of knowledge;
• Evaluation – evaluating and validating the ontology; and
• Application and Evolution – utilising the ontology in the intended application and further tuning and refinement.

The **NEON ontology network life cycle model** (Gómez-Pérez et al, 2010) makes an analogy between ontology development and software engineering. NEON proposes two different life cycle models:
• *waterfall model* – applicable when the ontology requirements are clearly definable at the outset and stay fairly stable throughout. There are five variations of waterfall models, which consider sequential order of different tasks in requirements gathering, knowledge acquisition and coding, application and evaluation.
• *iterative-incremental model* – applicable when requirements are being refined in the course of ontology engineering. Ontology construction is considered an iterative process conducted in incremental stages each including clarifying the requirements, ontology expansion and tuning, application and evaluation.
NEON provides extensive guidelines, accompanied by a support toolkit\(^2\), to conduct each step during the ontology lifecycle, e.g. initiation, reuse, merge, re-engineering, design, implementation, maintenance.

The iterative-incremental model fits with the iterative development methodology adopted in the Dicode service development. The supportive ontologies for these services also need to be iteratively developed and revised to fulfil the changes and improvements in the services.

### 2.3.2 Methodologies to Actively Involve Domain Experts

Dicode considers the involvement of stakeholders (domain experts, service developers, end users) in the composition and validation of DON. This is inspired by two methodologies which suggest possible ways to achieve the involvement of domain experts in ontology engineering.

The DynamOnt project (Gruber et al, 2006) pointed out that existing methodologies did not support domain experts because they lacked appropriate support for communities and collaboration. The project aimed at producing a methodology where domain experts could create lightweight ontological models that could be used as part of an "evolving conceptual model". To achieve this, DynamOnt proposed to reuse existing methodologies while:
- adding guidance for domain experts so they can act as knowledge engineers;
- encouraging collaboration;
- grounding the resultant ontologies using foundational ontologies such as DOLCE (Gangemi et al, 2002).

Similar to DynamOnt, the HCOME methodology (Kotis & Vouros, 2006) argued that traditional ontology construction methodologies such as METHONTOLOGY, Uschold & King, and On-to-Knowledge, rely too much on the knowledge engineer for development, maintenance and evolution of ontologies and minimize the role of the domain experts. The HCOME methodology proposes to support individual domain experts by enabling them to collaborate in the construction of ontologies with a community of knowledge workers.

DynamOnt and HCOME have both pointed out limitations of traditional ontology construction methodologies and suggested to involve domain experts by:
- considering ontology construction as a joint process involving both domain experts and knowledge engineers; and providing domain experts with suitable guidance to ensure their active involvement in ontology authoring.

The UK Mapping Agency, Ordnance Survey, developed the Kanga methodology (Kovac et al, 2006) for ontology construction, derived empirically, based on experiences at Ordnance Survey when building several ontologies in the topographical domain. Kanga distinguishes two main aspects in ontological development, namely the conceptual aspect and the computational or logical aspect. The methodology assigns a set of tasks and guidelines for building a conceptual domain ontology (domain ontology defined in terms of conceptual aspects specified and used by the domain expert) with examples from Ordnance Survey’s Hydrology Ontology. This conceptual ontology is an organised way of representing domain knowledge and it is written in structured English sentences or controlled natural language. The logical ontology represents the domain knowledge in a suitable formal language intended for machine use. The conceptual ontology is translated into a logical ontology by an ontology engineer. Kanga includes five steps:

\(^2\) http://neon-toolkit.org/wiki/Main\_Page
• identifying the purpose, scope and other requirements of the ontology;
• gathering source knowledge and documents;
• capturing ontology content in a knowledge glossary;
• writing the glossary content in structured English sentences;
• evaluating and validating the ontology and documentation, which domain experts need to follow to complete the conceptual domain ontology.

Kanga adds to the existing ontology methodologies which focus on domain experts’ involvement by clearly identifying the assumptions about domain experts, distinguishing the phases where domain experts or knowledge engineers should be involved, and clarifying domain experts' role in each phase. Importantly, Kanga does not sacrifice the expressivity of the resultant ontologies and describes how domain experts can be involved in the construction of highly expressive and interconnected ontologies by using a Controlled Natural Language interface.

The Dicode’s ontology framework will provide means for engaging stakeholders in ontology construction and use.

2.3.3 Tools for Intuitive Ontology Engineering

In practice, it is very difficult to achieve effective stakeholders’ involvement without appropriate tool support for intuitive ontology engineering. The available ontology construction tools are reviewed below.

Collaboration Ontology Engineering tools support a community of people (including domain experts and various stakeholders) to build ontologies, e.g. HCOME (Kotis & Vouros, 2006) and Web Protégé (Tudorache et al, 2008). These tools provide communication and Web 2.0 techniques, e.g. discussion forums, to aid users to propose, document and implement changes to the ontology. The main advantage of this approach is that it encourages ages the formation of a community of both domain experts and knowledge engineers to collaborate in building the ontology. These tools improve the communication between domain experts and knowledge engineers, which may motivate domain experts to provide more input into the ontology construction process. However, the means to edit the ontology are similar to traditional tools, e.g. Protégé, which makes domain experts heavily dependent on knowledge engineers to formalize the ontologies. The ontology constructs are actually composed by a group of knowledge engineers (who may or may not be domain experts), while the domain experts without knowledge engineering experience mainly provide the knowledge sources and are involved in the verification of the ontology.

Semantic Wikis (Ghidini et al, 2008) are extensions allowing the wiki manager to define a broad ontology structure that corresponds to wiki pages. Users then refine the ontology by editing and semantically tagging wiki pages. The wiki interface hides the ontology formalisms from the users, in this case domain experts, who can add information to the ontology model by editing wiki pages. Note that to make the interaction intuitive, an initial ontology needs to be created with input from both domain experts and knowledge engineers (e.g. to create semantic forms in Semantic Media Wiki). Semantic wikis offer a flexible approach for lightweight ontology engineering. However, they are inappropriate for heavy weight ontology engineering which requires more expressive logical formalisms, such as description logic and OWL.
Ontology Maturing (Braun et al, 2008; Schmidt, 2008) aims to reuse semi-structured data produced by knowledge workers such as emails, tags and existing schemas and classifications to produce lightweight ontologies and eventually heavyweight ontologies. This approach looks at ways for users to add formal semantics to existing data one layer at a time. Proposed tools, e.g. SOBOLEO (Braun et al, 2008), for extracting a lightweight ontology based on a set of tags or existing schema provide intuitive ways for domain experts to encode their knowledge of the existing data. However, this work is still in progress and more research is required to allow domain experts to insert more complex relations following the maturing approach. Further experimental studies are needed to examine how people without knowledge engineering skills contribute to the ontology maturing process.

Ontology Understanding aims to make it easier to understand what type of knowledge is represented by a particular ontology. For example, by extracting the main concepts in an ontology (Peroni et al, 2008) or by showing relevant metadata (Hartmann et al, 2005). Ontology Visualizations are also commonly used to show and gain insights into the structure of ontologies and linked data (Noppens & Liebig, 2008) and to provide visual interfaces for editing ontologies (Kozaki et al, 2002). Domain experts can benefit from these approaches by getting a high-level understanding of existing ontologies that they can reuse or extend. However, the reuse of extension of the ontologies requires domain experts to be able to understand and edit the ontologies at the axiom level, which requires logical modeling skills lacking in many domain experts.

Recently, the use of Controlled Natural Language (CNL) interfaces (Funk et al, 2007) to perform ontology engineering has been explored. A controlled natural language is a subset of natural language that can be accurately and efficiently processed by a computer, but is expressive enough to allow natural usage by non-specialists (Fuchs & Schwitter, 1995). Example CNLs tools that can be used to view, create and edit ontologies are: CLOnE (Funk et al, 2007), SOS (Cregan et al, 2007), ACE View (Kaljurand & Fuchs, 2006), and ROO (Deanux et al, 2010).

ROO overcomes key usability limitations of existing tools by integrating several features:

- a simple version of look ahead to provide suggestions by guessing what constructs the users might enter;
- show the parsed structure to help the user recognize correct sentence patterns;
- provide a flexible way to parse English sentences using robust language technologies;
- automatically translate to OWL;
- use templates to facilitate the knowledge entering process; maintain a text-based glossary describing parsed concepts and relationships; and
- distribute the CNL tool as a Protégé plug-in.

Table 4 compares ROO to existing ontology engineering tools.
Table 4. Comparison of ROO and existing ontology engineering tools (Denaux et al., 2011).

<table>
<thead>
<tr>
<th>Ontology Engineering Tools</th>
<th>Traditional</th>
<th>Collaborative</th>
<th>Semantic Wiki</th>
<th>Ontology Maturing</th>
<th>CNL-based ROO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ontology Engineering</td>
<td>Yes</td>
<td>Yes [at least one No (but initial team member)]</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>OWL knowledge required</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Stages requiring Knowledge</td>
<td>All</td>
<td>All</td>
<td>Creation of initial ontology</td>
<td>Formalisation (only for heavyweight ontologies) All stages (for Validation and Guidance)</td>
<td>Validation</td>
</tr>
<tr>
<td>Supported Ontology Expressivity</td>
<td>All</td>
<td>All</td>
<td>Lightweight</td>
<td>All, but currently more suitable for lightweight ontologies</td>
<td>All</td>
</tr>
</tbody>
</table>

Learning curve for domain experts | Very high | High | Low | Low | Low | Low | Low |

The Dicode ontology framework will utilise ROO for ontology construction and will provide further tools for facilitating ontology authoring and use.

2.4 Services and Semantics

2.4.1 Introduction

Dicode functionality will be provided in terms of services that can be combined to achieve complex decision making and collaboration tasks. The Dicode Ontological Framework intends to support Dicode services integration by enabling semantic interoperability, as well as facilitating the application and reuse of services by semantic annotations. This section reviews semantic concepts and technologies related to services and semantics.

2.4.2 Service Oriented Architectures

The OASIS Reference Model defines Service Oriented Architecture (SOA) as a “paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains”. The person or organization offering a capability is called a Service Provider and the entity having a need to be solved by one or more capabilities is called a Service Consumer.

The key concepts attached to a Service are:

- **Visibility**: Exposure of capabilities to be found by an entity with needs, typically done through Service Descriptions
- **Interaction**: The activity of using a capability, typically through the exchange of Messages between consumer and provider in a particular Execution Context
- **(Real World) Effect**: Purpose of using a capability, result of an interaction

Visibility presumes that:

- Potential service consumers know the existence of the service. This is known as Awareness.
- The service consumer and service provider engage to interact with each other, according to the service policies.
- There is a communication path for the exchange of messages between service consumer and provider, which means that the service must be reachable for the consumer. A
service is reachable through the exposed and available service endpoint, specified in the service descriptions.

- **Service Descriptions** (see Figure 2) contains the necessary information to facilitate visibility and interaction with a service in terms of inputs, outputs, associated semantics, as well as conditions for using the service. They allow potential service consumers to decide if the service is suitable for their needs and if their context satisfies the requirements of the service provider. A service description comprises service functionalities, the data and metadata models, the policies and contracts related to the services. The service functionalities (operations) and the related (meta-) data models are specified through the service interfaces.

![Figure 2. OASIS-SOA-RA class diagram of a service description](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm)

### 2.4.3 Service Description Languages

Service description depends on the underlying architectural model. The W3C specification for Web Service Architectures identifies four architectural models:

- The service oriented model, where the action is the key concept;
- The resource oriented model, where the resource is the key concept;
- The message oriented model, where the message is the key concept;
- The policy model, where policy is the key concept.

**WADL – Web Application Description Language**

[http://www.w3.org/Submission/wadl/](http://www.w3.org/Submission/wadl/)

The Web Application Description Language (WADL) is a description framework for RESTful web services (i.e. resource oriented web services) based on XML. WADL models the resources provided by a service and the relationships between them using a set of resource elements. Each of these contains parameter elements to describe the inputs and method elements that describe the request and response of a resource. The request element specifies how to represent the input and what types and specific HTTP headers are required. The availability of the resource is indicated by the resource element.

---


4. [http://www.w3.org/TR/ws-arch/](http://www.w3.org/TR/ws-arch/)
response describes the representation of the service’s response as well as any fault information to deal with errors. WADL is a W3C member submission.

**WSDL – Web Service Description Language**  
[http://www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl) (draft)

The Web Service Description Language (WSDL) is an XML format for describing network services (as a service oriented model). WSDL defines services as collections of network endpoints, or ports. The abstract definition of ports and messages is separated from their concrete use or instance, allowing the reuse of these definitions. A port is defined by associating a network address with a reusable binding and a collection of ports defines a service. Messages are abstract descriptions of the data being exchanged, and port types are abstract collections of supported operations. The concrete protocol and data format specifications for a particular port type constitute a reusable binding, where the operations and messages are then bound to a concrete network protocol and message format. In this way, WSDL describes the public interface to the web service.

WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to call one of the functions listed in the WSDL.

Using WSDL, resources can be exposed by both the Web Services Interoperability (WS-I Basic Profile) and the WSRF framework.

**WSRF – Web Services Resource Framework**  

The Web Services Resource Framework (WSRF) is a family of OASIS-published specifications for web services. Major contributors include the Globus Alliance and IBM.

A web service by itself is nominally stateless, i.e. it retains no data between invocations. This limits the things that can be done with web services, although workarounds exist - such as having the web service read from a database, for example, or using session state by way of cookies or WS-Session.

WSRF provides a set of operations that web services may implement to become stateful; web service clients communicate with resource services that allow data to be stored and retrieved. When clients talk to the web service they include the identifier of the specific resource that should be used inside the request, encapsulated within the WS-Addressing endpoint reference. This may be a simple URI address or a complex XML content that helps to identify or even fully to describe the specific resource in question.

Alongside the notion of an explicit resource reference comes a standardised set of web service operations to get/set resource properties. These can be used to read and perhaps write resource state, in a manner somewhat similar to having member variables of an object alongside its methods. The primary beneficiaries of such a model are management tools, which can enumerate and view resources, even if they have no other knowledge of them. This is the basis for WSDM.
SSDL – SOAP Service Description Language
http://www.ssdl.org/

The SOAP Service Description Language (SSDL) is a SOAP-centric description language for web services. It promotes message-orientation as the architectural paradigm for building distributed applications and enables protocol-based integration between web services. The SSDL effort is an attempt to explore ideas in the areas of contract and protocol description as well as web services implementations using message-oriented programming abstractions.

2.4.4 Semantic Technologies for Services

The service description languages suffer from being purely syntactical. Though some meaning may be embedded into naming schemes, operations with the same name may differ substantially. To overcome this shortcoming, embedding semantic concepts enhances service descriptions. These efforts focused on metadata descriptions are centred on the Semantic Web and its technologies, namely RDF and OWL. In context of services, semantic support is envisaged for:

- Mediation
- Discovery
- Integration
- Checking consistency

The following are relevant technologies and standards:

**MicroWSMO – Semantic Description of RESTful Web Services**
http://wsmo.org/TR/d38/v0.1/20080219/d38v01_20080219.pdf

MicroWSMO is a service ontology and a way of annotating RESTful web services with this ontology in order to achieve Semantic Web Service automation.

**SAWSDL – Semantic Annotations for WSDL and XML Schema**
http://www.w3.org/TR/sawsdl/

Recommendation of W3C. Semantic Annotations for WSDL and XML Schema (SAWSDL) defines how to add semantic annotations to various parts of a WSDL document such as input and output message structures, interfaces and operations.

SAWSDL distinguishes two kinds of annotations:

- The “modelReference” that generically references to semantic concepts acting as a hook for attaching semantics
- The “liftingSchemaMapping” and “loweringSchemaMapping” that map XML entities (elements and types) to corresponding concepts in, e.g., RDF and vice versa (see Figure 3). Schema mapping typically uses XSLT\(^5\) and SPARQL or XQuery\(^6\) and it is often compositional in that it preserves (parts of) the structure of XML entities.

\(^5\) http://www.w3.org/TR/1999/REC-xslt-19991116
\(^6\) http://www.w3.org/TR/xquery/
WSDL-S – Web Service Semantics  
http://www.w3.org/Submission/WSDL-S/

WSDL-S is a means to add semantics inline to WSDL. It is actually an extension to WSDL 2.0 but can be used for WSDL 1.1. According to WSDL-S inputs and outputs of WSDL operations are annotated with domain concepts while the operations themselves are annotated with preconditions and effects (post conditions). Also the service’s interface is annotated with category information that could be used while publishing services in registries such as UDDI. The semantic domain model used is external to these annotations and could be expressed in OWL or other ontology language of choice.

OWL-S - Semantic Markup for Web Services  
http://www.w3.org/Submission/OWL-S/

OWL-S (formerly DAML-S) builds on top of OWL and allows for the description of a Web service in terms of a Profile, which tells "what the service does/provides", a Process Model, which tells "how the service works", and a Grounding, which tells "how to access the service" [owl-s]. The service profile describes what is accomplished by the service, any limitations on service applicability and quality of service, and requirements that the service requester must satisfy in order to use the service successfully. The process model gives details about the semantic content of requests, the conditions under which particular outcomes will occur, and, where necessary, the step by step processes leading to those outcomes. In the process model a service can be described as an atomic process that can be executed in a single step or a composite process that, similar to a workflow, can be decomposed in other processes based on control structures like ‘if-then-else’ and ‘repeat-while’. Finally, Grounding descriptions supply information about the communication protocol and other transport information (such as port numbers) and the message formats and serialization methods used in contacting the service. The only currently specified grounding mechanism is based on WSDL 1.1 and will be extended to WSDL 2.0 as soon as it’s finalized.

---

OWL-WS – OWL for Workflows and Services

OWL for Workflows and Services (OWL-WS) is a workflow and service ontology supported by the NextGRID project. It is based on OWL-S with various extensions:

- Provision of abstract definition of workflows so that a task’s bindings (endpoints, etc.) can be specified at run time;
- Support for substitution of an abstract workflow with a concrete one at run time;
- Support for higher order workflows so that a task can have a workflow as input and return another workflow as output.

SWRL – Semantic Web Rule Language
http://www.w3.org/Submission/SWRL/

The Semantic Web Rule Language (SWRL) attempts to combine OWL with Horn clauses\(^8\). SWRL is not decidable. SWRL is decidable if the rules are \(DL\)-safe, i.e. each variable must occur in a context different to an OWL class or property. SWRL is not yet a standard.

SWSF - Semantic Web Services Framework
http://www.w3.org/Submission/SWSF/

The Semantic Web Services Framework (SWSF), initiated by the Semantic Web Services Initiative, includes the Semantic Web Services Language (SWSL) and the Semantic Web Services Ontology (SWSO). SWSL is a logic-based language for specifying formal characterizations of Web service concepts and descriptions of individual services. SWSO is an ontology of service concepts defined using SWSL and incorporates a formal characterization (“axiomatisation”) of these concepts in first-order logic.

WSMO (Web Services Modelling Ontology)
http://www.wsmo.org/

The Web Services Modelling Ontology defines the modelling elements for describing several aspects of Semantic Web services. These elements are Ontologies, which provide the formal semantics to the information used by all other elements, Goals which specify objectives that a client might have when consulting a Web service, Web services that represent the functional and behavioural aspects which must be semantically described in order to allow semi-automated use, and Mediators that are used as connectors and they provide interoperability facilities among the other elements. It also defines the Web Service Modelling Language (WSML) that formalizes WSMO and aims to provide a rule-based language for the Semantic Web.

WSML – Web Service Modeling Language
http://www.wsmo.org/wsml/

The Web Service Modeling Language (WSML) formalises the Web Service Modelling Ontology (WSMO). It has a twofold mission in:

- Developing a proper formalisation language for Semantic Web Services;
- Providing a rule-based language for the Semantic Web.

\(^8\) A Horn clause is a disjunction of atomic propositions with at most one positive literal.
2.4.5 Semantic-enhanced Service Coordination

According to (Barros et al, 2006), service coordination (or service composition) can be considered from different viewpoints:

- **Behavioural Interface** – captures the dependencies between interactions as provided or expected from the perspective of a particular service.
- **Choreography** – refers to the coordination of multiple services by message exchange from a global perspective.
- **Orchestration** – refers to the typically hierarchically organised coordination of interaction of several services and of internal actions controlled by a single service. Orchestrations may be presented in terms of formally defined workflows that are executed by a workflow enactor.

The viewpoints relate to each other in that an orchestration or choreography defines an expected behaviour interface of the services involved and, vice versa, the interfaces provided specify boundary conditions for orchestrations and choreographies.

Semantic enhancements support service coordination in several ways:

- **Resource Discovery**: Semantic annotations support resource/service discovery and proofing whether a resource/service is suitable for a particular purpose.
- **Compatibility**: Compatibility of services can be established not only on syntactic but also on semantic level.
- **Validity**: Semantic validity of the results of a service coordination can be checked by humans or automatically by machines. Factors to be considered in the semantic evaluation of a chain result are listed in ISO 19119:
  - Appropriateness of starting data: are the based datasets suited to the subsequent processing? For example, accuracy, and resolution of the data, thematic values are relevant.
  - Affect of services on data: how do the individual services affect the data, e.g., error sources and propagation.
  - Sequence of the services: how does the order of the chain affect the results? For example, should a spatial operation, e.g., ortho-rectification, be performed before or after a thematic operation, e.g., re-sampling the attribute values?
- **Mediation**: The coordination of services implies that the services used are aligned. Mediation is an interoperability mechanism outside the service components providing semantic matching of service capabilities. Semantic matching is needed when two services to be chained have incompatible exchange information models or when the service visibility elements should be matched to a particular request. Mediation services may be manually or automatically inserted into a workflow to align the input and output of closely related data or operation name, when necessary. They are also known as adaptor, façade, shim services, or simple shims. Relevant elements for mediation are:
  - A semantic enhanced description of services to allow semantic matching (Paolucci et al, 2002; Hull et al, 2005).
  - A matching vocabulary to describe relationships between services and matching degrees (Hull, 2008).
  - Definition and specification of different mediator service types to solve the matching according to different contexts. The context may be determined either through particular information models or through considering different matching
strategies, like translation, dereferenciation, comparison, parsing, etc. (Hull et al, 2004).

- **Workflow Matching:** Semantic mediation is not just needed for basic resource units as are data and services, but also to support finding and using composed resources, such as workflows. (Gil et al 2009) state that scientists discover workflows by given properties of workflow data inputs, intermediate data products, and data results. They provide an approach to workflow-matching based on:
  - An algorithm for enriching workflow catalogues with relevant properties of data and components
  - An algorithm for workflow-matching based on data-centred properties
  - Separation of reasoning steps to be called outside the data catalogue

Workflows matching introduce a new data type to common matching problems, since workflows are graphs. The (Punko & Rueping, 2009) approach is based on graph mining techniques, in particular those for graph kernels.

### 2.4.6 Conclusions

At the present stage, there is no established or widely accepted semantic modelling concept for services. However, the acceptance of SAWSDL seems to be rising due to the fact that the underlying WSDL is widely accepted for syntactic description of services. Hence Dicode will experiment with SAWSDL and evaluate its incorporation into the Dicode Ontological Framework. A problem is that SAWSDL relates to SOAP as service technology while presently a move to REST based services can be observed. MicroWSMO might be an alternative here or micro formats such as hREST. WSMO in general appears to be an elegant modelling framework for (web) services but it must be seen whether it integrates into the Dicode Ontological Framework.

---

9 [http://knoesis.wright.edu/research/srl/projects/hRESTs/](http://knoesis.wright.edu/research/srl/projects/hRESTs/)
3 Available Ontologies for Use-case Specific Layers in DON

This section will review ontologies that are relevant for the use cases, aspects of which can be used for constructing the Dicode ontology.

3.1 Medical Ontologies Relevant to Use Cases 1 and 2

Medical ontologies are developed to solve problems such as the demand for the reusing and sharing of patient data, the transmission of these data, or the need of semantic-based criteria for statistical purposes (Gomez-Perez et al, 2004). Some of the Ontology languages described before have been used in real biomedical environments and some are still being used. In this section some practical and real examples will be explained and readers could notice how these languages are very useful in the biomedical domain.

Medical Subject Headings (MeSH) is a taxonomy of medical terms used by the National Library of Medicine (NLM) for its bibliographies and cataloging (Sewell, 1964). This taxonomy or very light ontology is described using XML and can be downloaded from http://www.nlm.nih.gov/mesh/filelist.html.

During the epidemic of “Mad Cow Disease” (Transmissible Spongiform Encephalopathy, TSE) the World Wide Web published much information about it: from rumors and legends to valuable information web sites. Users found difficult to look for reliable information using the common search engines, because they perform their searches using word matching rather than by the semantic content of the search. Heflin, J., Hendler, J. and Luke, S. built a TSEs web site that facilitated the access by agents in order to enable the semantic-search about these diseases (Applying ontology to the Web: A case study, 1999). This web site was supported by a SHOE ontology and can be downloaded from http://www.cs.umd.edu/projects/plus/SHOE/onts/tseont.html.

Real Applications in XOL are not available because it was more a theoretical issue than a practical tool. As has been explained above XOL is a redefinition of some OKBC primitives into XML. It evolved very quick and OIL and DAM+OIL can be considered as an evolution of it. Real and practical DAM+OIL applications are widely available in Internet (Wroe et al, 2003; Pérez-Rey et al, 2005). However, XOL was, as the Molecular Biology Ontology Working Group stated, one of the first XML-based syntax defined to be used as a bioinformatics ontology exchange language to increase the likelihood that the language will see widespread acceptance (McEntire et al, 2000).

Dbpedia (http://dbpedia.org/) instances; an effort to extract structured information from Wikipedia and to make this information (Biomedical information too) available on the Web; is one of the most popular applications of RDFS. Other applications are, for example, the Lex-RDF model, that is an alignment, to RDFS W3C recommendation, of the LexGrid project which provides a common terminology model to represent multiple vocabulary (ICD-9-CM, the Gene Ontology, the HL7 Version 3 vocabulary, and SNOMED-CT) and ontology sources as well as a scalable and robust API for accessing such information. Finally, Bio2RDF proposes an approach for building mashups of bioinformatics in order to help the process of bioinformatics knowledge integration (Belleau et al, 2008).
An upper ontology, top-level ontology or foundation ontology is an ontology which describes very general concepts that are the same across all knowledge domains. For example, the Basic Formal Ontology (BFO, http://www.ifomis.org/bfo) is a set of ontologies which support ontology development for scientific research domain. BFO gives different series of perspectives on reality: An inventory of all entities existing at a time (SNAP perspective) and an inventory of all processes unfolding through time (SPAN perspective). In the biomedical domain it is applied as anatomy and physiology perspectives respectively. This ontology can be found implemented in OWL by Holger Stenzhorn et al. in http://www.ifomis.org/bfo/1.1#.

3.1.1 Biomedical ontologies

In this section, we provide some examples of biomedical ontologies already developed and available through the Internet.

SNO MED

The Systematized Nomenclature of Medicine (SNO MED), designed by the Nomenclature Committee of the College of American Pathologists, is a multi-axial coded nomenclature composed mainly in its origins by six main axes (Architecture of SNO MED: Its Contribution to Medical Language Processing, Cote, R.A., 1986). Next, are introduced that six axes, the first four axes contain all the terms needed to arrive at a general or specific disease diagnosis:

- Topography: This list is composed of parts of the human mind or body (Anatomy) as heart or appendix.
- Morphology: This list contains the abnormalities that affect the body (Pathologic Anatomy) as appendicitis or acute inflammation.
- Function: This list is formed by signs and symptoms as well as functions and physiological functional units as lung volumes and capacities or ions, enzymes and hormones.
- Etiology: This list contains terms related with the origin of diseases as bacteria or Streptococcus.
- Disease: This is a list, knowingly incomplete, of complex disease entities and syndromes as, for example, Crohn's disease or hemophilia.
- Procedure: Once the diagnosis is made or even to arrive at a more precise diagnosis whether simple or complex, the health care team usually takes some action. This is a list of possible action to perform as, for example, Gastrectomy.

The SNO MED RT, the reference technology, complements the SNO MED terms with a set of concepts to facilitate comparison and aggregation of data about the entire health-care process (Spackman et al, 1997). The main purpose of a reference terminology for clinical data is the retrieval and analysis of data relating to the causes of disease, the treatment of patients, and the outcomes of the overall health care process.

Finally, SNO MED Clinical Terms (SNO MED CT), as the web site (http://www.nlm.nih.gov/research/umls/Snomed/snomed_faq.html) explains, is an extensive clinical terminology that was formed by the merger, expansion, and restructuring of SNO MED RT® (Reference Terminology) and the United Kingdom National Health Service (NHS) Clinical Terms (also known as the Read Codes). It is the most comprehensive clinical vocabulary available in English (or any language). SNO MED CT is concept-oriented and has an advanced structure that meets most accepted criteria for a well-formed, machine-
readable terminology. It has been designated as a US standard for electronic health information exchange in Interoperability Specifications produced by the Health-care Information Technology Standards Panel and has also been adopted for use by the U.S. Federal Government, through the Consolidated Health Informatics (CHI) Initiative, for several clinical domains.

NCBI Organismal Classification
Taking into account the ranking of most viewed ontologies from Bioportal (http://bioportal.bioontology.org/), one of the most important ontology repositories at the date of this document, SNOMED CT Ontology is the most viewed and then, NCBI Organismal Classification.

This ontology is composed of five father branches: Cellular Organisms, Other Sequences, Unclassified Sequences, Viroids and Viruses. Therefore, the NCBI Organismal Classification is a taxonomic classification of living organisms and associated artifacts for their controlled description within the context of databases (http://bioportal.bioontology.org/ontologies/38802). This ontology belongs to the National Center for Biotechnology Informations.

ACGT Master Ontology on Cancer
ACGT is a European Union co-funded project aiming at developing open-source, semantic and grid-based technologies in support of post genomic clinical trials in cancer research. It addresses clinicians, bio-researchers as well as software developers' needs, providing an open platform where novel and powerful services can be offered and used by practitioners in the field (http://eu-acgt.org/).

During this project one of the elements produced was the ACGT Master Ontology on Cancer (Brochhausen et al, 2010). This new terminology source for oncological practice has for objective to develop a semantic grid infrastructure in support of multi-centric, post-genomic clinical trials, and thus enable the smooth and prompt transferring of laboratory findings to the clinical management and treatment of patients.

The amount of data on cancers and their treatment has exploded over the past years, due to advances in research methods and technologies. Recent research results have changed our understanding of fundamental aspects of cancer development at the molecular level. Data mining in a multi-dimensional representation is already a real need within this domain.

Gene Ontology
The Gene Ontology (GO) Consortium, as the official web site says (http://www.geneontology.org/GO.consortiumlist.shtml), is the set of model organism and protein databases and biological research communities actively involved in the development and application of the Gene Ontology. It is composed of different entities as the Berkeley Bioinformatics and Ontology Project (BBOP - http://www.berkeleybop.org) or the British Heart Foundation - University College London (BHF-UCL, http://www.geneontology.org/GO.consortiumlist.shtml) among others.

The Gene Ontology project is a major bioinformatics initiative with the aim of standardizing the representation of gene and gene product attributes across species and databases. Its goal is to reduce the time that the biologists are wasting looking for all of the available information about each small area of research. Moreover, this domain is characterized by the wide variations in terminology. For example, if a biologist is searching for new targets for
antibiotics, he might want to find all the gene products that are involved in bacterial protein synthesis, and that have significantly different sequences or structures from those in humans. If one database describes these molecules as being involved in 'translation', whereas another uses the phrase 'protein synthesis', it will be difficult for biologists and even harder for a computer to find functionally equivalent terms (http://www.geneontology.org/GO.doc.shtml).

The goal of the Gene Ontology Consortium is to produce a dynamic, controlled vocabulary that can be applied to all eukaryotes even as knowledge of gene and protein roles in cells is accumulating and changing. To this end, three independent ontologies accessible on the World-Wide Web are being constructed: biological process, molecular function and cellular component (Ashburner et al, 2000):

- Biological Process Ontology: A biological process is series of events accomplished by one or more ordered assemblies of molecular functions as cellular physiological process or pyrimidine metabolic process.
- Molecular Function Ontology: Elemental activities, such as catalysis or binding, describing the actions of a gene product at the molecular level. A given gene product may exhibit one or more molecular functions.
- Cellular Component Ontology: The part of a cell or its extracellular environment in which a gene product is located. A gene product may be located in one or more parts of a cell and its location may be as specific as a particular macromolecular complex, that is, a stable, persistent association of macromolecules that function together. This may be an anatomical structure as rough endoplasmic reticulum or nucleus, or a gene product group as ribosome, proteasome or a protein dimer.

Much information about this group of ontologies can be found either its official web site or other biomedical ontologies repositories as Bioportal. GO is revolutionizing the research activity in the biological domain and it is having much impact within it, as statistical data, the GO official web site links, at the date of this document, 3693 publications related with the Gene Ontology.

**Unified Medical Language System**

In 1986, the National Library of Medicine began a long-term research and development project to build the Unified Medical Language System (UMLS). The purpose of the UMLS is to improve the ability of computer programs to understand the biomedical meaning in user inquiries and to use this understanding to retrieve and integrate relevant machine-readable information for users (Lindberg et al, 1993).

UMLS is not an ontology itself unlike SNOMED, NCBI Organismal Classification, ACGT Master Ontology on Cancer or the Gene Ontology showed here. It is formed by three UMLS Knowledge Sources: the Metathesaurus, the Semantic Network, and the SPECIALIST Lexicon (http://www.nlm.nih.gov/research/umls/about_umls.html). This knowledge sources are thought to provide a semantic information load in order to facilitate the development of intelligent interfaces to biomedical information systems.

Metathesaurus and Semantic Network formalisms aim to help interfaces to link or map user queries to biomedical data. On the other hand the SPECIALIST Lexicon has been developed to provide the lexical information needed for the SPECIALIST Natural Language Processing
System (NLP). This Lexicon assists systems in the identification of the most appropriate information source or sources for the user query posed.

UMLS, as well as the three knowledge sources before viewed, contains a set of software supporting tools, for example, the MetamorphoSys which customizes the Metathesaurus for specific applications, for instance by excluding certain source vocabularies; MetaMap which retrieves the Metathesaurus concepts from an input text; and others.

**Health Behavioral Change Ontology**

One-on-one, face-to-face interaction with a health provider is widely acknowledged to be the "gold standard" for providing health education to and affecting health behavior change in patients and consumers. Automated health dialog systems — especially those which use speech and other audiovisual media — emulate this form of interaction to communicate health information to users in a format that is natural, intuitive and dynamically tailored (Bickmore & Giorgino, 2006).

Exercise, smoking cessation, diet, medication, drinking decrease, etc. are health behavior changes which have a very important impact within the society. Bickmore, T. et al. propose a framework for simulating a human health counselor who helps patients to change their health behavior through several conversation over time (Bickmore et al, 2011).

A Counseling Framework is essentially a collection of patient mental states and counselor therapeutic actions that can modify those states. This model is implemented employing an OWL ontology of health behavior change concepts (HBCO) and a public task modeling language (ANSI/CEA-2018). Basically the Behavioral Medicine Ontology specifies the kind of knowledge that follows:

- **A Theory Model**: Relationships among components of a few health behavior theories, constructs, and counseling techniques
- **A Behavior Model**: Knowledge about how this theories are applied to concrete health behaviors:
  - Whether the intervention is acquisition or cessation oriented
  - Whether the behavior can be incrementally modified or not
  - Recommended homework exercises or tips
  - Words or sentences the dialogue system use in this kind of conversations
- **A Protocol Model**: Knowledge about a particular behavior change intervention as duration of the intervention, last successful behavioral criteria, etc.
- **A User Model**: Knowledge about a concrete patient as gender, age or ethnicity. It could contain even the Electronic Health Record (EHR)
- **External Data Model**: Describes system data inputs and outputs as HER
- **A Task Model**: Knowledge about how the system enacts the intervention. Unlike the other knowledge sources are conceptual by nature, this representation is procedural and specifies which steps or stages the system should execute.
The Health Behavioral Change Ontology specifies the conceptualization of the theory-driven domain of conversational agent-based behavior change intervention. This domain is basically composed by the knowledge sources explained before.

**African Traditional Medicine Ontology**

African Traditional Medicine (ATM) is the result of diverse experience, mixing customs and knowledge about Nature, which has been transmitted by oral tradition along the history (Atemezing & Pavon, 2008). Traditional medicine refers to health practices, approaches, knowledge and beliefs incorporating plant, animal and mineral based medicines, spiritual therapies, manual techniques and exercises, applied singularly or in combination to treat, diagnose and prevent illnesses or maintain well-being (http://bioportal.bioontology.org/visualize/40223).
The development of this ontology is closely related with the motivation of building a formalism, easily extended as required and with the possibility of knowledge structuring and integration with other sources. This ontology development implies several challenges as the completion of ontologies in other domains as the intersection between botanic and Medicine. For example, the role of natural elements in rituals.

The Plant Ontology (The plant structure ontology, a unified vocabulary of anatomy and morphology of a flowering plant, Ilic, K., et al., 2007), The Pathogen Transmission Ontology (http://bioportal.bioontology.org/ontologies/44947), The Human Disease Ontology (http://do-wiki.nubic.northwestern.edu/index.php/Main_Page#Publications) and The Infectious Disease Ontology (Infectious disease ontology, Cowell, L.G., and Smith, B., 2010) are ontology resources used by the ATM Ontology.

Several actors can be identified in the ATM ontology: The Healer, the Fetishist, the Soothsayer and the Magician. Thus, concepts as Actor Functions, Treatment Proposed Processes, Symptoms and Diseases are considered as part of the ATM Ontology. The nine main or more general concepts within the ATM Ontology are:

- Disease Conception: A traditional conception of a disease cause.
- Environment: Geographical environment and site of care.
- Countries.
- Social Context: Such as communities, groups, institutions, occupations, persons, population, religion and philosophy.
- Traditional Action: Traditional actions where the use of magic or malefic power can cause illness.
- Traditional Believes of a given traditional group.
- Traditional Actors such as the explained before.
- Traditional practices included in the African traditional medicine.
- Traditional treatment: The way a treatment is done and the elements that enter it.

The ATM Ontology, has been developed following the same rules as the Gene Ontology Consortium. Therefore, each concept has been identified as ATM:nnnnn where nnnnn is a unique identifier and equivalent terms in other ontologies, as for example the Plant Ontology, are included in ATM Ontology with the identifiers used in that ontologies.

3.1.2 Biomedical ontologies search engines

In this section, we provide some examples of biomedical ontologies search engines already developed and available through the Internet.

Bioportal

Bioportal is an ontology repository and search engine used to access and share ontologies that are actively used in biomedical communities. Is possible to search for terms in ontologies as Melanoma, browse a list of ontologies and search biomedical resources that were automatically annotated with ontology terms. Moreover, Bioportal allows to create ontology-based annotations for personal texts, link personal project that uses ontologies to the description of those ontologies, find and create relations between terms in different ontologies, review and comment on ontologies and their components as users browse them.
BioPortal is an open repository of biomedical ontologies that provides access via Web services and Web browsers to ontologies developed in OWL, RDF, OBO format and Protégé frames. BioPortal functionality includes the ability to browse, search and visualize ontologies (Noy et al, 2009).

Bioportal provides, as it was mentioned before via web services, different information about the ontologies as: The taxonomy; Some term details as synonyms or semantic type; Graphical visualization; Different ontology views; and some ontology widgets to be added in personal web sites among others.

The Open Biological and Biomedical Ontologies: OBO Foundry
The Semantic Web era has enhanced the value of the available data in the World Wide Web, publishing it in a form that allows it to be integrated with other data. The most common approach to do this value enhancement has been using controlled vocabularies described by ontologies. The Open Biomedical Ontologies (OBO) consortium is pursuing a strategy to overcome the problem of overload of ontologies. Existing OBO ontologies, including the Gene Ontology, are undergoing coordinated reform, and new ontologies are being created on the basis of an evolving set of shared principles governing ontology development in OBO Foundry (Smith et al, 2007).

The OBO Foundry is a collaborative experiment involving developers of science-based ontologies who are establishing a set of principles for ontology development with the goal of creating a suite of orthogonal inter-operable reference ontologies in the biomedical domain (http://www.obofoundry.org/).

OBO ontologies are considered candidates if they meet ten principles: Openness and Availability; Common Shared Syntax (OBO and OWL); Unique Identifier Space; Version Control; Delineated Content; Terms Definitions; OBO Relation Ontology; Documented; Plurality of independent Users; and Collaborative Development with another OBO Foundry members (http://www.obofoundry.org/crit.shtml).

Ontology Lookup Service: OLS
The Ontology Lookup Service (OLS)-(http://www.ebi.ac.uk/ontology-lookup/) provides interactive and programmatic interfaces to query, browse and navigate an ever increasing number of biomedical ontologies and controlled vocabularies (Cote et al, 2008).

OLS, unlike BIO Portal and OBO Foundry, is more oriented to lookup terms or concepts belonging to some of the most used biomedical ontologies nowadays as the Gene Ontology (GE), Basic Formal Ontology (BFO) or Symptom Ontology (SYMP) among others.

The Ontology Lookup Service was created to provide a simple, centralized, integrated interface to query multiple biomedical ontologies by interactive and programmatic means, saving like this time to users that need to aggregate data from some different sources.

3.2 Ontologies Relevant to Use Case 3

Product ontology
Product ontology can be used to facilitate opinion mining about a product by identifying ‘hot topics’ about a product (Li & Du, 2011). This work presents an approach for blog mining
to identify the ‘opinion leaders’ – people who have highly interconnected and strong social standing.

**EmotionML**  
([http://www.w3.org/TR/emotionml/](http://www.w3.org/TR/emotionml/))

Sentiment analysis can be facilitated by generic emotion vocabularies. The WWW consortium is currently creating an Emotion Markup Language which will formalise the range of emotions humans experience, and can be used as a main source in affect analysis.

**Wordnet Affect**  
([http://wndomains.fbk.eu/wnaffect.html](http://wndomains.fbk.eu/wnaffect.html))

WordNet Affect extends Wordnet with synsets for representing affective concepts correlated with affective words. A special a-label EMOTION is used to indicate the synsets of affective concepts representing emotional state. Other a-labels are used for representing moods, situations eliciting emotions, or emotional responses.

**SentiWordnet**  
([http://sentiwordnet.isti.cnr.it/](http://sentiwordnet.isti.cnr.it/))

SentiWordnet 3.0 is a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications. It can be applied to automatically determine whether a term that is a marker of opinionated content has a positive or a negative connotation (Esuli & Sebastiani, 2006).

### 3.3 Open Datasets relevant to the use cases

The term “Linked Data” refers to a set of best practices for publishing and connecting structured data on the Web. These best practices have been adopted by an increasing number of data providers over the last three years, leading to the creation of a global data space containing billions of assertions — the Web of Data (Bizer et al, 2009). The Web of Data can be utilised to bootstrap various domain ontologies in the DON framework. For example, highly relevant to the use case B (Rheumatoid Arthritis), there are relevant datasets in the Linked Data Cloud that can provide useful domain specific knowledge. For example, diseasome ([http://www4.wiwiss.fu-berlin.de/diseasome/](http://www4.wiwiss.fu-berlin.de/diseasome/)) dataset contains linked data version of disorders and disease genes linked by known disorder-gene associations for exploring all known phenotype and disease gene associations, indicating the common genetic origin of many diseases. Executing following SPARQL query against the endpoint ([http://www4.wiwiss.fu-berlin.de/diseasome/snorql/](http://www4.wiwiss.fu-berlin.de/diseasome/snorql/)) provided on the diseasome website provides domain knowledge related to the Rheumatoid Arthritis.

```sparql
SELECT DISTINCT * WHERE {
}
```

Other relevant datasets are Drugbank ([http://www4.wiwiss.fu-berlin.de/drugbank/](http://www4.wiwiss.fu-berlin.de/drugbank/)) which contains RDF dataset of drug and clinical trial ([http://linkedct.org/index.html](http://linkedct.org/index.html)) dataset that contains semantic data on clinical trials.

Linked Data cloud is very rich in terms of life science datasets which closely matches to the nature of the first two use cases from the Dicode project.
4 Available Ontologies for the DON Generic Layer

4.1 Introduction

In this section some of the most popular generic collaboration and decision making ontologies are presented. They will be followed when defining the generic layer in the Dicode Ontology. In addition, we point at relevant workflow ontologies, which can be used for defining the service execution semantics in the DON generic layer.

4.2 Collaboration Ontologies

4.2.1 3C-based Collaboration Ontology

Oliveira et al. (2007) propose a collaboration ontology based on the 3C (i.e. communication, coordination and cooperation) model (Ellis et al., 2001). Following this model, the collaboration ontology is divided in three sub-ontologies: Cooperation, Communication and Coordination Ontology (Figure 5).

![Figure 5: The sub-ontologies forming the Collaboration ontology.]

A collaboration session (CS) in this approach is an event representing a period of time during which some agents collaborate with each other for a given purpose. Each CS is related to some important concepts such as the participants (agents contributing to achieve the objectives of a session), the objectives (state of affairs the participants aim at achieving), the artifacts, the coordination and the communication. Cooperation and interaction are subject to the commitments of each of the participants while collaboration involves information sharing among the members of a group.

In this shared workspace (CS), Cooperation is defined as the joint effort to achieve the common goal. Participants have to be familiar with the proper rules and procedures in order to avoid implicit rules and uncontrolled behavior in a CS. This motivates the need for a formal protocol, commonly known as coordination, defined by the Webster dictionary (Merriam, 2007) as the harmonious functioning of parts for effective results. According to Fuks et al. (Fuks et al., 2005), Coordination services support management and enforcement of group activities. In addition, they are necessary to avoid conflicts and repetitive actions and to solve problems that share common objectives, resources and activities (Ellis et al., 1991). Communication is essential to any collaboration system as its objective is to exchange knowledge among individuals (Fuks et al., 2005).

4.2.2 Collaboration Networks Ontology

Rajsiri et al. (2008) approach takes as input knowledge concerning collaboration coming from involved organizations and aims at automating the specification of virtual organization
collaborative processes. Rajsiri et al. focus on the collaborative network domain, especially for designing collaborative processes. The collaborative network ontology (CNO) proposed consists of collaboration attributes, the description of participants and the collaborative processes. There are three key concepts underlying such a network ontology:

- **The participant concept.** It concerns the characterization criteria of the collaboration (Rajsiri et al., 2007). A participant provides high level services and resources (such as the machine, the container, the technology), plays roles (seller, buyer, producer) and has business sectors.

- **The collaboration concept.** It concerns the characterization criteria of the collaboration (Rajsiri et al., 2007) and integrates the collaborative process meta-model (Touzi et al., 2007). The characterization criteria may be the common objective, the resources, the relationships or the topology, while the CIS (Collaborative Information System) and the CIS Services form the collaborative process meta-model.

- **The collaborative process concept.** It is an extension of the concepts developed by the MIT Process Handbook project (Malone et al., 1999).

The three above concepts are grouped together in Figure 6.

![Figure 6](image)

**Figure 6** The participant, collaboration and collaborative process concepts.

### 4.2.3 Collaboration Ontology based on Collaboration Engineering

Knoll et al. (Knoll et al., 2010) start by determining a set of potential user groups for a collaboration ontology. Based on Collaboration Engineering (Kolfschoten and de Vreede, 2009), three user group with different knowledge bases have been identified; the collaborator engineer (designer of collaboration processes, familiar with the factors that affect the outcome of the collaboration process), the practitioner (domain expert familiar with the needed resources, the client goal and the stakes of a collaboration process) and the participant (taking part in the collaboration process and familiar with the client goal). The scope of the ontology has been defined through competency questions (questions for identifying the area of knowledge that the ontology should focus on). The design pattern adopted (thinkLet) defines the actions of a participant with respect to his skills, a defined role and the capabilities (de Vreede et al., 2006). The collaboration ontology proposed by Knoll et al. (2010) is depicted on Figure 7 and its main concern is to describe the collaboration process from an external point of view. The following key concepts were introduced to define the Collaboration Ontology:

- **CollaborationClient:** a person or group of people with the need to collaborate
- **CollaborationObjective**: a goal set by the CollaborationClient to express the desired result
- **CollaborationTask**: a step to achieve a CollaborationObjective in a collaborative manner
- **CollaborationProcess**: the part in which participants interact for the purpose of collaboration; the implementation of a specific CollaborationTask
- **CollaborationResult**: the actual outcome of a Collaboration Process
- **Quality Criterion**: a criterion that results from the Collaborative Objective and against which the Collaborative Result is evaluated.

![Diagram of the collaboration ontology proposed by Knoll et al.]

**Figure 7**: The collaboration ontology proposed by Knoll et al.

### 4.3 Decision Making Ontologies

#### 4.3.1 Decision Support Ontology

(Rockwell et al., 2009) proposes a Decision Support Ontology (DSO) for Decision Making in the field of engineering design. Although this work’s initiative was to create an information model based on the engineering design literature, the proposed ontology deals with generic decision information capturing and therefore extends the boundaries of the specific domain. The main concepts and the corresponding relations among the proposed concepts are depicted in Figure 8.

The above work identified the following information basic types as key decision making concepts: design issue, alternatives, criteria and evaluation information (Ullman, 2002, Dwarakanath and Wallace, 1995). In other words, a decision is made on a specific design issue, the choice has to be made among two or more alternatives (solutions to the issue)
which are evaluated by using a set of criteria. The alternative chosen has to be the one mostly satisfying the criteria. The preference model is related to the objective function and the desired value of the objective function.

![Diagram](image)

**Figure 8**: Key concepts and relationships in the DSO (Rockwell et al., 2009, Ullman, 2002).

### 4.3.2 Ontology for Information Systems Engineering

Kornyshova and Deneckère (2010) have presented an ontology for enhancing data mining (DM) and for supporting data mining activities in the field of Information Systems Engineering (ISE). The proposed ontology aims at supporting IS engineers in generic DM cases and it includes the necessary components (elements and links) for DM in a number of situations. The ontology includes concepts, attributes and relationships (Kornyshova and Deneckère, 2010). Figure 9 depicts the UML class diagram of the ontology.

The DM situation is the central element of the ontology and groups together all the other ontology concepts. The DM object may be a process or a product element. A DM situation includes a DM problem, a set of alternatives and, maybe, a set of criteria. The criteria may be of different types (characteristics of alternatives, consequences of alternatives, decision’s makers’ goals or decision makers themselves). Further attributes may be set for each criterion such as its weight, preference rule or threshold. The identified criteria are used to evaluate all the given alternatives (resulting in the alternative values). In the scenario “described” from the proposed ontology, decision makers define the data mining problem, use the preference rules, weights and thresholds to express their preferences and validate the final decision.
Figure 9: The proposed Decision Making Ontology (Kornyshova and Deneckère, 2010)

4.3.3 Ontology for Decision Support in Health Informatics

Nykänen (2002) used as reference the model presented in (Ives et al., 1980) to present an extended ontology for a decision support system in health informatics. (Ives et al., 1980) defined a number of variables that have to be taken into account when dealing with information systems; three information system environments (operations, development and user environment), three information system processes (use development and information processes) and the information subsystem.

Nykänen (2002) suggests the application of the model presented in (Ives et al., 1980) to the development of a DSS in health informatics with some slight changes in the original variables. From the variables of this model, the variables concerning the operation environment and the operation process are omitted (they are not considered essential for information systems). New variables have been added to the model; environment variables (external environment, organizational environment, IS development environment, user environment), process variables (development process, use process) and information system variables (content information, representation, scale timestamps, knowledge types, application specialization).

The variables that have been selected for the conceptualization of the DSS come from three sources; the original model of Ives et al. (1980), analysis of how a decision support system has been defined in areas of information systems science and artificial intelligence and the results from three cases studies (Nykänen, 2002).
4.4 Workflow Ontologies

To this end, the Workflow Management Coalition (http://www.wfmc.org/) has produced a Workflow Reference Model in 1993 and updated it in 1995. The purpose of this model is to identify the characteristics, terminology and components for workflow management systems. Process semantics are represented in standards such as XPDL, BPEL4WS and BPML (Hollingsworth, 2004). The Workflow Reference Model recognises three different data classes: workflow control, workflow relevant, and application data.

Workflow templates, i.e. abstract workflows that can mix executable operators and tasks to be refined later into sub-workflows, are discussed in (Kietz et al, 2010) This representation helps users to structure and handle workflows, as it constrains the number of operators that need to be considered. Workflows can further be grouped in templates which foster re-use and simplifying KDD workflow construction.

A workflow description for collaboration services is provided in (Zhen et al, 2009). It looks at three layers: team tasks, team roles, and members (profiles). Different layers are represented separately and appropriate mappings are made.

MyGrid Ontology

Taverna (an open source domain independent Workflow Management System) uses myGrid Ontology which describes the bioinformatics research domain and the dimensions with which a service can be characterised from the perspective of the scientist. The ontology is logically separated into: the service ontology and the domain ontology. The domain ontology acts as an annotation vocabulary including descriptions of core bioinformatics data types and their relationships to one another, and the service ontology describes the physical and operational features of web services, such as, inputs and outputs.

myExperiment ontology

myExperiment is a collaborative environment where scientists can safely publish their workflows and experiment plans, share them with groups and find those of others. The myExperiment data model has three main underlying features: content management, social networking and object annotation. The myExperiment Ontology uses concepts from several ontologies: Dublin Core (meta data description), FOAF and SOIC (social networks), Creative Commons (licence description), OAI-ORE (resource aggregation), and DBPedia (user residency).
5 DON Framework Outline

Dicode will follow an iterative model for networked ontologies lifecycle. The ontology requirements will be gradually expanded and tuned to reflect the corresponding stages in the Dicode project.

Figure 10. DON iterative lifecycle.

DON will be created in three iterations, as indicated in Figure 10.

5.1 DON iteration 1

This iteration will capture stakeholder perspectives from the three use cases in Dicode. It will aim at identifying the specific and generic aspects in each use case, starting from the initial outline provided in Section 1.2.2 and a description of the Dicode use cases (Deliverable D2.2., due in month 8). The output of this iteration will be an OWL model of generic and use-case specific ontologies.

5.2 DON iteration 2

This iteration will include validation and tuning, after which DON will be used as a tool to compare the three use cases and to provide guidance to service developers regarding generic and specific descriptions of their services. Requirements for connecting services to DON will be derived, which will lead to expanding the generic layer with semantic categories linked to service performance and service workflows. We also envisage providing tool support for semantic content annotation as needed. At the end of this iteration, a first version of the integrated DON will be delivered, and appropriate documentation provided.
5.3 DON iteration 3

This iteration will repeat the main stages from iterations 1 and 2, starting with revisiting the ontology requirements, ontology tuning, and expansion. It is envisaged that at this stage, the research challenges outlines in Section 1.2 will be revisited, and appropriate evaluation will be conducted to examine how well these challenges have been addressed. We also expect extension of the ROO suite of services to automate the semantic annotation of semi-structured natural language service descriptions (e.g. by using appropriate information extraction techniques). The application of the ontology will be tested with several Dicode services, as well as with external services and applications. The final version of DON will be delivered, supported with appropriate documentation.

An initial vision of the final integration of DON into the Dicode Framework is presented in Figure 11. This vision will be reviewed, and amended accordingly, based on the forthcoming description of Dicode use cases, services, and integrated architecture (which will be specified in follow up Dicode deliverables from WP2 and WP5).

![Figure 11. Integration of DON in the Dicode Services Framework](image)

5.4 DON Publication

The Dicode Ontology will be published within the official web site of the project (http://www.dicode-project.eu). DON will be codified in OWL format and it will be freely available. The DON generic level will be provided together with semantic enhanced Dicode service descriptions to enable reusability and interoperability of Dicode services in use cases outside those considered in the project.
6 DON Iteration 1 Description

The Dicode Ontology Framework adopts an agile methodology. Following this, at the time of preparing this deliverable, the work concentrates on the first iteration. Based on the output of this iteration, the following iterations will be planned and conducted. This section will provide a detailed description of our plan of the first iteration. This has a twofold purpose. On the one hand, we will illustrate the main aspects to be considered when planning an iteration, including: (a) defining scope and purpose of DON in that iteration; (b) defining (in iteration 1), and extending (in the following iterations) the conceptualisation the generic and specific ontology layers; (c) formalisation, and (d) application. On the other hand, the iteration 1 description will be used for gathering the requirements for the Dicode multi-perspective ontology engineering tool (Deliverable 5.3).

6.1 Iteration 1 Purpose and Scope

**Purpose.** The purpose of DON in iteration 1 is:
- Gaining an understanding of data intensive collaboration and decision making;
- Facilitating comparison between the three Dicode use cases;
- Providing a common vocabulary to facilitate collaboration between Dicode service developers.

**Scope.** DON Iteration 1 will include 4 ontologies. The generic DON will include key classes and axioms describing data intensive collaboration and decision making. The 3 use case specific ontologies will include instantiations of key concepts from the generic DON to represent specific aspects of each use case. They will also include additional concepts to represent application specific aspects that have not been captured in the generic ontology.

![Figure 12. DON Framework in Iteration 1.](image-url)
6.2 Iteration 1 Conceptualisation: Generic Layer and Use Case Specific Layers

The conceptualisation will combine top-down (generic) and bottom-up (use-case specific) analysis. This will lead to two ontology layers: generic and use-case specific. Figure 12 represents the methodology followed in iteration 1.

6.2.1 DON Generic Layer: Collaboration Analysis

A ‘top-level’ ontology representing key dimensions of data intensive collaboration and decision making. This ontology which will be derived by reusing relevant concepts from the ontologies listed in Section 4.2. They conceptualisation methodology will follow an appropriate theoretical framework for domain analysis.

Possible theoretical frameworks that can be used for analysis of collaborative work are Activity Theory or Distributed cognition (Halverson, 2002). Rogers (1994) argues that distributed cognition enables ‘studying the dynamics of collaborative activity in situ’, while Nardi (1996) shows that Activity Theory can provide a rich framework, in terms of comprehensiveness and engagement, for capturing context. Furthermore, Activity Theory is recently widely used in for analysing user interaction in HCI and CSCW (Diaper, 2008), and there are practical examples demonstrating the usefulness of Activity Theory for identifying the upper level in an enterprise ontology (O’Leary, 2010).

6.2.2 DON Generic Layer: Decision Making Analysis

Possible theoretical frameworks that can be used for analysis of decision making are organization-oriented macro decision analysis (Anthony, 1965) and process-oriented micro decision analysis (Mintzberg et al, 1976). The two levels of analysis have been followed successfully for formulating a common vocabulary and a model to improve the manageability of the requirements engineering process for decision making activities (Aurum & Wohlin, 2003).

Naturalistic decision making (Salas et al. 2010) can be used to capture the tacit/intuitive aspects of expert decision making, considering: application of domain knowledge; pattern recognition; sensemaking; situation assessment; automaticity; and mental simulation.

The theoretical analysis will be grounded in the three Dicode use cases, which provide rich enough context for capturing generic aspects of data-intensive collaboration and decision making, as shown in Section 1.2.2.

6.2.3 DON Generic Layer: Data Analysis

The generic layer will include a description of the possible data sets users may utilise. Generic taxonomies will be used. Examples include:

**Classification for Statistical Learning from Data (Yang et al, 2010)**
- **Qualitative (Categorical):** data that can be placed into distinct categories. No arithmetic operations can be applied
- **Quantitative (Numeric):** numeric in nature and can be ranked in order. Admit to meaningful arithmetic operations
  - **Continuous:** can assume all values on the number line within their value range and obtaining by measuring
• **Discrete:** values that can be counted but cannot assume all values on the number line within their value range.

**Classification based on the level of data measurement scales (Yang et al, 2010)**

![Diagram of data classification based on measurement](image)

Figure 13. Structured data classification based on measurement (Yang et al, 2010).

**Classification based on Data Structure (Chakrabarti, 2003)**

• **Structured:** Data organized into semantic chunks or entities, with similar entities grouped together in relations or classes, and presented in a patterned manner.

• **Semi-structured (tagged, marked):** a form of structured data that does not conform with the formal structure of tables and data models associated with databases but contains nonetheless tags or other markers to separate semantic elements and hierarchies of records and fields within the data.

• **Unstructured:** Data that either does not have a data model or has one that is not easily usable by a computer program.

**Classification based on Data Pre-Processing (Hornick et al, 2007)**

• **Normalized:** data that compress the scale of its attribute’s values.

• **Binned:** data that has been reduced by the cardinality—the number of distinct values—of its attribute

• **Exploded:** explosion applies only to categorical (nominal or ordinal) attributes in order to transform such attributes into numerical attributes

• **Recoded:** data that its original value is replaced by another value for unification purposes

**Classification based on Data Format**

• **Data Type:** constraint placed upon the interpretation of data in a type system

• **Signal:** a format for signal data used in signal processing

• **Recording format:** a format for encoding data for storage on a storage medium

• **File format:** a format for encoding data for storage in a computer file

• **Content format:** a format for converting data to information
  • **Text Format:** a format for processing textual data
  • **Audio Format:** a format for processing audio data
  • **Video Format:** a format for processing video data

Note that there is a huge literature on data formats. Most data formats have been developed for specific domains, so that a biological data format would be of little use in a situation in which a multidimensional array data is needed.

**Classification for Data Visualizations (Shneiderman, 1996)**

• **1-dimensional:** textual documents, source code, etc.

• **2-dimensional:** planar and map data, e.g geographic maps, floorplans
• 3-dimensional: real-world objects such as molecules, human body
• Temporal: time lines data such as medical records, project management data, user’s webpage click streams in a website
• Multi-dimensional: relational and statistical databases in which items with n attributes become points in an n-dimensional space
• Tree: collections of items with each item having link to one parent item (except the root). Items and the links between parent and child can have multiple attributes.
• Network: extension to the tree data type where relationships among items are more complex and so items are linked to an arbitrary number of other items.

Data Properties relevant to data modelling (West and Fowler, 1999)
• definition-related properties
  • relevance: the usefulness of the data in the context of your business
  • larity: the availability of a clear and shared definition for the data
  • consistency: the compatibility of the same type of data from different sources
• content-related properties
  • timeliness: the availability of data at the time required and how up to date that data is
  • accuracy: how close to the truth the data is
• properties related to both definition and content
  • completeness: how much of the required data is available.
  • accessibility: where, how, and to whom the data is available or not available (e.g. security).
  • cost: the cost incurred in obtaining the data, and making it available for use.
Figure 14. Initial Data Classifications for Data Mining Services (Combining Classifications 1, 2, 3 & 4 in one taxonomy).
Taxonomy describing social data

Three layers processing social data:
- **Media Intelligence** layer applies semantic analysis to the content items, taking into account the content itself (text, images, video, speech), existing tags, personal, social and associated metadata (e.g. position, time)
- **Mass Intelligence** layer combines the information from mass user feedback in order to extract patterns and trends that cannot be extracted by single content items
- **Social Intelligence** layer analyses the usage and communication interaction patterns taking into account the needs and capabilities of communities and provides useful output both for the creation of the Media and Mass as well as for the WeKnowIt applications

Unstructured social data in various formats:
- Text (comments, reviews, posts)
- Images (flickr)
- Video (youTube)
- Speech.

Structured / Semistructured social data:
- Tags (annotations).
- User – Item ratings
- Social Metadata (e.g. tweets showing the location of user, time of posting a blog post)

- Aggregation
- Blog
- Blogroll
- Mashup
- Moblogging
- Microblogging
- Newsreader
- Podcast
- RSS
- Social Bookmarking
- Social Networking
- Tags
- Wiki

6.2.4 DON Use-case Specific Layers

Three ‘application-specific’ ontologies, one for each Dicode use case, will be derived to capture the specificity of data intensive collaboration and decision making in the use cases. These ontologies will be derived reusing domain ontologies from those listed in Section 4.1., and following scenario-based analysis. The initial specific dimensions have been pointed out in Section 1.2.2. Starting from there, we will also consider description of the data sets in each use case. To identify the use case specific aspects, relevant activity engaging use case stakeholders will be conducted, e.g. brainstorm sessions/workshops, collaborative ontology construction with an ontology engineering, ontology inspection.
6.3 Formalisation

The formal description of the Dicode ontologies will be in OWL. The formalisation will be conducted using the ROO tool (see Section 2.3.3).

6.4 Application

The ontologies in iteration 1 will be used to discover similarities and discrepancies between the use cases. For this, ontology processing will be conducted, using tools from the NEON Toolkit and the Protégé suite of plugins. Specifically, we envisage the use of:

- Ontology visualisation tools;
- Ontology mapping tools;
- Ontology alignment tools;
- Ontology analysis tools.
7 Conclusions

This deliverable has outlined the Dicode Ontological Framework. The Dicode Ontology (DON) will be developed and used for several purposes:

- gaining an understanding of data intensive decision making and collaboration by identifying generic and use-case specific aspects - this will facilitate the conceptual integration of the Dicode use cases;
- enabling the interoperability and integration between Dicode services (as well as services from third parties) - this will facilitate the deployment and re-use of Dicode services within and outside the Dicode project;
- providing semantic description to augment the functionality of some Dicode services;
- providing a common vocabulary to facilitate the collaboration between stakeholders and developers of Dicode services.

The key challenges addressed in the DON framework are:

- dealing with data, service, and stakeholder heterogeneity;
- identifying generic and specific aspects if the Dicode use cases;
- providing intuitive means for ontology authoring and use by people without knowledge engineering background and experience.

Relevant technologies and methodologies are reviewed in order to provide the foundations for the DON framework. Appropriate semantic standards and ontology modelling frameworks will be followed - it is envisaged that the ontology will be coded in RDF or OWL, to ensure sufficient reasoning level and interoperability with other ontological frameworks. An agile, iterative methodology will be followed, adapting the NEON ontology cycle model. Appropriate services will be provided for the intuitive ontology authoring and use, based on an ontology authoring tool called ROO. Following relevant semantic approaches for service interoperability, DON will be utilised to facilitate the integration of Dicode services.

The DON foundations include also relevant ontologies that can be used for the generic and use-case specific layers. A detailed review has been provided. It is envisaged that key concepts and modelling conventions from these ontologies can be re-used/followed in DON.

The DON framework includes a high-level description of the Dicode ontology and an outline of the ontology authoring and use process. DON will include different modules (layers) which describe the services, the subject domain, the users, the decision making and collaboration tasks. These layers are broadly classified in generic (representing the common aspects of data intensive decision making and collaboration) and specific (representing the unique characteristics of a specific application scenario involving data intensive decision making and collaboration, which will be illustrated with three testbeds corresponding to the Dicode use cases). Apart from separating the key layers, it is important to identify the linking between different them, i.e. how can we combine different modules and link concepts between modules.

The ontology authoring and use in Dicode will follow an iterative, agile life cycle. An initial vision of the ontology lifecycle has been presented. This vision will be reviewed, and amended accordingly, based on description of Dicode use cases, services, and integrated architecture (which will be specified in forthcoming deliverables in WP2 and WP5).
Following the agile approach, a detailed description of the first iteration is provided. This description illustrates the main aspects to be considered when planning an iteration, including: (a) defining scope and purpose of DON in that iteration; (b) defining (in iteration 1), and extending (in the following iterations) the conceptualisation the generic and specific ontology layers; (c) formalisation, and (d) application. In addition, the description of iteration 1 is used as the key source for specifying the requirements for the Dicode multi-perspective ontology engineering tool to be provided in Deliverable 5.3.
References


Duncan Hull (2008) Semantic Matching of Bioinformatic Web Services, A thesis submitted to the University of the degree of Doctor of Philosophy in the Faculty of Engineering and Physical Sciences, School of Computer Science.


Kotis, K. & G. Vouros. (2006). Human-centered ontology engineering: The HCOME methodology, Knowledge and


