



NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – “Semantic-based knowledge and content systems”

D7.7.1 Evaluation of the FADAS first prototype and recommendations to research

Deliverable Co-ordinator: Claudio Baldassarre, Yves Jaques

Deliverable Co-ordinating Institution: Food and Agriculture Organization of the United Nations (FAO)

Other Authors: Claudio Baldassarre, Yves Jaques

This is an abstract of generic template for the NeOn project, which shall contain all the necessary styles to help you produce a reasonably and consistently looking deliverable.

Document Identifier:	NEON/2008/D7.7.1/v1.2	Date due:	September 30, 2008
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	September 30, 2008
Project start date:	March 1, 2006	Version:	v1.2
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Community, grant number IST-2005-027595. The following partners are involved in the project:

<p>Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Martin Dzbor, Enrico Motta E-mail address: {m.dzbor, e.motta} @open.ac.uk</p>	<p>Universität Karlsruhe – TH (UKARL) Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 28 D-76128 Karlsruhe, Germany Contact person: Peter Haase E-mail address: pha@aifb.uni-karlsruhe.de</p>
<p>Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es</p>	<p>Software AG (SAG) Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com</p>
<p>Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com</p>	<p>Institut 'Jožef Stefan' (JSI) Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si</p>
<p>Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: jerome.euzenat@inrialpes.fr</p>	<p>University of Sheffield (USFD) Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: hamish@dcs.shef.ac.uk</p>
<p>Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de</p>	<p>Consiglio Nazionale delle Ricerche (CNR) Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 - 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it</p>
<p>Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de</p>	<p>Food and Agriculture Organization of the United Nations (FAO) Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Marta Iglesias E-mail address: marta.iglesias@fao.org</p>
<p>Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com</p>	<p>Laboratorios KIN, S.A. (KIN) C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es</p>

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed writing parts of this document:

N/A

Change Log

Version	Date	Amended by	Changes
0.5	14/08/2008	Claudio Baldassarre	Test Plan and Test Specification sections are completed in the scope of technical validation. Results of client installation, and use case test, are in Annex 5 and Annex 6.
0.6	18/09/2008	Yves Jaques	Addition of usability results
0.7	19/09/2008	Claudio/Yves	Compilation of conclusions/recommendations
0.9	25/09/08	Claudio Baldassarre	Refinement of content distribution, minor narrative corrections
1.0	26/09/08	Yves Jaques	Refinement of content distribution, minor narrative corrections
1.1	10/10/2008	Claudio Baldassarre	Reviewer comment addressed
1.2	23/10/2008	Yves Jaques	Reviewer comment addressed

Executive Summary

This document evaluates D7.6.1 Fisheries Stock Depletion Assessment System (FSDAS) and makes recommendations for future iterations using a set of commonly applied software evaluation methods. These methods verify that the software implements the functionalities and meets the constraints described in FSDAS requirements documents D7.1.1 and D7.1.2 as well as FSDAS architecture document D7.5.1. The methods also validate the software by examining whether it generally conforms to software engineering standards for development. Finally the methods examine software from a usability perspective by observing typical users performing supported tasks.

The report found that the FSDAS v1.0 only partially supports the basic requirements and functionalities earmarked for phase one development, and that a large portion of its usability defects are due to its inclusion as a feature within the overall NeOn Toolkit Eclipse platform, an approach that mixes its interfaces with those of the underlying toolkit. A number of recommendations are made to assist in the creation of a more compliant v2.0.

Table of Contents

Work package participants	3
Change Log	3
Executive Summary	3
Table of Contents	4
1. Background	7
2. Process	8
3. Test plan.....	9
3.1 Introduction	9
<i>Planning Background.....</i>	9
<i>Test Objectives</i>	9
<i>References.....</i>	9
3.2 Technical test items	10
<i>Interface</i>	10
<i>User management</i>	10
<i>Search.....</i>	10
<i>Query</i>	10
<i>Annotation.....</i>	10
<i>Communication</i>	10
<i>Reasoner</i>	11
<i>Data repository.....</i>	11
<i>Web Server.....</i>	11
3.3 Technical features tested	11
<i>Interface features.....</i>	11
<i>User management features</i>	11
<i>Communication features.....</i>	11
3.4 Usability features tested.....	12
3.5 Undelivered items	12
<i>Untested features.....</i>	12
<i>Untested use cases</i>	13
3.6 Testing approach	14
3.7 Item pass/fail criteria	14
3.8 Environmental needs	14
<i>Hardware</i>	14
<i>Software.....</i>	15
4. Test specifications	16
4.1 Technical assessment.....	16
<i>Server side testing of FSDAS.....</i>	16
<i>Client side testing of FSDAS</i>	16
4.2 Usability assessment	17
<i>User profile.....</i>	17
<i>User tasks.....</i>	18

<i>User post-test questionnaire</i>	18
5. Test reporting	19
5.1 Technical test report.....	19
<i>Covered/uncovered requirements ratio</i>	19
<i>Precision of requirement coverage</i>	19
<i>Software configuration</i>	19
5.2 Usability test report	19
<i>General</i>	19
<i>Taskbar</i>	19
<i>Menu</i>	20
<i>Exit</i>	20
<i>Registration</i>	20
<i>Help</i>	20
<i>Import wizard</i>	20
<i>Concept instance perspective</i>	21
<i>Ontology visualizer perspective</i>	21
<i>Concept query perspective</i>	21
<i>Data query perspective</i>	21
<i>Email results</i>	22
<i>Bookmark/Favourites</i>	22
<i>Save session</i>	22
6. Conclusions	23
6.1 Technical summary	23
6.2 Usability summary	23
<i>General</i>	24
<i>Taskbar</i>	24
<i>Menu</i>	25
<i>Exit</i>	25
<i>Registration</i>	25
<i>Help</i>	26
<i>Import wizard</i>	26
<i>Concept instance perspective</i>	27
<i>Ontology visualizer perspective</i>	27
<i>Concept query perspective</i>	27
<i>Data query perspective</i>	28
<i>Email results</i>	29
<i>Bookmark/Favourites</i>	29
<i>Save session</i>	30
7. Recommendations	31
7.1 Use case prioritization	31
7.2 Functional improvements	31
<i>Interface</i>	31
<i>Sessions</i>	31
<i>Email</i>	32
<i>Data sources</i>	32
<i>Queries</i>	32
List of Acronyms used in this deliverable	33
References	34

Annex 1 – Template report for the use cases testing.....35
Annex 2 – Usability testing templates37
Annex 3: List of FSDAS use cases.....42
Annex 4 – Template report for the FSDAS client installation43
Annex 5 – Result of FSDAS client installation44
Annex 6 – Result of FSDAS use cases test.....45
Annex 7 – Usability testing task summary68
Annex 8 – Usability testing post-task questionnaire summary.....71

1. Background

The Capability Maturity Model (CMM) defines the purpose of validation as being “to demonstrate that a product or product component fulfils its intended use when placed in its intended environment.” [CMM] There are two main areas of software proofing: validation and verification. “Validation demonstrates that the product, as provided, will fulfil its intended use; whereas, verification addresses whether the work product properly reflects the specified requirements. In other words, verification ensures that ‘you built it right;’ whereas, validation ensures that ‘you built the right thing.’”

Deliverable [D7.6.1](#), the FSDAS v1.0 prototype was produced based on a number of NeOn deliverables, particularly FSDAS requirements documents [D7.1.1](#) and [D7.1.2](#) and FSDAS architecture document [D7.5.1](#). Its aim is to “provide FAO Fishery systems with a great opportunity to develop an appropriate framework to manage fishery ontologies and their lifecycle, as well as to implement a semantic fishery stock depletion alert system that exploits those ontologies.” [[D7.1.1](#)]

D7.1.1. summarizes the scope and core requirements of the application:

- FSDAS users will be mainly officers in FAO Fisheries Department; in this sense, the user classes are all fisheries experts within some domain of fisheries: Fisheries Scientists, Fisheries Managers, Marine Biologists, Oceanographers, Fisheries economists, Fisheries legal experts, Fisheries engineers, Fisheries policy makers.
- Users are able to browse or search fisheries concepts and relations and use them to seek out related concept instances that have been identified using reasoning over the set of fisheries ontologies.
- Users are able to navigate and query concepts, relations and concept instances; to view all concepts, relations and the application interface in the five languages of FAO (when available).
- Users are able to view the digital resources related to the concept instances within the appropriate application on their desktop computer.
- Users are able to input ad-hoc queries, both using free-text and concepts/relationships, suggested by the currently loaded ontology set, that returns either related data instances or related concepts/relationships found in the currently loaded ontology set.
- Data instances returned by a query are grouped according to their related concepts and relationships; it should be possible to re-organize results according to any of the concepts/relationships that were used to select the result.

This deliverable documents the verification and validation activities carried out in the evaluation of this prototype and makes proposals for its next iteration.

2. Process

This document, in order to fit within the context and scale of this deliverable is based on a simplification of [IEEE 829-1998 Standard for Software Test Documentation](#). Some of the adaptations consist of implementing as deliverable sections, items that the IEEE standard envisages as separate document sets (e.g. test plan set, test specification set, test reporting set). For each section, only the part of the documentation relevant to the pursued purpose is implemented. In some cases a schematic view is given to what otherwise would be a wider document collection, e.g. test case specification documents. The choice to adapt the original IEEE documentation model should not however affect the soundness and completeness of the validation activity. All tests were performed as required to guarantee full support for the next iteration of FSDAS prototyping work.

The document contains 5 main sections:

Test Plan: covers the rationales collected in the period before and after the FSDAS application was delivered, and that led to the specifications of the tests to validate and verify the prototype. This section prescribes also the perspectives under which tests were performed and the parts of the application that required special investigation. The section finishes by covering the objectives of the FSDAS testing.

Test Specification: describes in more detail aspects mentioned in the plan: the requirements to be covered, the features to be tested and the use cases to be verified. The section refers to templates that embed the main properties representative of the tests for use cases; filling the schema for each use case gives a concise view of what would otherwise require more extensive documentation according to the [IEEE 829-1998 standard](#).

Test Reporting: describes the actual implementation of the specified tests.

Conclusions: summarizes test reporting; provides pinpointed improvements to the parts that have been tested and found insufficient with respect to the requirements; gives some qualitative opinions on the evaluated deliverable.

Recommendations: contains advice and suggestions for the next prototype of FSDAS based on the test results and following FAO expectations in terms of functionalities and requirements coverage.

Due to the combination of different tests and test items, it is not possible to preset evaluation criteria that cover the entire testing activity; generally speaking, a success criterion is requirement satisfaction and the extent of use case coverage as presented in D7.5.1. Usability on the other hand is necessarily a qualitative measure that seeks to offer guidance towards future improvement of the user experience. Because of the specificity of requirements and scope, and uniqueness of the application aim, one broadly valid success indicator is that Fisheries' users are able to speed up the collection of data that otherwise need to be retrieved and selected manually from a heterogeneity of data format and repositories.

3. Test plan

3.1 Introduction

The plan covers two test perspectives: technical testing and end-user usability. The verification activity sets the ground for end-users ensuring that the application is basically sound and communicates any encountered issues. The second part of testing focuses on validating usability by fisheries domain experts. A task script is used and direct user observation performed using think-aloud protocols. The results of these two groups of effort are then collated and analysed.

Planning Background

The specifications for the FSDAS application have matured in accordance with the increasing understanding that FAO has acquired about the evolving design and capabilities offered by the NeOn architecture. In a span of 22 months time, and through the production of three deliverables (i.e. [D7.1.1](#) FSDAS Requirements, [D7.5.1](#) FSDAS Architecture and [D7.1.2](#) Revised FSDAS Requirements) FAO was able to provide detailed specifications for first and next iterations of the application prototypes. D7.5.1 described the architecture design according to the requirements collected in D7.1.1, and also a design subset, focused on requirements considered priority for the first FSDAS version.

At a second stage FAO reconsidered the requirements of D7.1.1 in the light of project outcomes and perspectives, producing in D7.1.2 the final revision of requirements. Although this deliverable arrived during the implementation of [D7.6.1](#), it had some impact upon the design and implementation decisions that followed. The willing reader can refer to D7.1.2 for complete descriptions and motivations for requirements evolution.

Test Objectives

Guided by the experience acquired since the first attempt to define what the FSDAS application should feature, testing targets the following objectives:

- (a) Report on uncovered issues in first prototype that are relevant to FAO but not envisaged in NeOn.
- (b) Drive prioritization of functionalities development as the result of a realistic schedule of delivery.
- (c) Provide guidelines and recommendations to second prototype developers to improve the next FSDAS release.
- (d) Make suggestions with respect to any of the points in the evaluation criteria that need to be improved during ongoing work for the next release.

References

The following documents were used as sources of information to plan the FSDAS tests:

- [D7.5.1](#): Software architecture for the ontology-based Fisheries Stock Depletion Assessment System (FSDAS).
- [D7.2.1](#): Inventory of fishery resources and information management systems.

- [D6.1.1](#): Requirements on NeOn architecture.
- [D6.1.2](#): Report on user requirements V2.
- [D6.2.1](#): Specification of NeOn reference architecture & NeOn APIs.

3.2 Technical test items

The FSDAS application was designed and implemented as a client-server application, hence both halves are part of the test plan.

The client side released for the first version of FSDAS can be further partitioned into the following architectural components:

Interface

User interface access to all functionalities requiring widget elements must be tested. The graphical environment is mainly a target of usability tests to validate if the actual structure and layout are easily accessible by the user classes specified for FSDAS.

User management

These tests target the soundness of the representation of a user inside the FSDAS; i.e. the attributes that characterize his/her description and how the rest of the system makes use of this information.

Search

Searching functionality is limited to basic features for the first prototype, nevertheless look-up mechanisms must be tested for ontologies loaded within the application environment.

Query

This is a crucial set of features for FSDAS, hence tests must assure that users have access to all the planned mechanisms for retrieving ontological resources and their associated data instances. Accessibility, availability and correctness of result items must be tested.

Annotation

Tests run on this subsystem attempt to verify that users can add additional information to ontological elements and document objects that are valid within the user scope.

Communication

Communication is the partition of the FSDAS client that allows wrapping of objects generated using query features and send them via email to other users. Test focus on verifying that a communication instantiation can take place correctly.

The server side is composed of infrastructure components plus a number of web services consumed by the client side. This application tier is partitioned into the following parts:

Reasoner

This is the core of the server; the OntoBroker inference engine from Ontoprise is the connection cross point between the client and the distributed data sources. It is fed with a pool of ontologies (developed in [D7.2.2](#)) for which it builds an internal representation and uses as an interface to the data. Tests must verify this integration with data sources.

Data repository

At this stage of development the data consists of a single index of fisheries fact sheets. Tests target the integration of this data source to the inference engine to assure that user queries executed through the reasoner can correctly access the data.

Web Server

Network communication must be tested according to the specifications in [D7.6.1](#), particularly for configurations that consider proxy or firewall settings.

3.3 Technical features tested

In correspondence to the test items listed above from **0 to 0**, below is a list of features that the first prototype of FSDAS is expected to cover.

Interface features

- The system shall support several modes of viewing ontologies:
 - Rubber-band
 - Hierarchical
- The system shall allow authorized users to browse fisheries concepts, attributes and relations.
- The system shall be able to display multiple ontologies, including the relations eventually holding among their referenced elements

User management features

- Users shall be able to register, login and maintain a profile
- The default system settings are loaded according to user type, e.g. ontology loaded, underlying connected data systems, and a scope for data instance display.

Communication features

- Users shall be able to email formatted results.

- Users shall be able to suggest ontology modifications that was forwarded to the ontology owner via email.

3.4 Usability features tested

Usability testing focuses on the use cases that are known to be at least partially covered in this first prototype. The following use cases must be tested:

- UC1 Login
- UC2 Logout
- UC3 Register
- UC8 Browse Taxonomy
- UC11 Query Composition
- UC12 Query for Data related to individual
- UC14 Refine query
- UC17 Email results
- UC18 Propose ontology modification
- UC19 Add to favourites
- UC20 Save session
- UC31 View data instance summary

3.5 Undelivered items

The following features were not delivered and therefore cannot be tested:

Untested features

- *Users shall be able to view all concepts, relations and the application interface in the five languages of FAO, when available.*
- *Users shall be able to view the digital resources related to concept instances.*
- *Raw data that is not associated with a proprietary application shall be viewable within the system, and not require the launching of a separate application. This applies specifically to tabular data such as CSV files, hierarchical data such as XML files and HTML files, and image data such as PNG, JPEG and GIF files.*
 - *It shall be possible to view two data sources side by side for comparison.*
- *Data instances shall be presented within their associated ontological context and associated metadata crucial to understanding shall also be displayed.*
- *Users shall be able to bookmark and access favourite queries, ontologies, ontology concepts, default language and ontology subsets.*

- *Users shall be able to join a particular user type, e.g. biologist, economist etc.*
- *Users shall be able to search (look up) for ontological elements in all the ontologies loaded in the system*
- *Users shall be able to input ad-hoc queries, both using free-text and by highlighting concepts/relationships suggested by the currently loaded ontology set that shall return either related data instances or related concepts/relationships found in the currently loaded ontology set.*
 - *In this context Boolean logic, phrase matching and query refinement shall be supported by the system.*
- *Users shall be able to select to rank results based on the rankings filtered by their user type or other annotation data.*
- *Data instances returned by a query shall be grouped according to their related concepts and relationships.*
- *Users shall be able to annotate (including comments, keywords and quality scores) and rate concepts, attributes, relations and associated data instances.*

Untested use cases

The following use cases were not delivered and therefore cannot be tested:

- *UC4 Modify Profile*
- *UC5 Modify User Account*
- *UC6 Search ontological resource in ontology*
- *UC7 Search for related ontological resources*
- *UC9 Change language of the interface*
- *UC10 Change language of the ontological resource shown*
- *UC13 Visualize Data Source related to individual*
- *UC15 View ontological resource annotation*
- *UC16 View Data Source Annotation*
- *UC21 Generate RSS feed from current query*
- *UC22 Annotate retrieved document with comments on quality*
- *UC23 Index enrichment against ontology(ies) domain model*
- *UC24 Annotate retrieved document with comments*
- *UC25 Select ontologies to use for browsing documents or web pages*
- *UC26 Use ontology to support browsing*
- *UC27 Identify trend*
- *UC28 Compare data by reporter*
- *UC29 Discover Ontology*

- UC30 *Filter Ontology*
- UC32 *Re-group / rank data resource by annotation*

3.6 Testing approach

Testing covers two main perspectives and generates two different types of tests: technical and end-user usability. The first covers testing both the server and client side of FSDAS. The initial step is installation inside the FAO information system infrastructure of the server side of FSDAS to make sure that all repositories needed by the application can be successfully integrated and that any client instantiation can correctly communicate with the central server.

Client side testing begins by installing the application on different computers running Windows XP to ensure that the application meets non-functional hardware and software requirements. This is followed by the execution of the use cases planned to be delivered for the first prototype as listed and prioritized in [D7.5.1](#) and reported in [Annex 3](#). In Annex 3, for each use case a validation template is prepared (master copy in [Annex 1](#)) and filled with test results, final conclusions and recommendations.

Along with testing technical and technological aspects of FSDAS, usability is also an item of testing. An evaluation strategy is defined using semi-structured field observation based on task scripts ([Stone et al., 2005](#)) and executed using think-out-loud protocols ([Open University, 2001](#)). It consists of:

- A session script,
- user profile survey,
- series of scripted browsing tasks that covered major functionality,
- logbook to record user remarks and observer comments and
- a post-session questionnaire.

These usability tests are performed as a last step after the soundness and completeness of FSDAS is verified. A task script ([Annex 2](#)) includes typical activities a user might perform with the FSDAS system that cover the delivered use cases. Users are invited to act according to the task script actions and are asked a series of questions at each step.

3.7 Item pass/fail criteria

Due to the combination of different tests and test items, it is not possible to preset a single pass/fail value for the delivered functions; they are instead specified ad-hoc for the test scope, and clearly explained in the relevant sections of this document. Generally speaking, success criterion is requirement satisfaction and extent of use case coverage as in D7.5.1. Usability on the other hand is necessarily a qualitative measure that seeks to offer guidance towards future improvement of the user experience.

3.8 Environmental needs

Hardware

Testing of the *client* side of the application is performed on a computer system meeting non-functional requirements as described in D7.1.1 and amended in D7.1.2:

- Computer system must be equipped with not less than 1 gigabyte of RAM

The computer system running the *server* side also follows the requirements envisaged in D7.1.1 and amended in D7.1.2:

- Computer system must be equipped with not less than 4 gigabytes of RAM

Software

Operating System

The operating system of the computer systems running both the *client* and *server* of FSDAS is Windows XP.

Other platforms

The computer system running the *client* of FSDAS is also equipped with:

- installation of JRE (Java Runtime Edition) version 1.6.0_05

4. Test specifications

The IEEE standard for software testing envisages a collection of documents that were scaled down to this main section of the deliverable. As mentioned in 0 the tests nevertheless verify the technical soundness, completeness and usability of the prototype.

4.1 Technical assessment

This section deals with verifying that both tiers of FSDAS fulfil the requirements collected and translated to an architectural design. We distinguish evaluation of the client and server side.

Server side testing of FSDAS

The server side is a compound of: a reasoner application (OntoBroker) that holds the internal representation of Fishery ontologies, a data storage application (MySQL DBMS) that hosts data generated from the FSDAS and a web server (Apache Tomcat) upon which to deploy the web services providing the query functionalities. Testing focuses on assuring that the three components communicate and that the client connects and functions according to the specification.

Perform installation tests of FSDAS server

The server is delivered as three different applications, each with its own installation wizard. After performing the installation of OntoBroker, MySQL, and Tomcat, the instructions in D7.6.1 suggest how to preset other environmental variables. No extra conditions than the ones specified in the installation document were required. The process finalized successfully and the client connected as expected. A few tests were run to ensure that critical functionalities were responding.

Client side testing of FSDAS

This area is mostly concerned with use case verification, but initial steps insure that client installation follows the procedures described in [D7.6.1](#), considering that hardware and software environmental needs have been satisfied.

Perform installation tests of FSDAS client

FSDAS is packaged and distributed such that the execution of a single file runs the process that opens the main application environment. Hence this phase of the test is not really intended to execute a “wizard” installation process, but is to assure that the environmental needs are in place to guarantee correct running of the software. For this purpose trials were executed on 2 computer systems. Where a system was missing any of the requirements an update/upgrade was executed and the *installation* test ran. The results are reported in the form of templates ([Annex 5](#)) filled with test outcomes reported in [Annex 4](#). Other issues met during the installation phase are reported in the recommendations section as points to improve for the next release.

Perform use-cases test

Following IEEE recommendations on the structure of test specification documentation a template ([Annex 1](#)) was designed that holds all the properties of concern for client testing. The testing phase

was performed on all use cases considered priority and included in the prototype. The results are reported ([Annex 6](#)) in templates containing test outcomes. Other issues met during the usage of the client are reported in the recommendations section as important points to improve for the next release.

4.2 Usability assessment

Three users were selected from the FAO Fisheries and Aquaculture department. They were chosen with a view towards potential users of FSDAS. All of them regularly prepare reports, inventories and website content using fisheries data and reference data (e.g. taxonomies). All of them also had some experience with either programming or mark-up languages.

The users were tested over a period of several days at their regular workstations using the same version of FSDAS (FSDAS2.4+NTK1.1.1-B588). It is important to note that due to the integration of FSDAS within the NeOn toolkit some of the activities in fact test underlying toolkit services and user interfaces. All user test templates used to conduct the testing can be found in [Annex 2](#).

User profile

The user profile showed a group of proficient computer users well-versed in working with XML, structured documentation and metadata and taxonomic standards. Some also had programming and database experience:

background:

- Two users were fisheries information officers and one user works with resources and fisheries inventories.
- All were biologists by education.
- All worked in the Fisheries and Aquaculture Information and Statistics Programme (FIES).
- All were familiar with FAO fisheries taxonomies and classifications.

application experience:

- All users regularly used XML editors and MS-Office, with one also using Access and another DreamWeaver.
- Two users rated themselves as “expert” computer users and a third as “middle”.
- One user had programming experience in Pascal and Cobol and another in XSLT.
- All users were proficient in creating structured documentation with XML and HTML.
- Two users had built web sites.
- One user had created relational databases.

ontology experience

- Two users were able to describe what an ontology was.
- No users had used an ontology editor.

User tasks

All users were given the same set of tasks ([Annex2](#)) and observed as they interacted with the application. Tasks consisted of typical actions such as registering, logging in, importing an ontology, navigating an ontology and querying attached databases. Users were encouraged to state out loud their thoughts on the actions they were performing or thinking of performing. Pertinent remarks were recorded in logbooks by the observer, who also made additional comments based on the observation of their actions.

User post-test questionnaire

Users were given a set of questions ([Annex2](#)) to elicit their overall responses to using the product and gather any suggestions they might have for its improvement. Questions compared FSDAS to other similar applications, asked what was most confusing, what user would change, whether they felt confident about the results, what functionalities seemed missing, error messages, terminology and superfluous features.

5. Test reporting

Based on the tests performed a series of problems emerged that had to do with either functionality that was not delivered, functions that were not correctly implemented or functions delivered but with poor usability.

5.1 Technical test report

It is worth recalling that the use cases under verification are only those planned for the first iteration, ordered by priority.

Covered/uncovered requirements ratio

Of the tests performed to verify the delivery of the 25 use cases planned for the first prototype, only 9 satisfied the *pass* criteria. This had a strong impact on the extent to which features and use cases could be considered to be covered for the release.

Precision of requirement coverage

The use cases that actually satisfied the evaluation criteria passed only by considering no more than the basic principles for their execution; six of them were provided with comments (e.g. UC-3, UC-6 or UC-12) to improve the results when they are delivered in the next release.

Software configuration

The configuration of both halves of the application is straightforward when following the instructions provided with the deliverable D7.6.1. The client execution is as simple as executing a single file; the server installation requires an effort commensurate with other similar server installations.

5.2 Usability test report

This section is grouped by interface widget, wizard or perspective and reports on the difficulties users encountered as they attempted to perform (with some observer guidance) the tasks detailed in [Annex 2](#) which are linked to the more detailed results found in the results of testing [Annex 7](#) and the post-questionnaire [Annex 8](#). Conclusions and recommendations are provided in later sections.

General

- Missing help system.
- Error messages do not offer failure recovery information.
- Searches do not show indication of progress.

Taskbar

- Location and grouping of icons is not intuitive.

- Login does not appear on application launch.
- Logout icon not needed on taskbar.
- Preferences icon confusing.
- Email results and suggest onto modifications not distinguishable.
- Unavailable actions not greyed out.
- Email action location confusing.
- Save session identical to toolkit save.

Menu

- Menu confusing because is a mixture of toolkit and FSDAS menus.
- Grouping of menu items not intuitive.
- Login/logout badly placed.
- Unavailable actions are not greyed out.

Exit

- Exit calls the login screen if a user is not logged in.

Registration

- "Status" field doesn't mean anything to users.
- Users have trouble recalling userName.
- Registration does not email details to user.

Help

- Help enabled but non-existent.

Import wizard

- Terminology confusing.
- Import dialogue "filter" input box does not make sense.
- Web-dav import option confuses users.
- Unclear the relationship between Project creation and ontology import.
- The project creation options box has many confusing options.
- A newly created project is not immediately visible in the project drop-down box of the import dialog.
- Conversion error display makes it seem as if import process is frozen.

Concept instance perspective

- Instances of sub-concepts do not appear when the parent concept is selected.

Ontology visualizer perspective

- When a concept has been selected, opening the visualizer perspective does not automatically display its instances.
- Hovering over concept instances does not display the attributes for that instance

Concept query perspective

- Send buttons are hidden beneath right-most windows depending on screen resolution.
- Searching for concepts is case-sensitive and exact match.
- It is possible to type within the drop-down attribute selection boxes.
- Input box default text is confusing.
- Search by identifier is not useful.
- Scientific name search is missing.
- “F Logic Axiom” query is confusing.
- It is not clear what ontologies users are searching on.
- Filled input fields do not erase when highlighted and delete is pressed.
- “Send” is a confusing label.
- Results displayed by ID is not useful.
- Results are not clickable.

Data query perspective.

- Send buttons are hidden beneath right-most windows depending on screen resolution.
- It is possible to type within the drop-down attribute selection boxes.
- Input box default text is confusing.
- Filled input fields do not erase when highlighted and delete is pressed.
- “Send” is a confusing label.
- Query does not function if Data Sources and Data Query Settings windows are closed.
- “FAO indexes” is a meaningless label.
- Selecting from the DQ1 drop-down boxes does not make them active.
- DQ1 search does not work for attributes other than species.
- Layout of DQ1 and “Send” button is confusing.
- DQ5 attribute selection does not indicate type of code.
- Multiple ranking columns is a waste of screen space.
- Not possible to open the resource result by clicking on the URL.

Email results.

- Icon is badly placed.
- GmailClipse functionality is too heavy.

Bookmark/Favourites

- The labels are not consistent.
- The bookmarks window displays although the feature is not currently implemented.

Save session.

- Icon is badly placed.
- Icon same as toolkit save.

6. Conclusions

Although FSDAS v1.0 suffers from only partial delivery of needed features, an inconsistent policy in developing according to use case prioritization, and numerous functional and usability defects, it is nevertheless a good start to what could become a useful application for the fisheries domain. The basic architecture has been put in place, and a number of basic services integrated. All users tested were generally positive and felt that FSDAS had promise if it were only easier to use, included more resources, resources were clickable and integration between parts of the application were improved.

It is important to note that due to the integration of FSDAS within the NeOn toolkit, a number of the conclusions and recommendations relate not only to FSDAS but also to underlying toolkit services and user interfaces.

The application was quite stable and performant. It suffered from few major bugs and generated a minimum of errors. Test users were able to perform all of the assigned tasks to some degree and often without observer assistance.

In terms of the functionality to be delivered as part of the first version, there was only partial coverage of the expected use cases. In other cases, such as the included Gmailclipse plug-in, the functionality was somewhat orthogonal to what the requirements requested.

6.1 Technical summary

While limited in its coverage of the expected features and use cases, technically the first release of the FSDAS application is a good software delivery. It provides skeletons for both the client and the server tiers and follows basic functional and non-functional requirements needed to support users.

The actual implementation is a first step towards the final version. As such, even the functionalities that are actually included will for the next release require improvement.

Going into specific issues:

- The authentication service needs to be given finer grained parameters on which to distinguish and grant user access; the registration procedure must guarantee that users are indexed uniquely.
- The query interface must lose the current static design in favour of a more flexible data-dependent implementation, and the result section must allow access to returned data instances.
- The ontology handling must guarantee consistency and coherency between client and server (i.e. the ontology navigator and the OntoBroker repository).
- The composition of the query must be flexible enough to allow that all the resources in the ontologies (e.g. concept, individual, relations) can be part of a query formulation.
- The connection layer that links the federation of data repositories should allow to integrate DBMS' other than MySQL and SQLServer.

6.2 Usability summary

Unfortunately, many of the user interface's serious usability defects are due to its deployment as a toolkit feature, an approach that mixes its interfaces with those of the underlying toolkit causing

duplication of menus and buttons as well as providing access to toolkit widgets that have nothing to do with the scope of the FSDAS application. It was exactly concerns such as these that led to the early recommendations from FAO to use NeOn toolkit API's but to not build the application within the Eclipse framework. Resource constraints led to a requirements negotiation in which it was agreed to allow such an implementation. It was perhaps inevitable that borrowing not only the underlying services but much of the GUI from the generic toolkit would result in an application that is a pastiche. This deficiency will require close attention in order to come up with an application that feels unified to the end user.

The following tables take the problems noted during user testing (detailed results in [Annex 7](#)) and formulate them as a series of issues and proposed solutions.

General

Issue	Due to inclusion as NTK feature?	Proposed solution
Missing help system.	No	Help system should be added.
Error messages do not offer failure recovery information.	No	Error messages should offer failure recovery information.
Searches do not show indication of progress.	No	Searches should show a progress bar or other indication that work is happening.

Taskbar

Issue	Due to inclusion as NTK feature?	Proposed solution
Location and grouping of icons is not intuitive.	Yes	Not all NeOn toolkit icons should be shown, only those necessary to FSDAS. They should be organized according to the standards found in reference applications such as IE or Firefox browser or the MS-Office suite.
Login does not appear on application launch.	No	Login should appear automatically on application launch.
Logout icon not needed on taskbar	No	Logout icon should be removed from taskbar. It is not a regular action, is unaccompanied by a login icon and is not disabled when user is not logged in.
Preferences icon confusing	Partially	Preferences icon should not be an envelope image. It should be part of the user profile settings.
Email results and suggest onto modifications not distinguishable	No	Email results and suggest onto modifications should be made more distinguishable.

Unavailable actions not greyed out.	No	Actions that are not available should be greyed out.
Email action location confusing	Yes	Email actions should be moved to the left as in MS-Office suite.
Save session identical to toolkit save	Yes	Save session should not be an identical icon to the toolkit save.

Menu

Issue	Due to inclusion as NTK feature?	Proposed solution
Menu confusing because is a mixture of toolkit and FSDAS menus.	Yes	FSDAS functions and necessary toolkit functions should be integrated. Unneeded toolkit functions should be removed.
Grouping of menu items not intuitive	Yes	Grouping of items should be organized according to the standards found in reference applications such as IE or Firefox browser or the MS-Office suite.
Login/logout badly placed	Yes	Login/logout should appear under left-most file menu and should be disabled depending on user login status.
Unavailable actions are not greyed out	No	Actions that are not available should be greyed out.

Exit

Issue	Due to inclusion as NTK feature?	Proposed solution
Exit calls the login screen if a user is not logged in	No	Exit should not call the login screen if a user is not logged in.

Registration

Issue	Due to inclusion as NTK feature?	Proposed solution
"Status" field doesn't mean anything to users	No	"Status" field should be changed to "Domain" and should be a drop down list provided by FAO.
Users have trouble recalling userName.	No	Auto creation of the userName needs a hard to miss warning to user to record it somewhere.

Registration does not email details to user	No	Registration should email details to user (server side should have a configuration file to add the SMTP host).
---	----	--

Help

Issue	Due to inclusion as NTK feature?	Proposed solution
Help enabled but non-existent	No	Help should be disabled if non-existent

Import wizard

Issue	Due to inclusion as NTK feature?	Proposed solution
Terminology confusing.	Partially	Should be renamed to "Load Ontology" in File menu.
Import dialogue "filter" input box does not make sense	Partially	The Import dialogue "filter" input box should be hidden by default but be accessible via an "advanced options" button.
Web-dav import option confuses users	Partially	Web-dav import should only appear if user clicks an "advanced options" button.
Unclear the relationship between Project creation and ontology import	Partially	Project creation should be clearer. If there is no project users should receive a dialog box that contains only project creation and does not contain also the ontology selection box. The box should have a message saying "You need to create a project before you can load an ontology."
The project creation options box has many confusing options	Partially	The project creation options box should hide the "data model type" and only display it if user clicks an "advanced options" button.
A newly created project is not immediately visible in the project drop-down box of the import dialog.	Partially	When users create a new project the project should be visible in the project drop-down box of the import dialog instead of being blank as at present.
Conversion error display makes it seem as if import process is frozen	Partially	When showing conversion errors it should be clear that the conversion has nevertheless been performed and loaded. Error screen should say

		"Problems were encountered during conversion. Some semantics may have been lost but the conversion has completed."
--	--	--

Concept instance perspective

Issue	Due to inclusion as NTK feature?	Proposed solution
Instances of sub-concepts do not appear when the parent concept is selected	Partially	Instances should appear also when clicking a higher level concept if they are instances of sub concepts.

Ontology visualizer perspective

Issue	Due to inclusion as NTK feature?	Proposed solution
When a concept has been selected, opening the visualizer perspective does not automatically display its instances	Partially	If any concept has been selected, opening the visualizer perspective should display those concepts instead of the blank screen one finds at present.
Hovering over concept instances does not display the attributes for that instance	Partially	Hovering over concept instances should display the attributes for that instance

Concept query perspective

Issue	Due to inclusion as NTK feature?	Proposed solution
Send buttons are hidden beneath right-most windows depending on screen resolution	No	Needs reformatting so send buttons are not hidden beneath right-most windows
Searching for concepts is case-sensitive and exact match	No	Searching for concepts should not be case-sensitive nor exact match. If a search for related concepts is thus linked to several starting concepts the system should allow the user to choose the concept with which they want to search.
It is possible to type within the drop-down attribute	No	It should not be possible to type within the drop-down attribute selection boxes Input boxes should

selection boxes		not have term with quotes around it but should say something informative like "type species here"
Search by identifier is not useful	No	Search by identifier should be removed.
Scientific name search is missing.	No	Scientific name search should be added.
"F Logic Axiom" query is confusing	No	"F Logic Axiom" query should be removed or only accessible via "advanced options" button.
It is not clear what ontologies users are searching on	No	It should be clear what ontologies users are searching on and it should be possible to see those ontologies (i.e. the toolkit import and the FSDAS search must be integrated.)
Filled input fields do not erase when highlighted and delete is pressed	No	Filled input fields should be erasable when highlighted and delete is pressed.
"Send" is a confusing label	No	"Send" should be renamed "Query".
Results displayed by ID is not useful	No	Results should be displayed by name attribute, not ID
Results are not clickable	No	Results should be clickable and open ontology view to the concept instance in the ontology.

Data query perspective.

Issue	Due to inclusion as NTK feature?	Proposed solution
Send buttons are hidden beneath right-most windows depending on screen resolution	No	Needs reformatting so send buttons are not hidden beneath right-most windows
It is possible to type within the drop-down attribute selection boxes	No	It should not be possible to type within the drop-down attribute selection boxes Input boxes should not have term with quotes around it but should say something informative like "type species here"
Input box default text is confusing	No	Filled input fields should be erasable when highlighted and delete is pressed.
Filled input fields do not erase when highlighted and delete is pressed.	No	"Send" should be renamed "Query".

"Send" is a confusing label	No	Query should still function even if Data Sources and Data Query Settings windows are closed.
Query does not function if Data Sources and Data Query Settings windows are closed	No	"FAO indexes" should be labelled "FI Fact sheet indexes".
"FAO indexes" is a meaningless label	No	Selecting from the DQ1 drop-down box should make that box active, i.e. it should not be necessary to click the radio button to activate the attribute.
Selecting from the DQ1 drop-down boxes does not make them active	No	DQ1, search should also work for attributes other than species. Currently Family, group and order do not return results.
DQ1 search does not work for attributes other than species	No	Layout of DQ1 and "Send" button is confusing and should be changed.
DQ5 attribute selection does not indicate type of code	No	DQ5 attribute selection should indicate type of code not just "code"
Multiple ranking columns is a waste of screen space	No	Show just one ranking column in the results.
Not possible to open the resource result by clicking on the URL	No	Make it possible to open the resource result by clicking on the URL (note the index should be modified as displays relative URL)

Email results.

Issue	Due to inclusion as NTK feature?	Proposed solution
Icon is badly placed	Yes	Should be under the File menu at far left.
GmailClipse functionality is too heavy	No	This functionality should be much simpler and not rely on GMailClipse. It is too heavy. The need was simply to have results opening in the user's desktop email client using the mailto: protocol.

Bookmark/Favourites

Issue	Due to inclusion as	Proposed solution
-------	---------------------	-------------------

	NTK feature?	
The labels are not consistent	No	The labels should be consistent for this feature.
The bookmarks window displays although the feature is not currently implemented	No	It should not be possible to open the bookmarks window if this feature is not currently implemented.

Save session.

Issue	Due to inclusion as NTK feature?	Proposed solution
Icon is badly placed	No	Should be under the File menu at far left.
Icon same as toolkit save	No	Should be distinguished from toolkit save.

7. Recommendations

The following recommendations are offered as guidance for the development of FSDAS V2. They are based on the evaluation conclusions as well as the result of discussions and decisions taken during the June 2008 plenary in Dubrovnik. They indicate starting points from which to improve v1.0 but do not supersede or negate the requirements documents [D7.1.1](#) and [D7.1.2](#) nor the architecture document [D7.5.1](#).

The recommendations also touch upon general toolkit functionality that is used by FSDAS. It is hoped that the work packages responsible for these functionalities will cooperate with the FSDAS implementation team to insure a usable v2.0 of the product.

7.1 Use case prioritization

A common aspect to the entire deliverable is that the realization of the use cases didn't follow the priority assigned to them at the time of collecting the requirements. Although it is known that technical matters have hampered a linear development process, it was expected that the choice of what to implement would have been driven by the ranked list provided. It is straightforward to recommend that for the upcoming work, the effort is focused more in the direction identified by the use case priority: first the use cases that have failed to pass the tests, ordered according to priority, then remaining use cases still according to priority.

7.2 Functional improvements

A second recommendation is to improve on the points presented in the technical summary. Each of those points refers concisely to aspects that the developers need to read in detail in the use case test templates in [Annex 6](#) which cover fully the results for each tested case. Developers should also refer to the usability issues reported in section [6.2](#) and in testing results [Annex 7](#) and post-test questionnaire [Annex 8](#). For readability, not every issue is included in this section.

The following action items are grouped by broad functional area.

Interface

- The usability defects noted should be mitigated or repaired.
- Better GUI integration must be achieved between the underlying toolkit and the FSDAS customisation; e.g. by integrating menu and taskbar.
- Better integration between visualised ontologies and the ontologies used for querying must be achieved. The ontologies used should be identical and it should be possible to use the visualised ontologies to assist in query integration.

Sessions

- Simple support for User Profiles that enable the users to customize their workspace and bookmark their queries.

Email

- The Gmailclipse plug-in should be removed and a simpler mechanism put in its place for better alignment with requirements.

Data sources

- Use relational data base in fisheries as alternative data source repository (e.g. ASFA) to test more than the current Lucene indexes.

Queries

- FSDAS needs to sync the ontologies displayed in its scope, with the ones loaded in the server side of the application. Current solution of locally-loaded ontologies causes that users are likely to have loaded ontologies that have nothing to do with what they are searching.
- Support for different configuration settings of the server that reflect user groups on the client side (e.g. preset ontology set). This implies multiple running instances of the server-side (since OntoBroker must load its ontology set when started) and the client able to connect to various instances based on user selection.
- Make the concept and data query perspectives dynamic such that queries are built from attributes parameterized from values taken from the current focused ontology instead of the pre-cooked query panels in current version.

Improve query result display to make it more interlinked, and show provenance of retrieved document with respect to the query executed and the ontologies used to specify it. Currently the ontology screens and the results screens are totally decoupled. Also, current results are not clickable to open attached resource

List of Acronyms used in this deliverable

API	Application Programming Interface
CMM	Capability Maturity Model
CSV	Comma Separated Values
DBMS	Database Management System
FAO	Food and Agriculture Organisation of the United Nations
FI	Fisheries Department, FAO of the UN
FIES	Fisheries and Aquaculture Information and Statistics Programme
FSDAS	Fisheries Stock Depletion Assessment System
GIF	Graphics Interchange Format
HTML	HyperText Markup Language
IEEE	Institute of Electrical and Electronics Engineers
JPEG	Joint Photographic Experts Group
JRE	Java Runtime Edition
OWL	Web Ontology Language
PNG	Portable Network Graphics
RAM	Random Access Memory
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Transformations

References

- [CMM] Capability Maturity Model, <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr011.pdf>
- [D6.1.1] Requirements on NeOn Architecture, <http://www.NeOn-project.org/ACollab/drafting/revisions.php?id=257>
- [D6.1.2] Report on user requirements v2, <http://www.NeOn-project.org/ACollab/drafting/revisions.php?id=862>
- [D6.2.1] Specification of NeOn reference architecture and NeOn APIs, <http://www.NeOn-project.org/ACollab/drafting/revisions.php?id=489>
- [D7.1.1] FSDAS Requirements, http://www.NeOn-project.org/ACollab/get_file.php?id=475
- [D7.1.2] FSDAS Revised Requirements, http://www.NeOn-project.org/ACollab/get_file.php?id=1152
- [D7.2.1] Inventory of Fisheries Resources and Systems, http://www.NeOn-project.org/ACollab/get_file.php?id=709
- [D7.2.2] Fisheries Ontologies, http://www.NeOn-project.org/ACollab/get_file.php?id=1017
- [D7.5.1] FSDAS Architecture, http://www.NeOn-project.org/ACollab/get_file.php?id=1019
- [D7.6.1] FSDAS v1, <http://secse.atosorigin.es:10000/FSDAS-Client/FSDAS2.4+NTK1.1.1-B588.zip>
- [IEEE] IEEE (1998) Standard for software test documentation, IEEE Computer Society.
- [UID] Stone D., Jarrett C., Woodroffe M., and Minocha S. (2005) User Interface Design and Evaluation, Morgan Kaufmann.
- [UIE] Open University (2001) M873 User Interface Design and Evaluation: Evaluation, Milton Keynes, Open University.

Annex 1 – Template report for the use cases testing

Use Case Identifier	
Use Case Description	
Title	
Requirements specification	
Design specification	
Input Specification	
Output Specification	
Special procedural requirements	
Intercase dependencies	
Pass/Fail	
Special Criteria	
Result	
Comments	

Template fields description

1. Use Case Identifier: the sequential numbering given to use cases since D7.1.2
2. Use Case Description
 - a) Title: the name associated with the use case number
 - b) Requirement Specification: the excerpt(s) from the user requirements list that the use case is aimed to cover
 - c) Design Specification: the pointer to the component in the architectural design that is aimed at embedding the functionality provided by the use case test.
3. Input specification: the input used in the testing phase
4. Output specification: the output produced in correspondence to the input specification
5. Special procedural requirements: specification of any particular preparation for an optimal execution of the use case.
6. Intercase dependencies: dependency of this test from another in the list
7. Pass/Fail
 - a) Special criteria: the condition(s) that is(are) not explicit when commonly understanding the test
 - b) Result: the actual response of the test PASSED/FAILED

- c) Comments: the list of comments that are related to the execution of the test; they can include good and bad points about the use case.

Annex 2 – Usability testing templates

FSDAS User test instructions

Dear John Doe,

Thank you for agreeing to be part of the user testing of the first prototype of the Fish Stock Depletion Assessment System (FSDAS). I'll be reading from a script to insure that each user session is conducted in a similar way.

The purpose of this user session is to observe users interacting with the system using a pre-prepared set of tasks in order to help us evaluate it. This user session is not meant to test or grade your skills as a computer user in any way so you should not feel embarrassed or under pressure to perform or answer any questions in a particular way.

This session will consist of the following activities:

- A brief survey on your background.
- Installation of FSDAS.
- Description of the purpose of FSDAS.
- A set of tasks to evaluate the current usability of FSDAS.
- A post-questionnaire to evaluate the current usability of FSDAS.

Let's go ahead and begin with the survey.

[Administer survey]

Now that we've completed the survey let's launch FSDAS.

[Have user navigate to network folder containing FSDAS and have them launch FSDAS.exe.]

Now that FSDAS has launched let me give you some background on its purpose. FSDAS is a system that uses fisheries concepts and the relationships between those concepts to perform searches for fisheries resources coming from various fisheries information systems. This is just the first prototype so it uses a limited set of concepts and only one information source – FIGIS fact sheets. In the future it will use more concepts and access more databases.

I will ask you to perform a series of tasks. During each task I'll record your remarks and some information about your experiences.

[Follow task script and complete logbook for each task as indicated]

Now that you've performed the tasks, I would like to ask you a series of questions about the experience.

[Administer post-questionnaire]

That's it. Thank you for your time.

FSDAS User Survey	userID _____
 background	
1. What is the official title of your position at work?	

2. In what fisheries domains do you have experience?	

3. In what fisheries domains do you currently work?	

4. Are you familiar with fisheries reference data, classification systems, etc.? If so, which ones?	

application experience	
5. What applications do you commonly use for work?	

6. How would you rate your level of computer experience?	

7. Do you know any programming languages?	

8. Do you know any mark-up languages?	

9. Have you ever built a web site?	

10. Have you ever created a relational database?	

ontology experience	
11. Describe what an ontology is:	

12. Have you ever used an ontology editor such as Protégé?	

FSDAS Task Script

1. Launch FSDAS.exe
2. Find “login” and select it.
3. Register as a new user.
4. Login.
5. Find “import” and import the vessel ontology file **vessels_v2.0.owl** (found in subfolder **fao_ontologies** of FSDAS application folder) as an F-Logic ontology.
6. In the imported ontology find the concept instance “trawler”.
7. Find and visualize the “by_type” concept instances using the visualizer.
8. Select the “concept query” perspective.
9. Use the concept query search interface to generate results.
10. Bookmark the concept results.
11. Select the “data query” perspective.
12. Use the data query search interface to generate results.
13. Email results.
14. Save session.
15. Exit.

FSDAS Logbook

Userid			
Place	Work		
Role of computer	Work		
Date	2008,		
Operating system	XP		

Task description no.:		Screen name:
Script	User's remarks	Observer's comments:
(when screen appears) <ul style="list-style-type: none"> • Do you know what to do next? • Do you recognise what to do, or did you have to ask what to do? • What are you trying to do? • Looking at the system response, do you think you made the correct choice? 		

Post-user test questionnaire	UserId:_____
1. Compared to other browsing / searching applications what are your general comments when comparing FSDAS?	
2. What aspects of the interface were the most confusing?	
3. What aspects would you change?	
4. Did you feel confident of the validity of the results?	
5. What functionalities or views seemed missing?	
6. If you received any error messages, did they help you to interpret the problem and resolve it?	
7. Was the terminology used in the actions familiar to you? Did it follow the conventions of other workflows you have experienced? If not, what was different?	
8. Did you ever feel lost while performing the tasks? When and where?	
9. Do you feel the system could assist you in your regular work tasks?	
10. Were there features that seemed superfluous?	

Annex 3: List of FSDAS use cases

Use Case	High Priority	Medium Priority	Low Priority	Second Iteration
1 Login	x			
2 Logout	x			
3 Register	x			
4 Modify Profile		x		
5 Modify User Account	x			
6 Search ontological resource in ontology	x			
7 Search for related ontological resources	x			
8 Browse Taxonomy	x			
9 Change language of the interface	x			
10 Change language of the ontological resource shown	x			
11 Query Composition	x			
12 Query for Data related to individual	x			
13 Visualize Data Source related to individual	x			
14 Refine query	x			
15 View ontological resource annotation	x			
16 View Data Source Annotation			x	
17 Email results			x	
18 Propose ontology modification		x		
19 Add to favourites		x		
20 Save session		x		
21 Generate RSS feed from current query				x
22 Annotate retrieved document with comments on quality			x	
23 Index enrichment against ontology(ies) domain model			x	
24 Annotate retrieved document with comments			x	
25 Select ontologies to use for browsing documents or web pages				x
26 Use ontology to support browsing				x
27 Identify trend				x
28 Compare data by reporter				x
29 Discover ontology				x
30 Filter ontology				x
31 View data instance summary				x
32 Re-group/rank data resource by annotation				x

Annex 4 – Template report for the FSDAS client installation

PC type	
Software equipment	
Operation system	
Java runtime environment	
Hardware equipment	
Processor on board	
RAM on board	
Pass/Fail	
Result	
Comments	

Template fields description

1. Pc type: desktop or laptop computer system
2. Software equipment
 - a) Operating system: this is currently singular valued, i.e. Windows XP; distribution for other OS are under development
 - b) Java runtime environment: the version of the JRE currently resident on the computer system
3. Hardware equipment
 - a) Processor on board: the CPU frequency powering the computer system
 - b) RAM on board: the amount of random access memory powering the computer system
4. Pass/Fail
 - a) Result: the actual response of the test PASSED/FAILED
 - b) Comments: upgrade/update of any of the above items.

Annex 5 – Result of FSDAS client installation

PC type	Desktop
Software equipment	
Operation system	Windows XP
Java runtime environment	1.6.0_06
Hardware equipment	
Processor on board	Intel Pentium4 3.20Ghz
RAM on board	1 gigabyte
Pass/Fail	
Result	Passed
Comments	a) This installation needed that the JRE was updated to latest release; although the application worked fine also with previous installation of JRE 1.6.0_03

PC type	Laptop
Software equipment	
Operation system	Windows XP
Java runtime environment	1.6.0_06
Hardware equipment	
Processor on board	Intel Pentium Mobile 2.00 GHz
RAM on board	1 gigabyte
Pass/Fail	
Result	Passed
Comments	a) This installation needed the set up of proxy parameters; after specifying proxy address and port the login feature returned to work correctly

Annex 6 – Result of FSDAS use cases test

Use Case Identifier	UC-1
Use Case Description	
Title	Login
Requirements specification	Users shall be able to [...] login [...]
Design specification	This functionality belongs to UserAccountManager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
User input	Username: NeOn Password: test
Output Specification	
Access granted	True
Special procedural requirements	
Set proxy parameters	If login procedure doesn't respond user should check the settings to proxy the connection to FSDAS server.
Intercase dependencies	
n/a	n/a
Pass/Fail	
Special Criteria	a) Component should not grant access to unregistered users, or user with wrong password.
Result	Passed
Comments	<p>b) Component was tested with an unregistered username, and a misspelled password for a registered user. In both cases system didn't grant the access.</p> <p>c) The login component can distinguish if the error is on username or password; another valuable point is that the password is case sensitive.</p>

Use Case Identifier	UC-2
Use Case Description	
Title	Logout
Requirements specification	n/a
Design specification	This functionality belongs to UserAccountManager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
GUI interaction	Mouse click on "Logout" button in the top bar
Output Specification	
Screen notification	User is notified of successful logout with a new login mask
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Passed
Comments	a) According late modification of this use case (see D7.1.2 table17), the user should be asked to save his/her working context. At this stage this is not yet happening; the actual version simply presents a new login mask upon user logout.

Use Case Identifier	UC-3
Use Case Description	
Title	Register
Requirements specification	Users shall be able to register,[...]. Users shall be able to join a particular user type, e.g. biologist, economist etc.
Design specification	This functionality belongs to UserAccountManager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
User input	Details of registration: Username: <i>Claudio</i> Surname: <i>Baldassarre</i> email: tester@fao.org Status: <i>tester</i> Password: <i>pass</i>
Output Specification	
Screen notification	User is notified of successful registration with an info message reporting his/her username: <i>Claudio</i>
Special procedural requirements	
n/a	n/a
Intercase dependencies	
n/a	n/a
Pass/Fail	
Special Criteria	<ul style="list-style-type: none"> a) Component should distinguish the registered users based on a unique attribute of registration, and do not allow other user to enter duplicates for that value. b) The component should apply a mask to attribute field like "Email" to verify that the user input is valid for that box. c) The role of the registering user must be assigned by a member of the FSDAS administration team
Result	Passed
Comments	<ul style="list-style-type: none"> a) If two users input same name, surname and email value the logic of the component overrides the password of the first of them who executed registration. This can be used by malicious users who can take control of other users' account. b) The logic of the component should consider the email as the only unique attribute to distinguish two users, and inform that another user with the same email already exists to prevent overriding. c) The logic of the component should apply a mask to validate the input of registration fields like the "Email". At the moment a valid email value can be any string not necessary in the form of xxx@yyy.zz d) According to late modification of this use case (see D7.1.2 table17), the registration request is sent for action to an FSDAS administrators team member. At this stage this is not yet happening. e) According to late modification of this use case (see D7.1.2 table17), the registering user is assigned with a role from an FSDAS administrators team member. At this stage this is not yet happening. f) According to late modification of this use case (see D7.1.2 table17), the registering user can also specify other profile attributes e.g. organization, preferred languages, ontologies, etc. At this stage this is not yet happening.

Use Case Identifier	UC-4
Use Case Description	
Title	Modify profile
Requirements specification	Users shall be able to [...] maintain a profile
Design specification	This functionality belongs to UserAccountManager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
User input	n/a
Output Specification	
Screen notification	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Register
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) This use case is not yet implemented; it is set as medium priority and to be realized for first prototype. b) The main interface is provided with a button to get access to this functionality once it was implemented

Use Case Identifier	UC-5
Use Case Description	
Title	Modify User Account
Requirements specification	Users shall be able to [...] maintain a profile
Design specification	This functionality belongs to UserAccountManager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
User input	n/a
Output Specification	
Screen notification	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-3	Register
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	a) This use case is not yet implemented; it is set as high priority and to be realized for first prototype.

Use Case Identifier	UC-6
Use Case Description	
Title	Search ontological resource in ontology
Requirements specification	Users shall be able to search (look up) for ontological elements in all the ontologies loaded in the system
Design specification	This functionality belongs to NTK core, provided by the <i>com.ontoprise.ontostudio.search</i> package
Input Specification	
User input	Local name (or substring) of the element to look up: <i>land</i>
Output Specification	
Screen display of query result	List of ontological elements (Concept, Property, Instance) that match the user query
Special procedural requirements	
Ontology in to the workspace	User needs first to load the ontologies on which to perform the search in to the workspace
Query specification	User can specify the type of ontological elements to include in the result set (Concept and/or Property and/or Instance)
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Passed
Comments	<ul style="list-style-type: none"> a) User should be able to specify if search is case Case-sensitive or not; actually this option is greyed in the search panel b) Search panel reports a tab with title "Search File" which is out of scope; this must be removed c) Items in the result set are not enable to directly link to the element in their defining ontology

Use Case Identifier	UC-7
Use Case Description	
Title	Search for related ontological resources
Requirements specification	Users shall be able to input ad-hoc queries, both using free-text and by highlighting concepts/relationships suggested by the currently loaded ontology set that shall return either related data instances or related concepts/relationships found in the currently loaded ontology set
Design specification	This functionality belongs to Ontology Query Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
User input	Combination of ontological elements specified as local name string value related by an Subject-Relation-Object relationship e.g. get species X in the same group as specie Y: <i>For species with "identifier" "10002.0" return the other species in its "Species Group"</i>
Output Specification	
Screen display of query result	List of ontological elements (Concept, Property, Instance) and ontology provenance, that match the user query: 31005_10058 from species.owl 31005_10055 from species.owl 31005_10054 from species.owl 31005_10050 from species.owl ...
Special procedural requirements	
Enable the right work perspective	User need to open the "FSDAS Concept Query" perspective (panel) where s/he finds the appropriate widgets to perform the searches
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Passed
Comments	<ul style="list-style-type: none"> a) All the query examples proposed in D7.6.1 have been tested and the application responded with no errors. b) There is no mechanism to narrow on the ontologies target of the search from the one stored in the repository c) Items in the result set are not enable to directly link to the element in their defining ontology d) The actual shape of the "Concept Query Perspective" is rather static; for next release of FSDAS it is planned that a form of dynamicity will allow the user to have more degree of freedom in composing his/her query. A dynamic interface will display the values for the relation, subject, and object according the loaded ontologies in the work space, e.g. loading the "Species" ontology would give search boxes for <i>species</i>, <i>family</i>, <i>order</i>, with auto-complete boxes. For networked ontologies like "Species" and "Water Areas" user would then see the "swims in" relation and then selections for water area.

Use Case Identifier	UC-8
Use Case Description	
Title	Browse Taxonomy
Requirements specification	The system shall allow authorized users to browse fisheries concepts, attributes and relations. The system shall support several modes of viewing ontologies: <ul style="list-style-type: none"> - Rubber-band - Hierarchical
Design specification	This functionality belongs to Ontological Display Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
Enable the right work perspective	User need to open the "FSDAS Visualization" perspective (panel) where s/he finds the appropriate widgets to perform the visualization
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Passed
Comments	a) The application offers two ways of browsing the ontology: (i) by the "Rubber band" view, and (ii) a classical tree view in the "Ontology Navigator" panel that is proper of NTK core.

Use Case Identifier	UC-9
Use Case Description	
Title	Change Language of the interface
Requirements specification	Users shall be able to view all concepts, relations [...] in the five languages of FAO, when available.
Design specification	This functionality belongs to Display Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	a) This use case is not yet implemented; it is set as high priority and to be realized for first prototype.

Use Case Identifier	UC-10
Use Case Description	
Title	Change Language of the ontological resource shown
Requirements specification	Users shall be able to [...] the application interface in the five languages of FAO, when available.
Design specification	This functionality belongs to Ontological Display Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	a) This use case is not yet implemented; it is set as high priority and to be realized for first prototype.

Use Case Identifier	UC-11
Use Case Description	
Title	Query composition
Requirements specification	See UC-7
Design specification	See UC-7
Input Specification	
See UC-7	See UC-7
Output Specification	
See UC-7	See UC-7
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Passed
Comments	a) The actual shape of the "Concept Query Perspective" is rather static; for next release of FSDAS it is planned that a form of dynamicity will allow the user to have more degree of freedom in composing his/her query. A dynamic interface will display the values for the relation, subject, and object according the loaded ontologies in the work space, e.g. loading the "Species" ontology would give search boxes for species, family, order, with auto-complete boxes. For networked ontologies like "Species" and "Water Areas" user would then see the "swims in" relation and then selections for water area.

Use Case Identifier	UC-12
Use Case Description	
Title	Query for data related to individual
Requirements specification	Users shall be able to view the digital resources related to concept instances. Users shall be able to select to rank results based on the rankings filtered by their user type or other annotation data. Data instances returned by a query shall be grouped according to their related concepts and relationships
Design specification	This functionality belongs to Ontology Query Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
User input	User specify WHERE clause by selecting from combination of ontological elements as clause parameters, and narrowing the scope to specific <i>type</i> of concepts: <i>get data resources FOR "group" WHERE common name is shark</i>
Output Specification	
Screen display of query result	List of path to data resource (documents) that match the user query
Special procedural requirements	
Enable the right work perspective	User need to open the "FSDAS Data Query" perspective (panel) where s/he finds the appropriate widgets to perform the searches
Check the sources target of the query	User need to check the repository that was target of query execution: <i>FAO indexes of RTMS DB</i>
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Passed
Comments	<p>a) The actual shape of the "Data Query Perspective" is rather static; for next release of FSDAS it is planned that a form of dynamicity will allow the user to have more degree of freedom in composing his/her query. A dynamic interface will display the values for the relation, subject, and object according the loaded ontologies in the work space, e.g. loading the "Species" ontology would give possibilities to narrow down scope pf search FOR "species", "family", "order", with auto-complete boxes. The WHERE clause also benefit from dynamically load the values of the boxes.</p> <p>b) Better integration is needed between concept view and data instance view. It should be possible to view an ontological instance, and its associated data instances, and be able to view this in some kind of tree or graph view, that is navigable.</p> <p>c) For queries that span ontologies and domains, would be nice to know for each results what caused them to be selected by the reasoner, what domain or system they come from and what ontologies were involved.</p> <p>d) Items in the result set are not enable to directly link to the element in their defining ontology</p> <p>e) The result set cannot be arranged by clicking on the column heading e.g. <i>Ranking, Last modified, Abstract</i></p> <p>f) The format for the ranking value is misleading because it takes the floating representation with exponents such that values far bigger then one are reported as over the unit: e.g. 3.654543E-4.</p> <p>g) According to late modification of this use case (see D7.1.2 table 17) the system should display related cluster of documents to which a data resource belongs</p>

Use Case Identifier	UC-13
Use Case Description	
Title	Visualize data source related to individual
Requirements specification	Data instances linked to proprietary applications such as DOC and PDF are URL link
Design specification	This functionality belongs to Data Source Display Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	User need to open the "FSDAS Visualization" perspective (panel) where s/he finds the appropriate widgets to perform the visualization
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) Items in the result set are not enable to directly link to the element in their defining ontology b) This use case is not yet implemented; it is set as high priority and to be realized for first prototype.

Use Case Identifier	UC-14
Use Case Description	
Title	Refine query
Requirements specification	See UC-7, UC-12
Design specification	See UC-7, UC-12
Input Specification	
See UC-7, UC-12	See UC-7, UC-12
Output Specification	
See UC-7, UC-12	See UC-7, UC-12
Special procedural requirements	
See UC-7, UC-12	See UC-7, UC-12
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	Re-perform both type of search narrowing or widening search scope
Result	Passed
Comments	a) The execution of this use case is possible by changing the value of the search field in the perspectives (panels) of both types of searches.

Use Case Identifier	UC-15
Use Case Description	
Title	View ontological resource annotation
Requirements specification	n/a
Design specification	This functionality belongs to Annotation Manager and Ontological Resource Display Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	a) This use case is not yet implemented; it is set as high priority and to be realized for first prototype.

Use Case Identifier	UC-16
Use Case Description	
Title	View data source Annotation
Requirements specification	Data instances shall be presented within their associated ontological context and associated metadata crucial to understanding shall also be displayed.
Design specification	This functionality belongs to Annotation Manager and Data Source Display Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) This use case is not yet implemented; it is set as low priority and to be realized for first prototype. b) The main interface is provided with a button to get access to this functionality once it was implemented

Use Case Identifier	UC-17
Use Case Description	
Title	Email results
Requirements specification	Users shall be able to email formatted results.
Design specification	This functionality belongs to Communication Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) The integrated email client requires an existing account to Gmail that is out of requirements; it is more suitable that FSDAS users can use their existing email account. b) This use case is not yet implemented; it is set as low priority and to be realized for first prototype. c) The main interface is provided with a button to get access to this functionality once it was implemented

Use Case Identifier	UC-18
Use Case Description	
Title	Propose ontology modification
Requirements specification	Users shall be able to suggest ontology modifications that was forwarded to the ontology owner via email.
Design specification	n/a
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) This use case is not yet implemented; it is set as medium priority and to be realized for first prototype. b) The main interface is provided with a button to get access to this functionality once it was implemented

Use Case Identifier	UC-19
Use Case Description	
Title	Add to favourites
Requirements specification	Users shall be able to bookmark and access favourite queries, ontologies, ontology concepts, default language and ontology subsets
Design specification	This functionality belongs to User Profile Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) This use case is not yet implemented; it is set as medium priority and to be realized for first prototype. b) The main interface is provided with a button to get access to this functionality once it was implemented

Use Case Identifier	UC-20
Use Case Description	
Title	Save session
Requirements specification	n/a
Design specification	This functionality belongs to Session Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	a) This use case is not yet implemented; it is set as medium priority and to be realized for first prototype.

Use Case Identifier	UC-22
Use Case Description	
Title	Annotate retrieved document with comments on quality
Requirements specification	Users shall be able to annotate (including [...] quality scores) and rate concepts, attributes, relations and associated data instances.
Design specification	This functionality belongs to Annotation Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) This use case is not yet implemented; it is set as medium priority and to be realized for first prototype. b) The main interface is provided with a button to get access to this functionality once it was implemented

Use Case Identifier	UC-23
Use Case Description	
Title	Index enrichment with keywords
Requirements specification	Users shall be able to annotate (including [...] keywords [...]) and rate concepts, attributes, relations and associated data instances.
Design specification	This functionality belongs to Annotation Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) This use case is not yet implemented; it is set as medium priority and to be realized for first prototype. b) The main interface is provided with a button to get access to this functionality once it was implemented

Use Case Identifier	UC-24
Use Case Description	
Title	Annotate retrieved document with comments
Requirements specification	Users shall be able to annotate (including comments, [...]) and rate concepts, attributes, relations and associated data instances.
Design specification	This functionality belongs to Annotation Manager component that is not part of the NTK core but was designed and implemented for the FSDAS application.
Input Specification	
n/a	n/a
Output Specification	
n/a	n/a
Special procedural requirements	
n/a	n/a
Intercase dependencies	
UC-1	Login
Pass/Fail	
Special Criteria	n/a
Result	Failed
Comments	<ul style="list-style-type: none"> a) This use case is not yet implemented; it is set as medium priority and to be realized for first prototype. b) The main interface is provided with a button to get access to this functionality once it was implemented

Annex 7 – Usability testing task summary

Launch FSDAS.exe

- Some insignificant problems with workspace errors since multiple users were accessing the same FSDAS instance in a network folder.

Find “login” and select it. Register as a new user. Login.

- Users had difficulty finding login. Thought it was strange that there was a logout button but no login. Thought it was strange to find it under menu “FSDAS File.” One asked if it couldn’t be automatic on application load. Another said it was strange to have “logout” in the middle of the toolbar.
- One user closed the application before logging in, and got a login screen.
- Registration “Status” field was very confusing. No one knew what it meant. All had to be helped.
- Users tried the help button but didn’t function.
- Users were confused by the auto-creation of the userName. Two out of three were unable to log-in as they hadn’t noticed the creation of userName and hence could not log-in. Had to be helped.
- Two users thought it strange they had to log-in after registering.

Find “import” and import the vessel ontology file vessels_v2.0.owl (found in subfolder fao_ontologies of FSDAS application folder) as an F-Logic ontology.

- Users had some difficulty finding the “import” function. They tended to look in the FSDAS menus. One had to be helped.
- Were confused by the need to create a project first.
- Did not understand project creation options. All had to be helped.
- Upon return project was not listed by default in project drop-down. Confused 2 users.
- Did not know whether to select file system or web-dav for import. Two had to be helped.
- Upon import there were some OWL-F-Logic conversion errors. They didn’t know whether to click “OK” or not. Error screen was not very informative. All had to be helped.

In the imported ontology find the concept instance “trawler”.

- All users began by clicking on the concepts and then scanning the textual view that appears by default in the centre perspective.
- Two users had little difficulty finding the instance, while one user clicked the top-level “Vessel type” concept and not seeing any changes to the screen, began hunting around in file menus and toolbar icons. Eventually they returned to the sub-concepts and were able to visualize and select instances.

Find and visualize the “by_type” concept instances using the visualizer.

- All users followed exactly the same path. They selected the “by_type” concept and then began hunting in menus and toolbar. They all found the “graphic perspective” which when clicked gave a blank visualizer page. They then all began hunting everywhere for some other possibility. None were able to accomplish the task without help (requires a right-click on the concept and the selection of “view in visualizer” from the right-click menu).

Select the “concept query” perspective.

- One user tried to right click on the concept again since this called for the visualizer perspective in the previous task
- One user noted the “Q” on the query icon was helpful in identification.

Use the concept query search interface to generate results.

- For all users, the send buttons for the query perspective were hidden beneath the right-most windows due to their lower screen resolution.
- No user was able to get a result set without being explained that the matching was exact match and case-sensitive. They all asked why they couldn't have a “contains” search that was not case-sensitive.
- One user typed in the drop-down boxes containing the attribute values.
- One user thought they should put quotes around the terms since the default values have quotes. General feeling that the filled fields were confusing.
- All users noted that search by Identifier was useless, and that scientific name search was missing.
- Confused by top query “F Logic Axiom”.
- Two users asked what ontologies they were actually searching on and how they could see them.
- A user noted that filled fields did not erase when highlighted and delete is pressed.
- A user noted that “Send” was a confusing title.
- Users tried to click results and were dismayed that they didn't open in the same kind of ontological view they had seen earlier. The fact that they displayed using the Identifier made it difficult to identify them.

Bookmark the concept results.

- All users found either the bookmark or the favourites but were confused as to why there were both and questioned whether they had different functionality.
- Selecting the button opens a window that is empty and users tried to manually add things to it.
- One user selected a result and clicked bookmark and expected that result to be bookmarked.

Select the “data query” perspective.

- Some hunting for the perspective but nothing serious. One user opened it by using the top right “switch perspective” option.

Use the data query search interface to generate results.

- For all users, the send buttons for the query perspective were hidden beneath the right-most windows due to their lower screen resolution.
- One user closed the right-most windows showing the data sources used and the ranking options. This caused no error but the queries did not work at all and seem dependent on these windows being open.
- A user noted that “FAO indexes” should be “FI Fact sheet indexes”.
- All users found it strange that for DQ1 they had to not only select from the drop-down box but also click the radio button to activate the attribute. This caused no results for one user. Another user selected both fields and then typed something in the input box and got no results as they had not clicked a radio button.
- For DQ1, only the search by species works. Family, group and order do not return results.
- All users found it confusing that the topmost “Send” button was for what appeared to be two queries.
- A user questioned why they had both ranking and ranking code in the results. Seemed redundant. The same user also noted that the ranking was more accurate in the current web search engine that uses the same index.
- Users were dismayed they could not actually open the resource result by clicking on the URL (also noted index should be modified as is a relative URL)

Email results.

- All 3 users mistakenly selected the “preferences” icon.
- Two users noted there were 3 icons that all had envelopes and looked very similar.
- Two users indicated they expected it under the File menu all the way to the left, or that the icon should be all the way to the left as it is in MS-Office suite.
- Two users indicated they would never put their Gmail password into an application they knew little about.
- All users indicated the interface seemed enormously complex when they just wanted to email some results.
- Two users suggested they would simply like the application to open their desktop email client with the links to the resources that were in the result set.

Save session.

- All users went immediately to the File menu and then secondly clicked the “Save” icon that is part of the overall toolkit. They then generally found the correct icon but noted it was identical to the regular Save.

Exit.

- No problems noted.

Annex 8 – Usability testing post-task questionnaire summary

- Location and grouping of buttons is not intuitive.
- It feels like there is a duplication of menus and menu titles are vague.
- Online help is very much needed.
- Panels often do not seem inter-related.
- Search interface crowded and not intuitive.
- Results seem accurate but can't be opened for verification or use.
- Confusing to have icons same as one is used to but that trigger a different action.
- Error messages did not offer help on how to recover from the failure.
- During some slow responses on searches, there was no indication that progress was happening.
- The visualizer is pretty, but useless.
- The application has promise if it were easier to use, included more resources, resources were clickable and integration between parts was better.