Fastaxon User Manual

T-76.115
Group: Muuntaja

Version History

Owner of the document: Quality Manager Pekka Uusitalo
Acceptor of the document: Esko Simpanen

Version	Date	Author(s)	Status	Description
0.1	9.3.2004	Pekka Uusitalo	Dreft	Initial version
1.0	15.3.2004	Esko Simpanen	Accepted	Released version

Contents

1 Introduction

Welcome to the FASTAXON - easy to use, fast and reliable software for building the big, scalable and flexible taxonomies - like you have dreamed.

You'll find that FASTAXON is the tool that will make the creation and management of taxonomies - and taxonomy based catalogues look like chid play.

1.1 About this Manual

This document describes the basic features of FASTAXON software. In chapter 2 there is a short introduction to the terminology and the scientific theory behind the FASTAXON software. In chapter 3 a quick overview of a FASTAXON system is presented. Chapter 4 describes the designer interface and it's functionalities. Chapter 5 is dedicated to the Expression Builder and applying Compound Term Composition Algebra to the FASTAXON project. In chapter 6 the Object Indexer will be described. Chapter 7 covers the using of the end-user interface. Installation and Administration issues can be found from chapter 8 and a short troubleshooting section is in chapter 9.

1.2 Technical requirements

The FASTAXON software runs only on Microsoft Windows 2000 or higher. It is tested with MySQL 1.4 database server and Microsoft Internet Explorer 6.0.

1.3 How I can get FASTAXON?

FASTAXON will be published under VTT public licence.

1.4 Acknowledgments

FASTAXON was implemented by

- Mika Hakkarainen,
- Pekka Korhonen,
- Tero Leppanen,
- Esko Simpanen,
- Hannu Tornroos,
- Pekka Uusitalo,
- Pentti Vanska.

as a student project for the course Software Project (http://www.soberit.hut.fi/T-76.115) (2003-2004) of the Helsinki University of Technology (http://www.hut.fi). The client of this project was Raimo Launonen from VTT Information Technology (VTT Technical Research Centre of Finland, http://www.vtt.fi/tte/indexe.htm). The technical advisor of this project was Yannis Tzitzikas (http://www.csi.forth.gr/~tzitzik).

2 Introduction to FASTAXON

2.1 What is FASTAXON?

FASTAXON is a system for building very big taxonomies and taxonomy-based catalogs in a quick, flexible and scalable manner.

Specifically, FASTAXON allows

- creating and updating faceted taxonomies,
- specifying the meaningful compound terms of a faceted taxonomy,
- creating and browsing taxonomy-based Catalogs

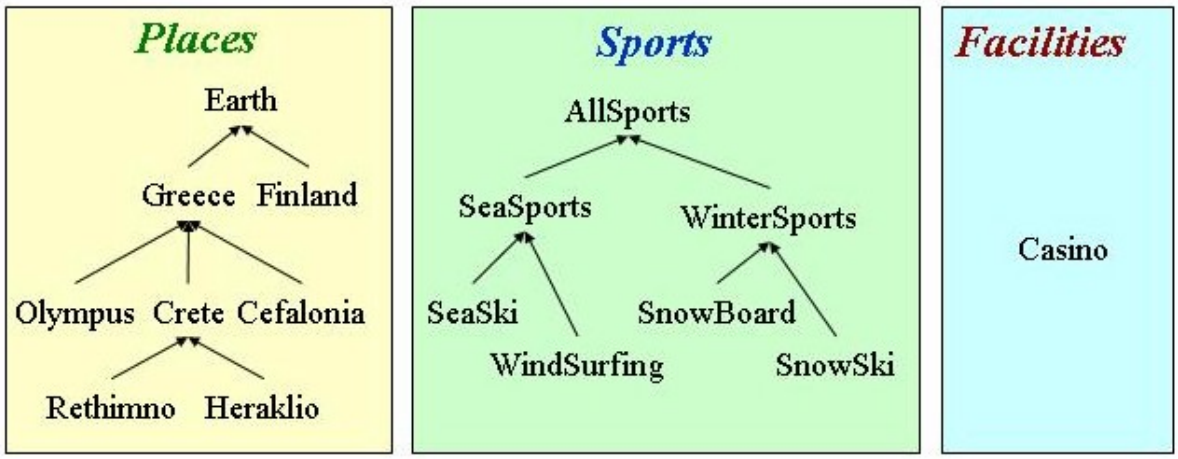
The functionality of FASTAXON is based on *faceted taxonomies* and the *Compound Term Composition Algebra (CTCA)*.

2.2 What is a *Faceted Taxonomy?*

Roughly, a *taxonomy* is a hierarchically-organized set of terms. *Taxonomies* is probably the oldest and most widely used conceptual modelling tool still used is in Web directories (e.g. in Google and Yahoo!), Content Management (hierarchical structures are used to classify documents), Web Publishing (many authoring tools require to organize the contents of portals according to some hierarchical structure), Web Services (services are typically classified in a hierarchical form), Marketplaces (goods are classified in hierarchical catalogs), Personal File Systems, Personal Bookmarks for the Web, and Libraries (e.g. Thesauri).

In designing a taxonomy, one has to define (a priori) appropriate terms and their subterms, However, as pointed out long ago by Ranganathan in the 1920s, the design of a taxonomy can be done in a more convenient and a more systematic manner, if we first identify a number of different aspects of the domain and then design one taxonomy per aspect. This process results in a *faceted taxonomy*, i.e. a *set* of taxonomies, called *facets*.

For example, suppose that we want to index Hotel home pages and suppose that we want to provide access to these pages according to three facets, (a) the *Place* of the hotels, (b) the *Sports* that are possible in these hotels, and (c) the *Facilities* that they offer. The next figure shows a faceted taxonomy.



Picture 1. Faceted taxonomies.

Each object (here hotel home page) can now be indexed using a *compound term*, i.e. a set of terms containing one or more terms from each facet. For example, a hotel in Crete providing sea ski and wind-surfing facilities would be indexed by the compound term {Crete, SeaSki, Windsurfing}.

2.3 What is the *Compound Term Composition Algebra* (CTCA)?

Notice that several compound terms over the above faceted taxonomy are *meaningless*. For instance, we cannot do any winter sport in the Greek islands (Crete and Cefalonia) as they never have enough snow, and we cannot do any sea sport in Olympus because Olympus is a mountain.

Meaningless compound terms cause serious problems in indexing and browsing that prevent the design and deployment of faceted taxonomies for real and large scale applications. On the other hand the ability to infer the *meaningful* compound terms of a faceted taxonomy would be very useful. It could be exploited in the indexing process in order to aid the indexer and prevent indexing errors. Such an aid is especially important in cases where the indexing is done by many people who are not domain experts. For example, the indexing of Web pages in the Open Directory (which is used by Netscape, Lycos, HotBot and several other search engines) is done by more than 20.000 volunteer human editors (indexers). In practice, if we could infer the meaningful compound terms in a faceted taxonomy then we would be able to generate navigation trees *on the fly*, having only meaningful compound terms as nodes.

Consider again the faceted taxonomy of our running example, and suppose that only Cefalonia has a Casino. According to the above, the partition of compound terms to the set of *valid* (meaning-full) compound terms and *invalid* (meaning-less) compound terms is the following:

Valid Compound Terms	Invalid Compound Terms
Earth, AllSports	Olympus, SeaSports
Greece, AllSports	Crete, WinterSports
Finland, AllSports	Cefalonia, WinterSports
Olympus, AllSports	Rethimno, WinterSports
Crete, AllSports	Heraklio, WinterSports

Cefalonia, AllSports	Olympus, SeaSki
Rethimno, AllSports	Olympus, WindSurfing
Heraklio, AllSports	Crete, SnowBoard
Earth, SeaSports	Cefalonia, SnowBoard
Greece, SeaSports	Rethimno, SnowBoard
Finland, SeaSports	Heraklio, SnowBoard
Crete, SeaSports	Crete, SnowSki
Cefalonia, SeaSports	Cefalonia, SnowSki
Rethimno, SeaSports	Rethimno, SnowSki
Heraklio, SeaSports	Heraklio, SnowSki
Earth, WinterSports	Olympus, SeaSports, Casino
Greece, WinterSports	Crete, WinterSports, Casino
Finland, WinterSports	Cefalonia, WinterSports, Casino
Olympus, WinterSports	Rethimno, WinterSports, Casino
Earth, SeaSki	Heraklio, WinterSports, Casino
Greece, SeaSki	Olympus, SeaSki, Casino
Finland, SeaSki	Olympus, WindSurfing, Casino
Crete, SeaSki	Crete, SnowBoard, Casino
Cefalonia, SeaSki	Cefalonia, SnowBoard, Casino
Rethimno, SeaSki	Rethimno, SnowBoard, Casino
Heraklio, SeaSki	Heraklio, SnowBoard, Casino
Earth, WindSurfing	Crete, SnowSki, Casino
Greece, WindSurfing	Cefalonia, SnowSki, Casino
Finland, WindSurfing	Rethimno, SnowSki, Casino
Crete, WindSurfing	Heraklio, SnowSki, Casino
Cefalonia, WindSurfing	Olympus, AllSports, Casino
Rethimno, WindSurfing	Crete, AllSports, Casino
Heraklio, WindSurfing	Rethimno, AllSports, Casino
Earth, SnowBoard	Heraklio, AllSports, Casino
Greece, SnowBoard	Crete, SeaSports, Casino
Finland, SnowBoard	Rethimno, SeaSports, Casino
Olympus, SnowBoard	Heraklio, SeaSports, Casino
Earth, SnowSki	Olympus, WinterSports, Casino
Greece, SnowSki	Crete, SeaSki, Casino
Finland, SnowSki	Rethimno, SeaSki, Casino
Olympus, SnowSki	Heraklio, SeaSki, Casino
Earth, AllSports, Casino	Crete, WindSurfing, Casino
Greece, AllSports, Casino	Rethimno, WindSurfing, Casino

Cefalonia, AllSports, Casino	Heraklio, WindSurfing, Casino
Earth, SeaSports, Casino	Olympus, SnowBoard, Casino
Greece, SeaSports, Casino	Olympus, SnowSki, Casino
Cefalonia, SeaSports, Casino	Finland, AllSports, Casino
Earth, WinterSports, Casino	Finland, SeaSports, Casino
Greece, WinterSports, Casino	Finland, WinterSports, Casino
Earth, SeaSki, Casino	Finland, SeaSki, Casino
Greece, SeaSki, Casino	Finland, WindSurfing, Casino
Cefalonia, SeaSki, Casino	Finland, SnowSki, Casino
Earth, WindSurfing, Casino	Finland, SnowBoard, Casino
Greece, WindSurfing, Casino	
Cefalonia, WindSurfing, Casino	
Earth, SnowBoard, Casino	
Greece, SnowBoard, Casino	
Earth, SnowSki, Casino	
Greece, SnowSki, Casino	

However, even from this toy faceted taxonomy it is more than obvious that partitioning the set of compound terms to those that are valid and those that are invalid is a formidably laborious task for the designer. This is the reason why FASTAXON supports the *Compound Term Composition Algebra (CTCA).* The *Compound Term Composition Algebra (CTCA)* is an algebra that can significantly reduce the required effort. According to that approach the designer can use an algebraic expression to define the valid (meaningful) compound terms by declaring only a *small* set of valid or invalid compound terms from which other (valid or invalid) compound terms are then *inferred.*

Specifically, using the Compound Term Composition Algebra we can define the above partition using the expression:

(Places [-N] Sports) [+P] Facilities

with the following **P** and **N** parameters:

N={ {Crete,WinterSports}, {Cefalonia, WinterSports} } and

P={ {Cefalonia, SeaSki, Casino}, {Cefalonia, Windsurfing, Casino} }

For more about the Compound Term Composition Algebra please refer to the article [1].

More articles (older or subsequent) can be found in www.csi.forth.gr/~tzitzik/publications.htm. There is also an XML DTD that allows publishing taxonomies of compound terms which are defined by using the Compound Term Composition Algebra. It is called XFML+CAMEL . For example, the representation of the faceted taxonomy and the above algebraic expression using XFML+CAMEL can be found here.

For any comments or questions concerning CTCA and XFML+CAMEL please contact with Yannis Tzitzikas (http://www.csi.forth.gr/~tzitzik/)

3 Quick overview of FASTAXON program

Fastaxon software consists of several integrated modules that are divided to tree interfaces. Modules are designer interface including the expression builder, object indexer interface and the end-user interface.

3.1 How I can use FASTAXON?

A FASTAXON project includes

- taxonomies,
- facets,
- algebraic expressions, and
- an object base.

For creating a new project one has to

1. create the desired taxonomies
2. define the desired facets
3. formulate the expression that specifies the meaningful compound terms
4. start indexing objects using the navigation tree

For more details refer to the following chapters.

4 Designer interface

Picture 2. The Designer login screen.

4.1 Project

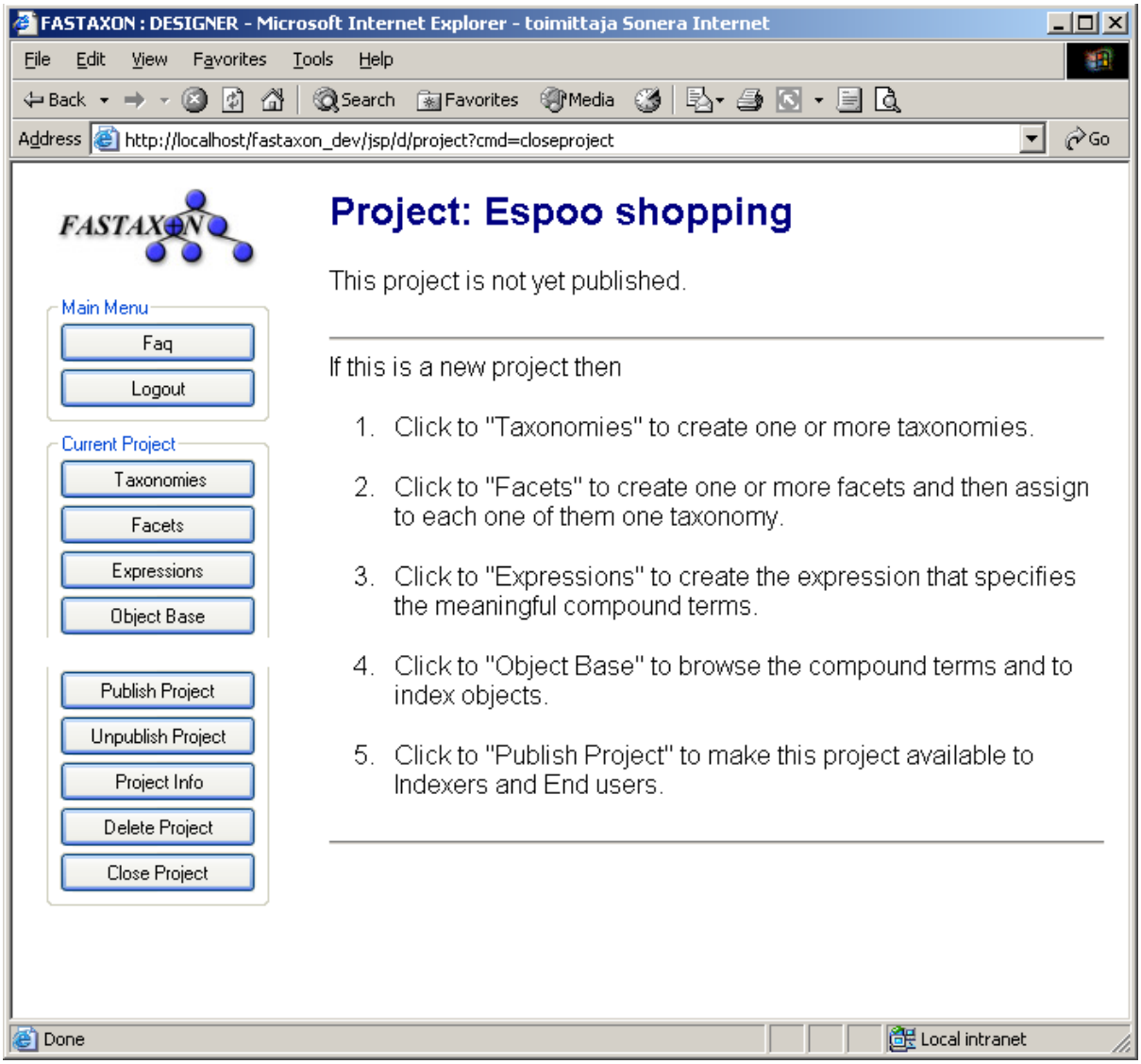
Project is a main container of data in FASTAXON system. Project is owned by one designer and it holds project specific data (taxonomies, facets, expressions) inside it.

4.1.1 Create a new project

Create new project by selecting menu command **Create Project.**
Project name must be unique. Project is owned by the designer and other designers can't modify it or it's content.

4.1.2 Open existing Project

Open existing project by selecting menu command **Open Project**. Select project from a list. Only projects owned by the designer are visible in a project list.



Picture 3. The Project navigation screen.

4.1.3 Publish Project

When project is open it can be published. Select menu command **Publish Project** for publishing the current project. Publishing the project means that it becomes visible for end-users. It means also that editing the project is not allowed.

4.1.4 Unpublish Project

Published projects can be unpublished by selecting menu command **Unpublish Project.**

4.1.5 Project Info

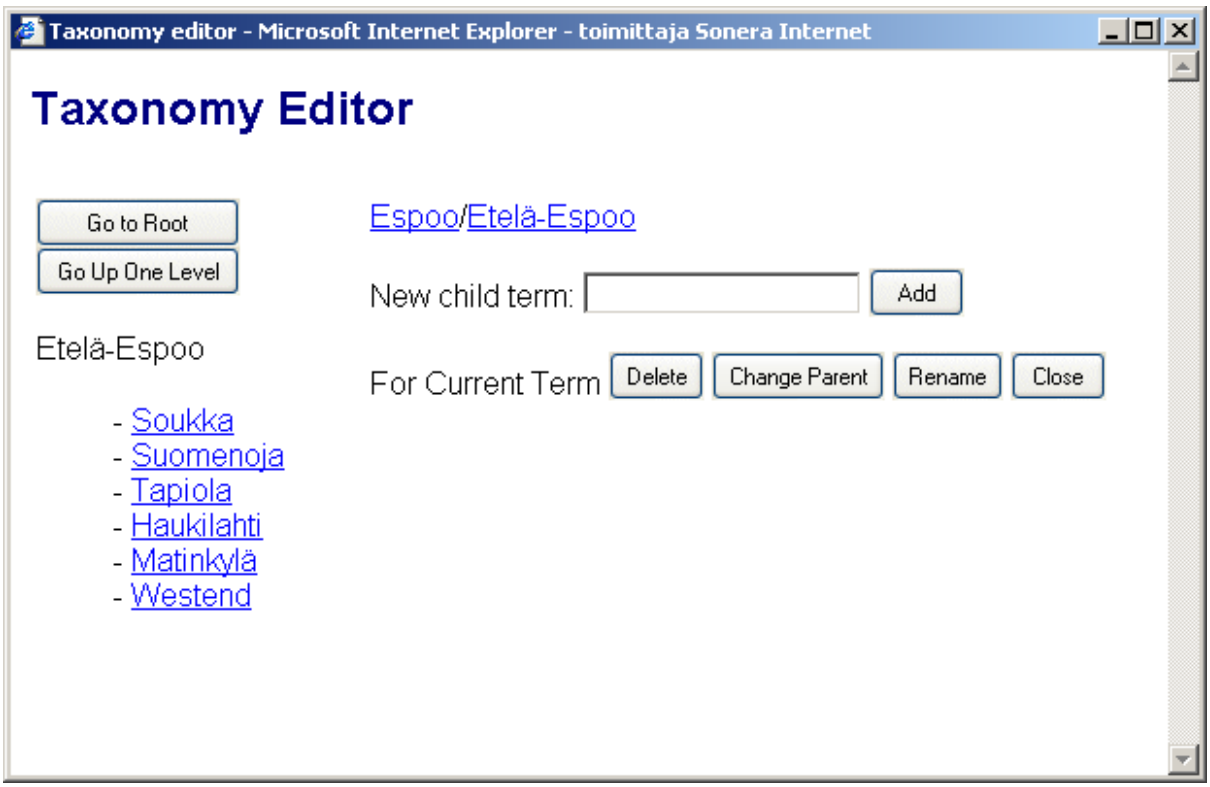
Information about the project can be viewed by selecting menu command **Project Info.**

4.1.6 Delete Project

When project is open it can be deleted. Select menu command **Delete Project** for deleting the current project.

4.2 Taxonomy

Taxonomy is a hierarchically organized set of terms. Currently FASTAXON supports only tree hierarchies. This means that in current implementation every leaf has only one parent but one parent can have any amount of leafs. Terms can be added only as leafs.



Picture 4. The Taxonomy Editor screen.

User can add, modify and delete taxonomies and terms in taxonomies by selecting menu command **Taxonomies.** This command will open a list of taxonomies in the project.

4.2.1 Create a new taxonomy

Create new, empty taxonomy by selecting command **Add Taxonomy.**
Taxonomy name must be unique. Taxonomies are not visible between projects. When taxonomy is opened first time the root term is required to set.

4.2.2 Open and closing existing taxonomy

Open existing taxonomy by clicking taxonomy name by mouse. Taxonomy editor window is opened and the root term is selected. Taxonomy is closed by selecting **Close** command

4.2.3 Browsing the taxonomy tree

When taxonomy is opened the root term of taxonomy is selected as current term. All childs of the current term are visible as a list below the term.

User can browse down in a tree towards the leaf by selecting one of the childs. Path from current term to the root is visible in a top of a taxonomy editor window.

Users can go up in a tree towards the root by several ways. By selecting **Go to root** a user can go directly to the root. By selecting **Go up one level** a user can go previous level. Users can also use path from current term to root and select any of the terms in the path to be a current term.

4.2.4 Add and delete terms to the taxonomy

Browse to the correct position in the taxonomy tree. Add a term by defining a name and select **Add term**. Current term can be deleted to by selecting **Delete current term.**

4.2.5 Rename term

Any term in taxonomy can be renamed by selecting **Rename** command when the term is selected.

4.2.6 Change Parent

Taxonomies can be modified by moving branches or leaves under another parent.

4.2.7 Delete Taxonomy

When taxonomy list is visible select command **Delete** beside the taxonomy for deleting it

4.3 Facet

The facets are used to give the taxonomies a name. Facets in a project must be unique but one taxonomy can have several facets. Select command **Facets** to view and modify the facets of a current project.

4.3.1 Add Facet

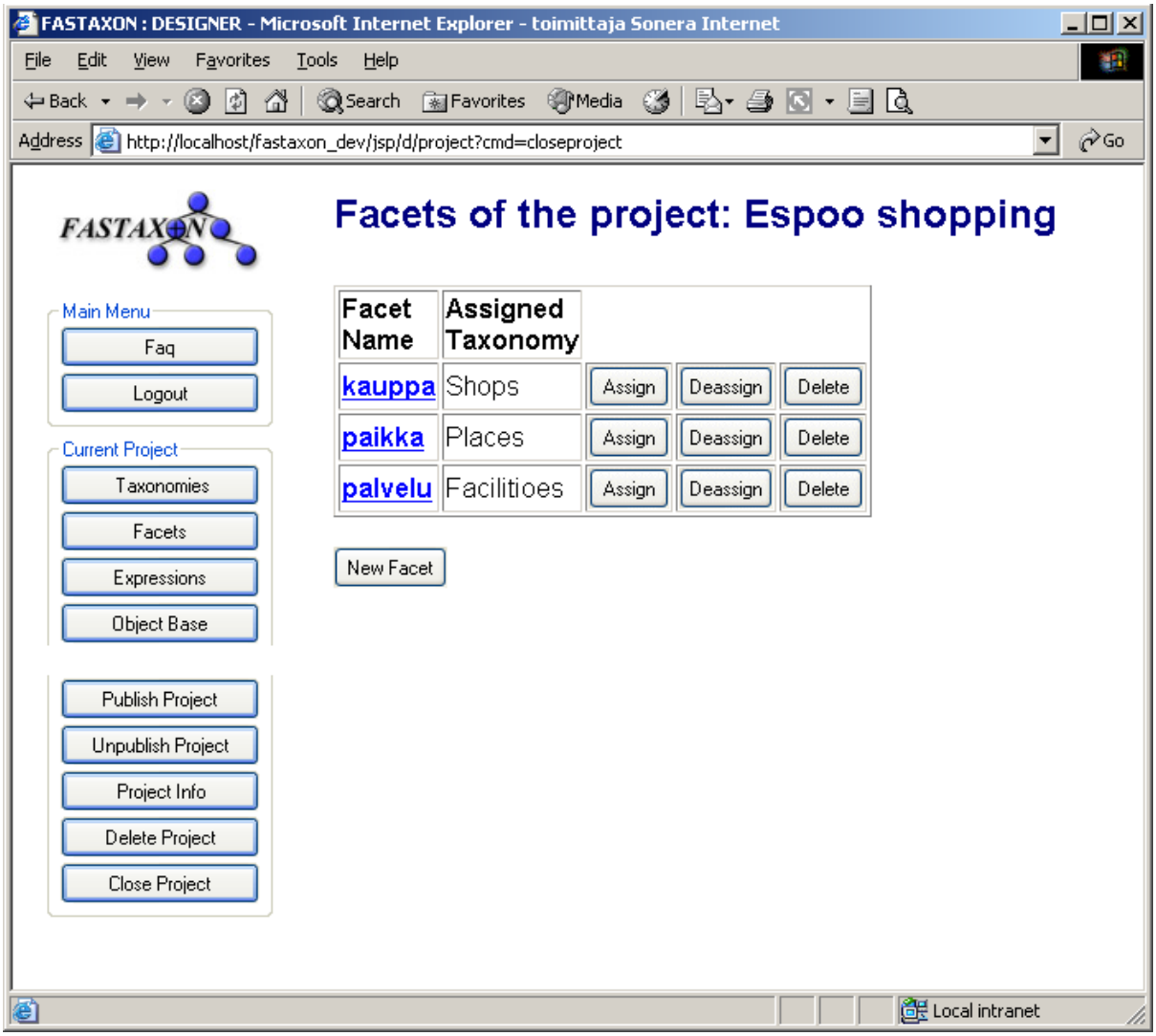
Select **Add facet** command to add new facets to the project. Add unique names for every facet in one project.

4.3.2 Assign and Deassign a taxonomy to the Facet

Select a facet from a facet list by clicking it by mouse. Select a taxonomy to be assigned from a dropdown list. Deassign taxonomy by selecting the command **Deassign.**

4.3.3 Delete Facet

When list of facets is visible select command **Delete** beside the facet for deleting it

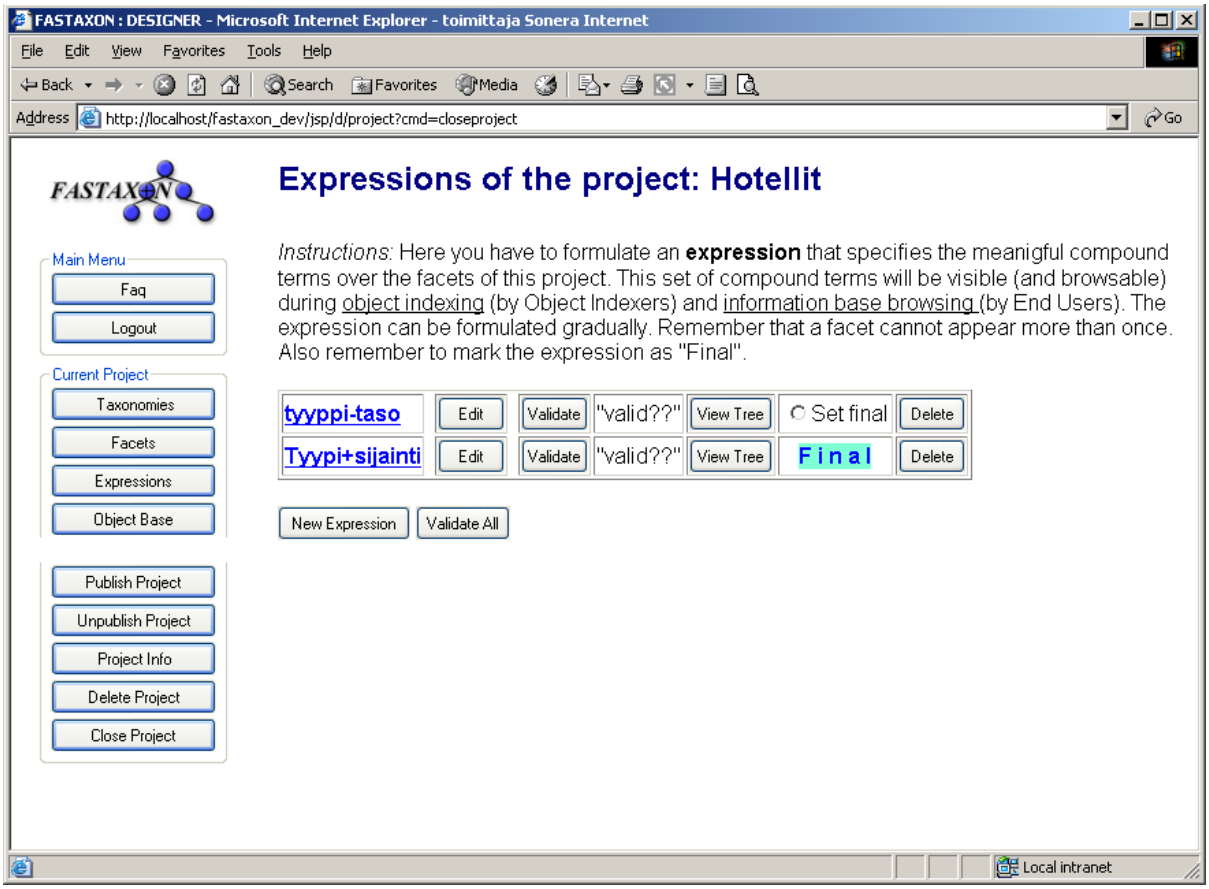


Picture 5. Facet navigation screen.

5 Expression builder

Expressions are rules created with compound term composition algebra that allow users to determine the set of compound terms that is valid. Expressions use facets as objects and modify the compounds by using four different modifiers (PLUS, MINUS, SELFPLUS and SELFMINUS). When proper set of expressions is created designer will validate the navigation tree for object indexing.

By selecting menu command **Expressions** users can see the list of expressions of the project. In Expression navigating screen users can add new expressions and modify the existing ones. Also validation of navigation trees is done here.



Picture 6. Expression navigation screen.

5.1 Creating expressions

By selecting a command **New Expression** users can add new expressions to the project. Expression consists facets or existing expressions and compound term composition algebra modifier. All expressions must be connected to each other by creating a new expression between them. The expression that connects all other expressions will be set as **Final.**

5.1.1 PLUS

Plus term will validate selected compound terms.

5.1.2 MINUS

Minus term will invalidate selected compound terms

5.1.3 SELFPLUS

SelfPlus term will validate term itself

5.1.4 SELFMINUS

Selfminus term will invalidate term itself.

5.2 Validating navigation tree

When expressions are created the navigation tree must be validate. Validation is done by selecting menu command **validate** for single expression or by selecting **validate all** foe all expressions. The validation will create a navigation tree that can be viewed by selecting a menu command **View Tree.**

5.3 Editing an expression

Expression can be edited by selecting a menu command **Edit** beside the expression. After editing the validation of the expression must be done again.

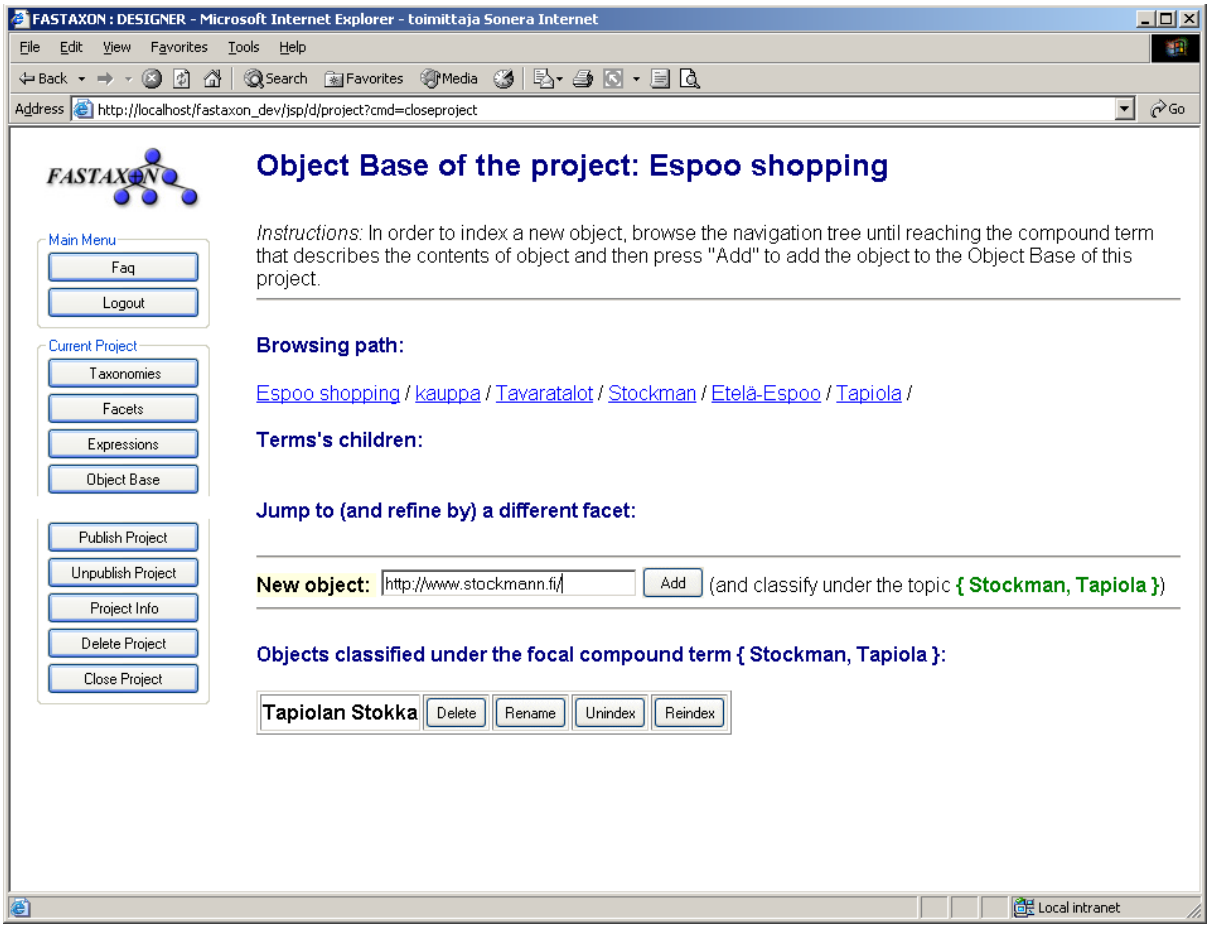
5.4 Deleting an expression

An expression can be deleted by selecting a menu command **Delete** beside the expression.

6 Object Indexer interface

Separate Object indexer interface will not be implemented in this project. Object indexing will be done in the designer interface with the designer users rights. Functionalities of the object indexing will be the same in a designer interface that was required in a separate Object Indexer interface.

6.1 Object Base



Picture 7. Object Indexing screen.

7 End-User Interface

When the project is published by the designer it becomes visible to the end-users. End-users can browse the selected information base and search the wanted objects.

Picture 8. End-User Welcome screen.

8 Installation and Administration

8.1 Installation

Installation procedure is described in a separate installation instructions document.

8.2 Administration

Adding the users for the Fastaxon software for the moment must be done by modifying directly the MySQL database.

9 Troubleshooting

10 References

[1] Y. Tzizikas, A. Analyti, N. Spyratos, P. Costantopoulos. ”An Algebraic Approach for Specifying Compound Terms in Faceted Taxonomies”, EJC2003, Japan. (the paper in pdf, slides) .