

---

**IFS - TU Vienna**

---

**SemanticLIFE  
User Requirements Specification  
For Query Component**

**Version <1.0>**

SemanticLIFE	Version: <1.0>
User Requirements Specification	Date: <07/Feb/2006>
<document identifier>	

## Revision History

Date	Version	Description	Author
<07/Feb/06>	<1.0>	Document outline based on Shuaib's suggestion	Hoang Huu Hanh
<09/Feb/06>	<1.0>	Description in detailed	Hoang Huu Hanh
<10/Feb/06>	<1.0>	Description modifying and adding	Hoang Huu Hanh

SemanticLIFE	Version: <1.0>
User Requirements Specification	Date: <07/Feb/2006>
<document identifier>	

## Table of Contents

1.	Introduction	4
2.	Overall Description	4
3.	Specific Requirements of Query Component	4
3.1	List of Tasks	4
3.2	Concerned Information / Data Objects	4
3.3	Filtering / Control Options	5
3.4	Tracking of Task State (by Saving Its Status)	5
3.5	Status Messages	5
3.6	Visible/Invisible Navigation	5
3.7	Undo/Redo Functionalities	5
3.8	Copy/Paste Functionality	5
3.9	Scratchpad functionality	5
3.10	Active Items list functionality	5
3.11	Caching functionality	5
3.12	User Interfaces	5
4.	Requirements of Query Component for the SemanticLIFE System	5
4.1	List of Tasks	5
4.2	Concerned Information / Data Objects	6
4.3	Filtering / Control Options	6
4.4	Tracking of Task State (by Saving Its Status)	6
4.5	Status Messages	6
4.6	Visible/Invisible Navigation	6
4.7	Undo/Redo Functionalities	6
4.8	Copy/Paste Functionality	6
4.9	Scratchpad functionality	6
4.10	Active Items list functionality	6
4.11	Caching functionality	7
5.	Supporting Information	7

SemanticLIFE	Version: <1.0>
User Requirements Specification	Date: <07/Feb/2006>
<document identifier>	

# User Requirements Specification

## 1. Introduction

This document gathers the user requirements on Query mechanism of the SemanticLIFE. It also points out crucial points of the query mechanism.

## 2. Overall Description

It is important to determine what tasks user needs to perform by our system. Then we can decide where to perform those tasks, i.e., distributed in UI and business components, or entirely in business components. Later, the detailed design about UI and business components would follow. At least, try to fill 2 data collection forms (one for component of your interest like annotation / query / classes & instances / ontology management / user profile / dynamic associations / privacy & access control / ..., and the other for over all system SemanticLIFE in context of Scenario 2). The following list (tasks and information objects) is numbered according to importance of each item

## 3. Specific Requirements of Query Component

### 3.1 List of Tasks

Tasks or operations to be performed by Query component:

1. Retrieving information from ontologies
2. Forming queries in four ways:
  - a. QBE
  - b. Templates
  - c. Context-based (source-based)
    - i. Get links
    - ii. Get related data
    - iii. Named queries
  - d. Free text queries (using WordNet and Ontology to refine/expand the queries)
3. Query expansion initial generated queries especially the .
4. Querying the demanded information upon the user's request which is generated using the above ways.
5. Query results transformation
6. Saving/Logging queries
7. Opening existing queries
8. Saving/Logging query results
9. Semantic query caching to improving the performance of the query execution.

### 3.2 Concerned Information / Data Objects

1. Type of information object: the concerned information objects for query component are class, instance, property ... and the ontologies.
2. Nature: all SemanticLIFE data sources
3. Frequency:
4. Any other specificity about nature of information items: annotation data, user profile / ontology

SemanticLIFE	Version: <1.0>
User Requirements Specification	Date: <07/Feb/2006>
<document identifier>	

### 3.3 Filtering / Control Options

What are the options for user control?

- Start/Stop/Resume/Schedule queries

### 3.4 Tracking of Task State (by Saving Its Status)

One task may consist of several steps. Is it beneficial to track those steps for replaying later?

### 3.5 Status Messages

Is it necessary to display the status information for this task? Somehow: YES

Status information is to be displayed:

- Percent of completing query execution
- Inquiring data sources
- Number of found items and their related data sources

### 3.6 Visible/Invisible Navigation

Although a task can be made automatic as far as possible, however, it is sometimes beneficial to display the task execution steps to the user for developing a mental image.

### 3.7 Undo/Redo Functionalities

During execution of a sequence of queries, is it beneficial to undo/redo the selected queries instead of undoing all.

### 3.8 Copy/Paste Functionality

For example, apply one type of recently made annotation on another collection of items. Or copy one sub-query from a named query to another named query.

### 3.9 Scratchpad functionality

Query component uses Scratchpad as a temporary cache of on-going executing queries

- Drag and drop saved queries, being-formed queries, etc. to the scratchpad.

### 3.10 Active Items list functionality

...

### 3.11 Caching functionality

- Caching queries
- Caching results of the query
- Based on analysis the caches, new template queries could be generated.

### 3.12 User Interfaces

See Q-GUI document

## 4. Requirements of Query Component for the SemanticLIFE System

### 4.1 List of Tasks

Tasks or operations to be performed by Query component in SemanticLIFE:

1. Forming queries by above mentioned ways

SemanticLIFE	Version: <1.0>
User Requirements Specification	Date: <07/Feb/2006>
<document identifier>	

2. Querying the demanded information upon the user's request which is generated using the above ways.
3. Query results transformation
4. Saving/Logging queries
5. Opening existing queries
6. Saving/Logging query results

#### 4.2 Concerned Information / Data Objects

1. Type of information object: the concerned information objects for query component are class, instance, property ... and the ontologies.
2. Nature: all SemanticLIFE data sources
3. Frequency: quite often.
4. Any other specificity about nature of information items: annotation data, user profile / ontology

#### 4.3 Filtering / Control Options

What are the options for user control?

- Start/Stop/Resume/Schedule queries

#### 4.4 Tracking of Task State (by Saving Its Status)

One task may consist of several steps. Is it beneficial to track those steps for replaying later?

#### 4.5 Status Messages

Is it necessary to display the status information for this task? Somehow: YES

Status information is to be displayed:

- Percent of completing query execution
- Found items
- 

#### 4.6 Visible/Invisible Navigation

Although a task can be made automatic as far as possible, however, it is sometimes beneficial to display the task execution steps to the user for developing a mental image.

#### 4.7 Undo/Redo Functionalities

During execution of a sequence of queries, is it beneficial to undo/redo the selected queries instead of undoing all.

#### 4.8 Copy/Paste Functionality

For example, apply one type of recently made annotation on another collection of items. Or copy one sub-query from a named query to another named query.

#### 4.9 Scratchpad functionality

Query component uses Scratchpad as a temporary cache of on-going executing queries

- Drag and drop saved queries, being-formed queries, etc. to the scratchpad.

#### 4.10 Active Items list functionality

YES

SemanticLIFE	Version: <1.0>
User Requirements Specification	Date: <07/Feb/2006>
<document identifier>	

#### **4.11 Caching functionality**

- Caching queries
- Caching results of the query
- Based on analysis the caches, new template queries could be generated.

### **5. Supporting Information**

Hoang Huu Hanh, hanh@ifs.tuwien.ac.at