Usability evaluation in the SEWASIE (SEmantic Webs and AgentS in Integrated Economies) project

Tiziana Catarci[1] , *Tania Di Mascio*[1], *Paolo Dongilli*[2], *Enrico Franconi*[2], *Giuseppe Santucci*[1], *Sergio Tessaris* [2]

[1]University of Rome “La Sapienza”, Italy, <*lastname*>@dis.uniroma1.it
[2]Free University of Bozen-Bolzano, Italy, <*lastname*>@inf.unibz.it

Abstract

In this paper we describe the usability evaluation experiments of the Query tool system done in the context of the SEWASIE (SEmantic Webs and AgentS in Integrated Economies) project. The usability evaluation is an important step of the User-Centered Design Methodology, followed to develop the SEWASIE system. This European IST project aims at enabling a uniform access to heterogeneous data sources through an integrated ontology. From the architectural point of view, it is composed of different tools, in particular, the Query tool allows the user to construct the query by a focus plus context diagrammatic interface generating precise and unambiguous query expressions. The main goal of our experiment is to demonstrate the easy of use of the Query tool independently of the domain user experience. This study confirms that the Query tool system is usable as for the end users (domain-expert users) as for the non-domain expert users.

1 Introduction

In this paper, we describe the usability evaluation experiments of the Query tool system done in the context of the SEWASIE project that aims at enabling a uniform access to heterogeneous data sources through an integrated ontology. The overall project strictly follows the User-Centered Design Methodology (UCDM) involving users from the very beginning both in the design and test steps. From an architectural point of view, SEWASIE aims at providing an open and distributed architecture based on intelligent agents (e.g., query agent and brokering agent) facing scalability and flexibility issues, i.e. the ability to fit in changing and growing environments and to interoperate with other systems, while offering one central point of access to the user.

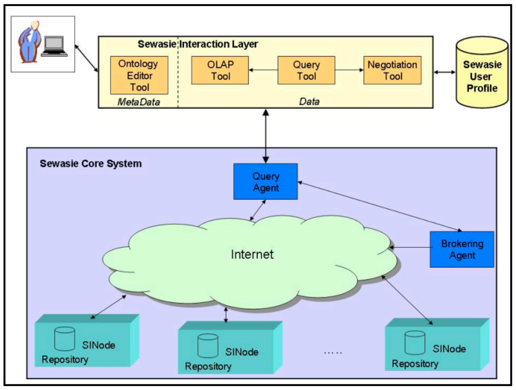


Figure 1.1: Architecture of the Sewasie System.

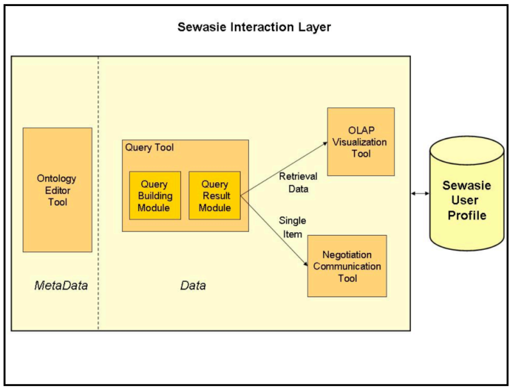


Figure 1.2: Interaction Layer.

The Figure 1.1 highlights the major Sewasie components wrt the topics the paper deals with. The Sewasie **Interaction Layer**, which is a crucial component of the overall SEWASIE architecture, is composed of tools (see the Figure 1.2) which work together to offer an integrated, easy to use user interaction with the SEWASIE system. In particular, the Query tool allows the user to construct the query by a focus plus context diagrammatic interface generating precise and unambiguous query expressions; moreover, interface presentation and behaviour are entirely guided by the ontology. The Sewasie **User Profile**, based on a domain-interest model, is used to offer the most appropriate set of tools depending on the user's expertise, goals, and interests. In the Sewasie **Core System** we identify, as repository of the local ontology, the Sewasie Information Nodes (SINodes), which work to define and maintain a single administrative, or logical node of information presented to the network. The brokering agent is responsible for maintaining a view of the knowledge handled by the network, as well as the information on the specific content of some SINodes, which are under direct control. The query agent is the carrier of the user query from the Interaction Layer to the SINodes, and has the task of solving a query, interacting with the brokering agent.

This paper is organised as follow, Section 2 defines the general and operational guidelines that will guide the overall SEWASIE's experiment evaluation sessions. Section 3 describes the Query tool experiments session and the Section 4 collects the conclusions.

2 Methods

Several different definitions of usability exist ([5], [6], and [7]). A very comprehensive definition of usability is given as *"the extent to which a product can be used with efficiency, effectiveness, and satisfaction by specific users to achieve specific goals in specific environments."* It can be evaluated by assessing three factors: the effectiveness (i.e., the extent to which the intended goals of the system can be achieved, the efficiency (i.e., the time, the money and the mental effort spent to achieve these goals), and the satisfaction (i.e., how much the users feel themselves comfortable using the system).

The usability is an essential quality of the overall Software Systems (SS). For this reason the usability evaluation depends on different aspects, i.e., the types of users, the characteristics of the tasks, the equipment (hardware, software, and materials), and the physical and organizational (e.g., the working practices) environment.

There are many ways to evaluate the interaction quality between the users and the SS, and then there are many Usability Evaluation Methods (UEMs). They are referred to any method or technique used to perform usability evaluation, with emphasis on formative usability evaluation i.e., usability evaluation/testing used to improve usability of an interaction design at any stage of its development. A common difference among UEMs is based on skill of evaluators (in general, a person using a UEM to evaluate usability of an interaction design). In the Expert-based criteria, experts are requested to evaluate a prototype, comparing it w.r.t. existing rules and guidelines; in the User-based criteria, evaluators assess usability through real users, having them "using" a prototype. In particular, while the **Expert-based** Criteria UEMs include, among others, Heuristic Evaluation method [8], Cognitive Walkthrough method, and Expert-based method [9]; the **User-based** Criteria UEMs [10] includes, among others, Observational evaluation method, Survey evaluation method, and Controlled experiment method. Among observational evaluation methods, we focus on Verbal Protocols [11], and Think Aloud Protocol [12].

Since the SEWASIE system follows the UCDM, we decide using the UEMs assessing usability through real users. Hereinafter, we briefly describe the user-based criteria UEMs; for details of these methods and for the complete description of the Expert-based Criteria UEMs we refer to [2].

Observational evaluation method involves real users that are observed when performing tasks with the system (depending on the stage of the project, what "the system is" ranges from paper mock-ups to the real product). This method offers a broad evaluation of usability. Depending on the specific situation, we may either apply the observational evaluation by direct observation or record the interaction between users and system. This recording (done by video camera) is more valuable, but it is much expensive (especially for the time required to analyze the recorded data).

- Think Aloud Protocol provides the evaluator with information about cognitions and emotions of a user while the user performs a task or solves a problem. The user is instructed to articulate what s/he thinks and feels while working with a prototype. The utterances are recorded either using paper and pencil or using audio and/or video recording. By using the Think Aloud Protocol, the evaluator obtains information about the whole user interface. This protocol is oriented towards the investigation of the user's problems and decisions while working with the system.
- Verbal protocols aim at eliciting the user's (subjective) opinions. Examples are interviews and questionnaires. The difference between oral interview techniques and questionnaire based techniques lies mainly in the effort for setup, evaluating the data, and the standardization of the procedure.

Survey evaluation method. In this case, structured questionnaires and/or interviews are used to get feedback from the users. This method offers a broad evaluation of usability since from the user's viewpoint it is possible to identify the critical aspects in user-system.

Controlled experiment method . This method is particularly valid to test how a change in the design project could affect the overall usability. It may be applied in any phase during the development of a system; it provides more advantages when it is possible to test separately the alternative designs, independently by the whole system. This method mainly aims at checking some specific cause-effect relations, and this is possible by controlling as many variables as we can.

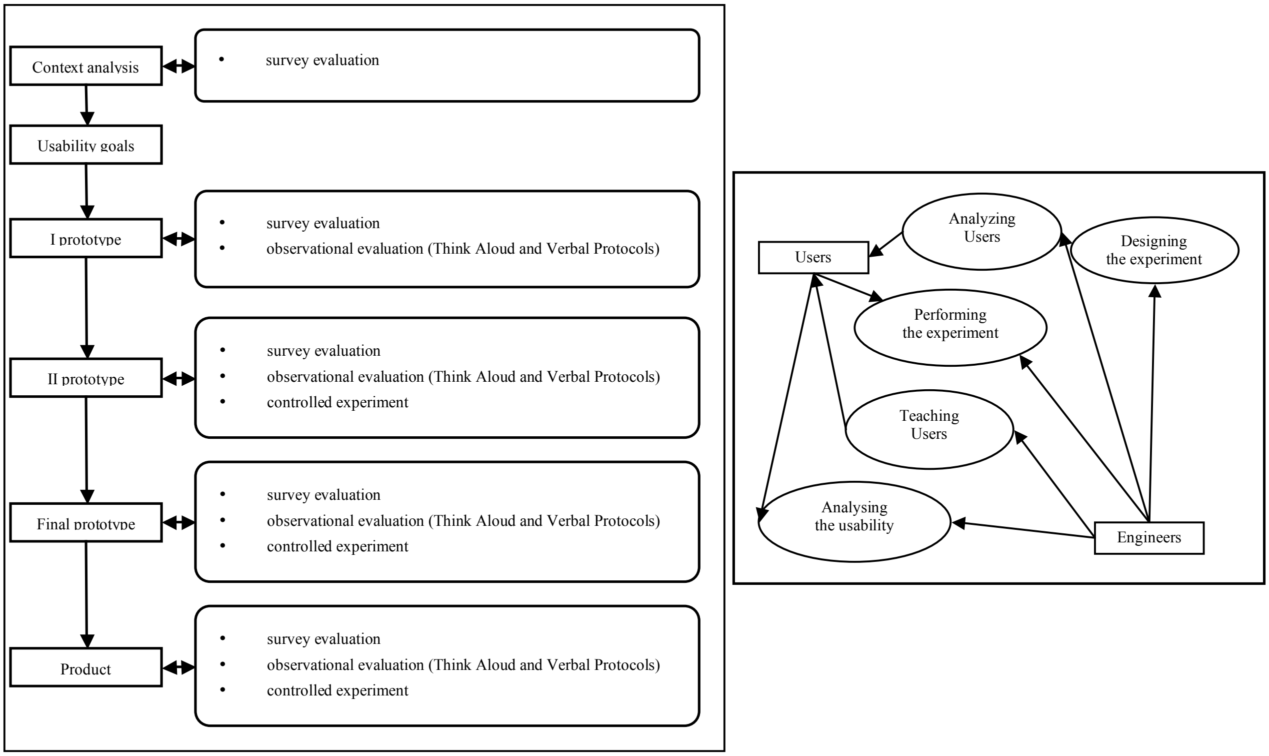


Figure 2.1: Stages of UCDM and suitable evaluation methods, and the main schema followed for the experiments.

In the Sewasie project, we recognize different tools (e.g., Query tool, Negotiation tool, etc.). These tools, in a context of UCDM, evolve through different design steps, shown on the left part of the Figure 2.1 together with the different UEMs methods that are appropriate to evaluate the usability of each component along its evolution. For the usability evaluation of the Query tool, we follow the schema depicted in the right of the Figure 2.1 that represents the user-based criteria UEMs main schema. It is composed of the different following steps:

Analyzing users → a few user classifications have been proposed in the literature (see e.g., [13]). Each of them identifies a certain number of features, which permits the labelling of a homogeneous group of users. The number and the kinds of groups differ depending on the specific classification. However, there is at least a general agreement on the initial splitting of the users into two large groups: those who have had a certain instruction period and have technical knowledge, and those who do not have specific training in computer science. Actually, in the experiment we call those two groups skilled and unskilled users respectively. The several features roughly characterize the unskilled user: s/he interacts with the computer only occasionally, s/he has little, if any, training on computer usage, s/he has low tolerance for technical aspects, s/he is unfamiliar with the details of the internal organization of an information system. Usually, this user does not want to spend extra time in order to learn how to interact with a system, and finds it irritating to have to switch media, e.g., to manuals, in order to learn how to interact with the system. Moreover, the unskilled user wants to know where s/he is and what to do at any given moment of the interaction with the system. Notice that the unskilled user is very similar to Cuff's [14] casual users. On the other hand, skilled users possess knowledge of considered systems, information systems, etc., and often like to acquire a deep understanding of the system they are using.

Designing the experiment → the main goal of the experiment design is to propose a complexity model and to validate the metrics used to measure the system usability. In order to estimate the usability, the evaluators provide not only to define precisely what one is going to watch/measure, but also they provide to develop tasks for users to perform; moreover, they measure relevant parameters (metrics) of user performance, and they validate values collected during the experiments.

Teaching users → the goal of the usability evaluation experiments is to measure the effectiveness and the efficiency of the system and the user's satisfaction using it, discarding each aspect involving the learning time of the different environments. For this reason, this step aims at making users aware of system functionality and experiments modalities. Following this guideline during the teaching users step, we set up exhaustive explanation about each tool. In this way the users were totally acquainted with the usage of the system and, during the final test, they were free of concentrating exclusively on the tasks execution.

Performing the experiment →during this step, the evaluators provide to teach users about the experiment and to assign users the developed task. Moreover, they take notes of any conditions or events, which occur during the experiment.

Analysing the usability → the evaluators collect the information on each performed test and in order to obtain statistically significant metric values, is very important validate such results with an Anova test (AN analysis Of Variance test) [10]. While the analysis of results is in charge of the evaluators, all people involved in the experiment, as mentioned in the User Centered Design Methodology [1] perform the evaluation of the usability.

3 Query tool experiments session

In this section, we describe the evaluation experiments of the Query tool [4]. This tool guides the users in the construction of the query, it shows the results of the queries to the users, and it allows connecting it with the other Sewasie tools for example the OLAP tool and the Negotiation tool. The user does not need to know the terms to compose the query; neither the system needs to consider alternative meanings of a given query. Interface presentation and behaviour are entirely guided by the ontology. The results are represented using a table-like visualization. Table fields contain the query expression, the URIs of the data sources.

The method we use for the experiments is the observational evaluation method and, in particular, the Think Aloud and Verbal Protocols (above mentioned). Also, we record the tests with a video camera in order to valuate rigorously a lot of information, for example the critical points during the interaction (when the user has to consult the manual, when and where s/he is blocked, etc.), the time a user spends to perform a task, the mistakes a user makes, and so on. To describe the evaluation experiments of the Query tool, we adopt the schema above described.

Analysing users

Tree people are involved in this session of the usability evaluation experiment. In particular, while two people are very skilled in computer science, the other one is unskilled in computer science and he uses the computer only at work. These users belong to the *Employees of Provincial and Municipal offices* class, (defined in [1]) and they well represent the end-users for the Query tool environments. We consider, the end-users, as Domain Expert users, differently from the five students (that perform the complexity model experiment session) that we classified as Non Domain Expert (NDE). This classification in NDE and DE is very important in our context; in fact, the main goal of our experiment is to demonstrate the easy of use of the Query tool independently of the domain user experience.

Designing experiments

The objective of our study is to measure and understand the use complexity of the Query tool. More specifically, we are interested in determining how much is difficult for the users to construct queries, and to understand its results. In order to evaluate which is the quality of the interaction between the domain expertise of the users and the query paradigm used in the Query tool environment to construct queries, we develop different tasks for the users (the query writing and query reading tasks); moreover, we design a model of complexity, a number of query of increasing complexity, and a questionnaire to capture relevant aspects of the interface interaction.

In the *model of complexity*, for each query we assign the complexity tree (see the Figure 3.1). In the schema, the nodes represent the concepts of the ontology and the weighted edges represent the relations among them, moreover:

• C1 = 0,1	if edge = add propriety;
• C2 = 0,2	if edge = add concept;
• Pn = n	n = number of nodes per level
• l	number of levels

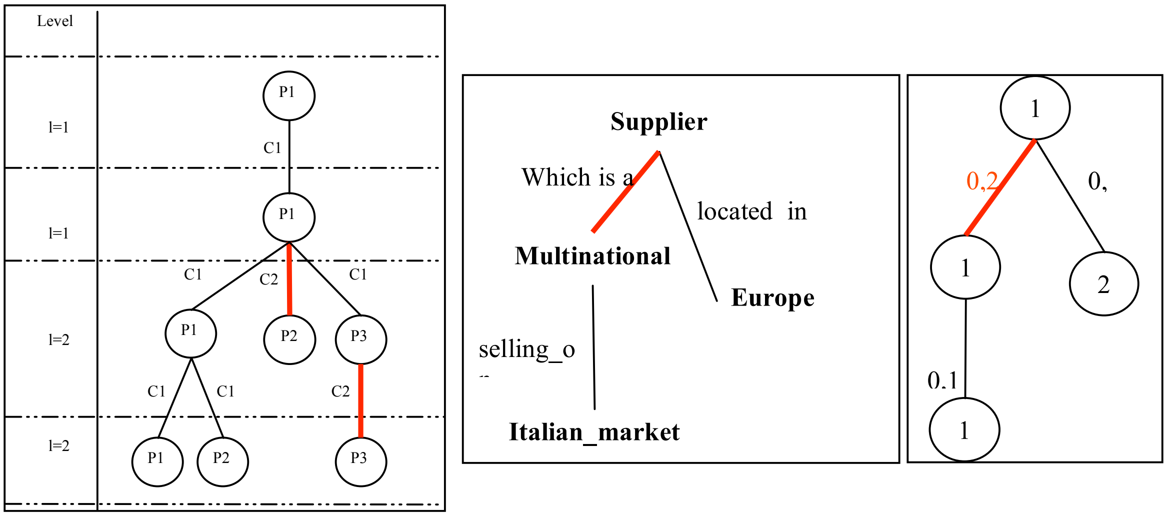


Figure 3.1: Complexity query tree.

Starting from the query tree, we define a function to calculate the complexity of the query, expressed whit the following formula:

$$\sum_{l=1}^{num\_lel} l * c_{1;2} \left(\sum_{n=1}^{num\_nod\_per\_level} p_n(l) * f(n,l) \right) \qquad f(n,l) = \begin{cases} evr\_num\_suc\_level & if \quad l < l\max \\ 1 & if \quad l = l\max \end{cases}$$

For example, for the query “Supplier located in Europe which is a Multinational selling on the Italian Market”, represented by the complexity tree depicted in the Figure 1.4, the value of the complexity is:

1*[0.1*1]*2+2*[0.2*1+0.1*2]*0.5+3*[0.1*1]*1 = **0.9**

Using this model, we devised *a list of queries* computing their complexity; the values of increasing complexity are showed in the Table 3.1, where we highlight several characteristics of the queries (e.g. number of level, the average number of successive per node, etc.) in order to well know the queries structure.
The values of the metrics we use to describe the performance for the usability evaluation are: the time spent to compose the query, the number of steps used to compose the queries, the number of focus change, the number of mistakes, the number of cancellations, and the number of clicks on the Query Manipulation Pane.

Query_i		Num Level	Num Node	Avr_num suc_per_node	Avr_num node_per_level		Complexity
Query_1		2	2	1,00	1,00		0,30
Query_2		3	3	1,00	1,00		0,60
Query_3		2	3	2,00	1,50		0,80
Query_4		3	4	1,25	1,33		0,80
Query_5		5	8	1,39	1,60		2,45
Query_6		6	9	1,20	1,50		2,80
Query_7		6	11	1,50	1,83		4,00
Query_8		7	12	1,42	1,71		5,60
Query_9		8	11	1,39	1,38		6,05

🟨 Low Complexity

🟦 Medium Complexity

🟪 High Complexity

Table 3.1: Complexity of queries.

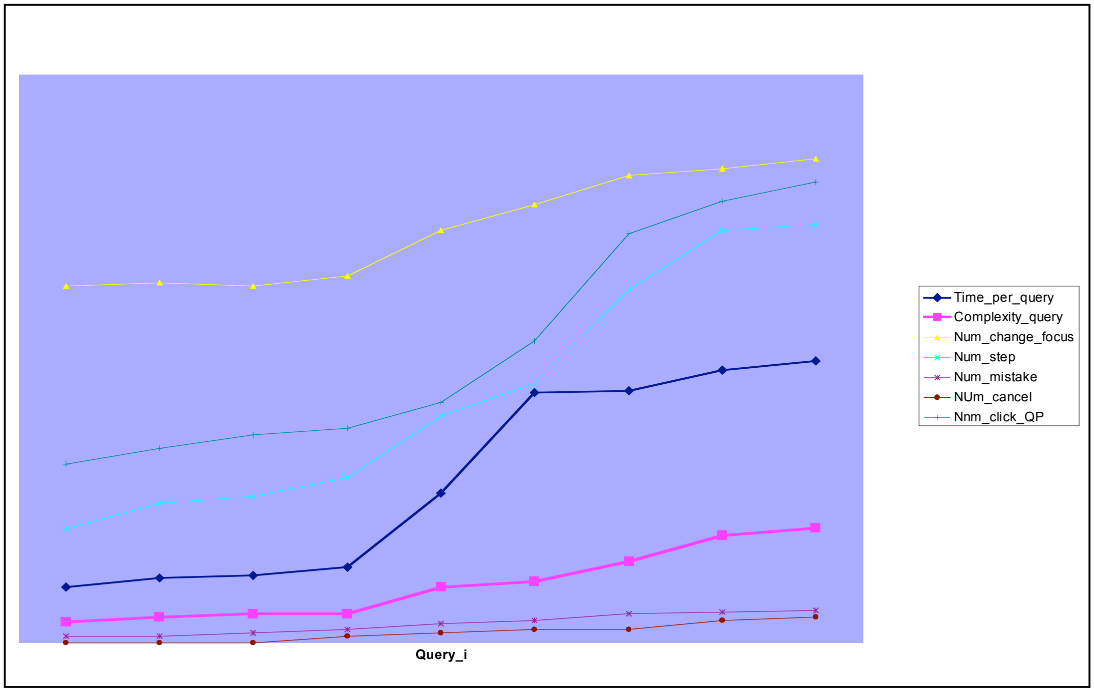


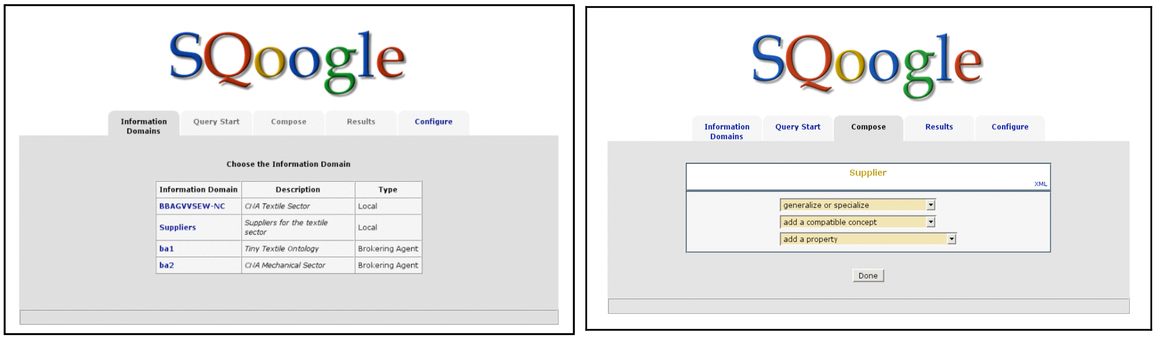
Figure 3.2: Complexity vs average values of metrics.

The complexity model, the performance metrics, the queries, and the *questionnaire* designed have been validate by a session of experiment performed with the five students mentioned in the Analysing users step (Non Domain Expert users). This test session is well described in [3], here we only highlight that the query complexity (pink line of the Figure 3.2) has the same increasing behaviour of the metrics observe during this preliminary test session. Inspecting the time spent values in the Figure 3.2 makes clear that the users learned the system using it; in fact, for the queries of high complexity (Query_7,8,9), the function of the time spent metric increases more slowly than the other metrics.

Starting from the result of this session of experiments, we note that the complexity model for the queries designed is quite stable (no change of constants in the formula, see above when we describe the query tree); and the questionnaire is validated by different type of users, and it results usable for the end-users.

Teaching users

This step aims at making users aware of system functionality and experiments modalities. While the experiments modalities are above emphasized, and the Query tool is completely described in [2], here we show, in detail, the Query tool interface, that has undergone some changes in order to enhance its user-friendliness and overall usability for both novice and experienced users. Query start, query composition, and query execution constitute a natural flow from the first web application tab on the left to the last one on the right hand side (Figure 3.3). The available tabs are Information Domains, Query Start, Compose, and Results.



Choose the Information Domain

Information Domain	Description	Type
BBAGVVSEW-NC	*CNA Textile Sector*	Local
Suppliers	*Suppliers for the textile sector*	Local
ba1	*Tiny Textile Ontology*	Brokering Agent
ba2	*CNA Mechanical Sector*	Brokering Agent

Supplier

generalize or specialize

add a compatible concept

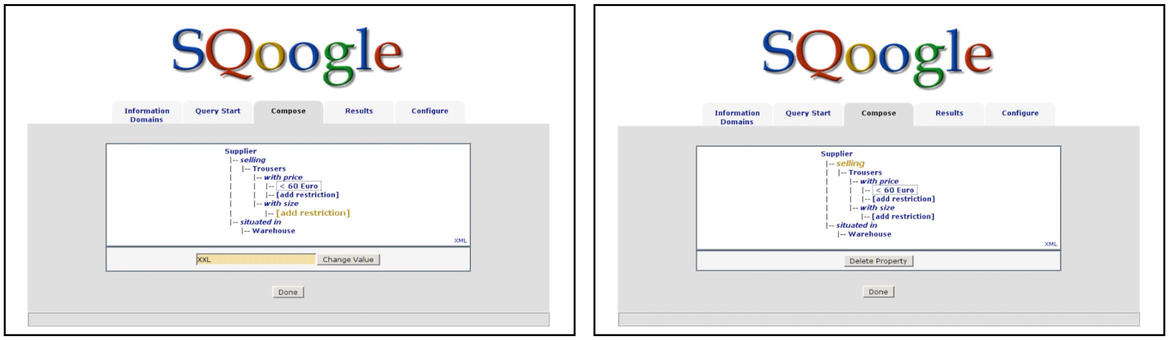
add a property

Figure 3.3: The Information Domain and the Compose Panels.

There is also a fifth tab "Configure" which is meant only for administrative purposes. The user starts choosing the domain of interest ("Information Domains" tab) among a list of all domains available. In the tab called "Query Start" the user has to select an entry point, in other words the concept of the chosen domain that will be set as the query head. After this the "Compose" tab is entered which represents the Query Manipulation Pane. The user sees the entry point (selected in the previous tab) and from that one she can start building the query. It is followed by the possible operations the user can do: generalize or specialize, add or remove a compatible concept, replace it with an equivalent concept, and add or remove a property.

The query is represented as a tree (see Figure 3.4). As soon as the query grows, the user can change the focus selecting e.g. another concept, the restriction of a property. Depending on the property selected, it can have either a concept as range or a basic data type (e.g. a string, an integer, a float) which can be filled with a value (restriction). What follows is a table of all focusable objects followed by the operations the user can perform on them:

- concept → generalize or specialize, replace with equivalent, add a compatible, add a property, delete concept;
- property → delete property (the right part of the Figure 3.4);
- restriction → change value (the left part of the Figure 3.4).



```
Supplier
 |-- selling
 |    |-- Trousers
 |         |-- with price
 |         |    |-- < 60 Euro
 |         |    |-- [add restriction]
 |         |-- with size
 |              |-- [add restriction]
 |-- situated in
      |-- Warehouse
```

XXL

```
Supplier
 |-- selling
 |    |-- Trousers
 |         |-- with price
 |         |    |-- < 60 Euro
 |         |    |-- [add restriction]
 |         |-- with size
 |              |-- [add restriction]
 |-- situated in
      |-- Warehouse
```

Figure 3.3: The Query tree in the Compose Panel.

When query composition is over, the user can click the "Done" button and the "Results" tab is brought to foreground. The query is shown as in the previous tab, but now the user can only select the information she is interested to know. The selections represent the columns of the query result table. Columns can be added or removed from the table. When the result table is configured as desired, the search can be started clicking the "Search" button and the results are displayed. The described flow can then be reversed whenever the user needs to modify the query, choose another entry point, or select a different information domain.

Performing the experiment

The experiment session was carried out at University of Modena and involved the tree people above mentioned. Each user has a workstation. The Query tool is on line at the address http://frida.inf.unibz.it/sewasie/index.html.

This test is composed of two sub-sessions: the first for the two skilled people and the second for the unskilled user. For each sub-session, we design two tasks set to be perform by the users:

- skilled users tasks → to compose different queries (Query_1,4,9 for the User_1; Query_2,5,8 for the User_2, see the Appendix IV of [3]), and to read the results of the assigned queries, analysing them;
- unskilled users tasks → to propose thinking the queries that the user composes usually, constructing them using the Query tool, and to read the results of these queries, analysing them.

Before to perform the tests, we set up the training session, instructing users about the tasks to perform. During the performance, we observe the users and to collect the measures of metrics defined above. In this phase, we ask users to think aloud, by describing their intentions, expectations, and their problems. In particular:

Event description (Friday 04/02/2005)

09:00→9:30 : the engineers instruct the skilled users about modalities of the experiment and they introduce the main goal of the Query tool without describing the functionalities of the tool;

09:30→10:00 : the users interact with the tool to understand how it works. During this auto-training session, the engineers record with a camera relevant performances. After that, each subject was presented with tasks;

10:00→10:30 : while the first user perform the tasks, the engineers observe the session of test, and they record the users' utterance using camera (Think Aloud Protocol).
10:30→11:00 : the engineers propose to the users the designed questionnaire and users make it. The engineers collect questionnaires.
11:00→11:30 : while the second user perform the tasks, the engineers observe the session of test, and they record the users' utterance using camera (Think Aloud Protocol).
11:30→12:00 : the engineers propose to the users the designed questionnaire and users make it. The engineers collect questionnaires.
12:15→12:45 : the engineers teach the unskilled user about modalities of the experiment, and they describe the user the Query tool functionalities. After this training session, the subject was presented with Query tool tasks.
11:45→12:15 : while the user perform the tasks, the engineers observe the session of test, and they record the users' utterance using a camera (Think Aloud Protocol).
12:15→12:45 : the engineers propose to the user the questionnaire and the users compose it. The engineers collect the questionnaire.

Results

We calculated the complexity of queries defined by the unskilled users, in particular, in the Table 3.2, we highlight some characteristics of the queries performed in this section of the experiments

Query_i	Num Level	Num Node	Avr_num suc_per_node	Avr_num node_per_level	Complexity			
Query_1	2	2	1,00	1,00	0,30		🟨	Low Complexity
Query_4	3	4	1,25	1,33	0,80		🟦	Medium Complexity
Query_9	8	11	1,39	1,38	6,05		🟪	High Complexity
Query_2	3	3	1,00	1,00	0,60			
Query_5	5	8	1,39	1,60	2,45		🟩	User_1 skilled
Query_8	7	12	1,42	1,71	5,60		🟧	User_2 skilled
Query_10	3	6	2,13	2,00	1,20		🟨	User_3 unskilled
Query_11	4	6	1,44	1,50	1,70			

Table 3.2: Complexity of the performed queries.

For the queries performed by the end users, we show the values of the metrics describing the performance for the usability evaluation (see Table 3.3). These measures are calculated using the video recorded during the experiments session.

				Time spent		Num change focus		Num step		Num mistake		Num cancellation		Num QMP click
Utente_1		Query_1		0,40		0		2		0		0		0
Utente_1		Query_4		1,17		0		3		0		0		1
Utente_1		Query_9		12,20		1		10		1		1		2
Utente_2		Query_2		1,30		0		2		0		0		2
Utente_2		Query_5		9,00		1		4		1		1		2
Utente_2		Query_8		14,00		1		9		2		1		3
Utente_3		Query_10		8,30		0		4		0		0		2
Utente_3		Query_11		10,15		0		6		1		1		4

Table 3.3: Values of the metrics of the performed queries for each User.

The auto-training session of the skilled users turned out some usability aspects, in particular it is not very clear:

- the conceptual difference between *Add a compatible term* button and *Add a propriety* button (see the Figure3.2),

- why some proprieties are present two or more times in the *Add a propriety* list,
- why it is impossible clicking freely on the tab that represent the main activities to construct the query,
- why the system takes a long time to answer a query.

Different observations come up from the unskilled user that asked for:

- biggest fonts for the natural language query representation (in the text box);
- a method to compact the query manipulation pane;
- the possibility to customize the values in the add concept list.

The Figure 3.4 shows the histogram containing the results of the questionnaire assigned to the end users, (Non-domain expert users vs Domain expert users). In particular, we use a color code to identify the Non Domain Expert and the Domain Expert users.

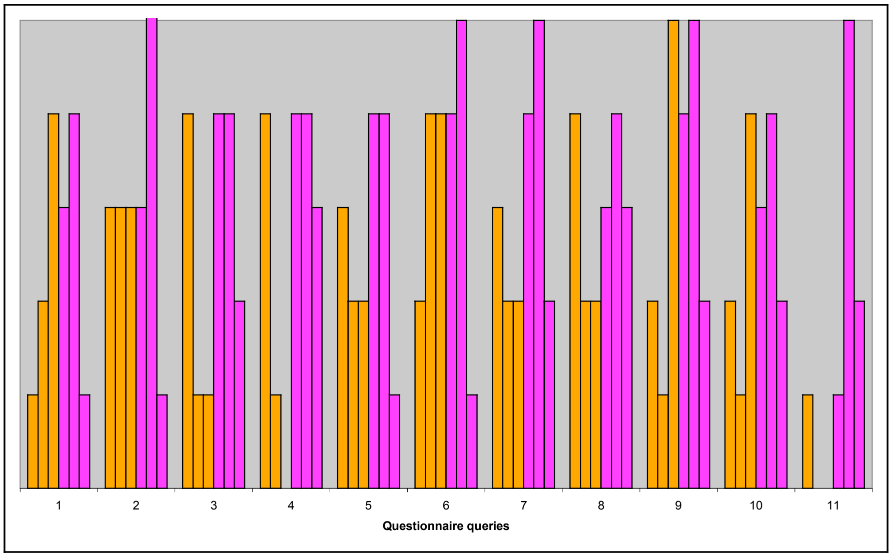


Figure 3.4: Histogram describing the results of the questionnaire.

We calculated the histogram in order to understand the relationship between the users' satisfaction and their domain experience.

Analysing the usability

Nevertheless, the observations of the skilled users, done during the auto-training session, these users after the brief training are able to perform the requested writing and reading tasks. In the case of unskilled user, easily, he proposes two valid queries, and the values of time spent to build these queries is relatively low (see Table 3.3); moreover, the number of mistakes is irrelevant. Therefore, we conclude that the overall functionality and philosophy of the Query tool interface are well understood by all users.

Moreover, we highlight that the value of time spent to construct queries is independent from the domain expertise of users. In fact, this performance measure is only function of queries complexity. In order to demonstrate that, we calculated the average values of time-spent to construct the low complexity queries, the medium complexity queries, and the high complexity queries for the two classes of users (Non Domain Expert = NDE, and Domain Expert = DE), collected in the Table 3.4, validating such results with an Anova test.

		Low	Medium	Hight
NDE		1,06	9,31	13,56
DE		1,05	8,87	13,10

Table 3.4: Average values of the users time-spent for each class of users.

Finally, it is worth noting that the questionnaire highlights that the user satisfaction to achieve the specific writing tasks is independent of the user domain experience; in fact observing the histogram in the Figure 3.4, we note that there are non significant gap between the values representing the average of result values of the Non Domain Expert users (orange color) and the same values of the Domain Expert users (pink color).
In our context, this aspect is a very strong point, because it demonstrates that the system can be used independently of the user domain expertise, in others words each class of user, described in the analysis of context [1], is able to construct queries using the interface of the Query tool.

4 Conclusions

In this paper, we have described the usability evaluation experiments of the Query tool system, done in the context of the SEWASIE project. The usability evaluation is an important step of the User-Centered Design Methodology (where users are involved from the very beginning both in the design and test steps), followed to develop the SEWASIE project. In fact, it is worth noting that the improvements of the Sewasie Query tool, described along the teaching users step, are the final result of an iterative testing process.

The main goal of our experiment was to demonstrate the easy of use of the Query tool independently of the domain user experience. Wee used the observational evaluation method and, in particular, the Think Aloud and Verbal Protocols. We described the evaluation experiments adopting a general user-based criteria schema.

The designed aspects (e.g., the model complexity) have been validated by a preliminary session of experiment performed with the non-domain expert users (five students). In particular, this test session highlighted that the query complexity has the same increasing behaviour of the metrics and that the users learned the system using it. Moreover, these results validated the query complexity mode and the questionnaire. Considered the positive results, we have performed the usability experiments session, starting from the training session, instructing users about the tasks to perform, and observing them in order to collect the required figures.

Concluding, the users were able to perform the requested writing and reading tasks. Therefore, we have concluded that the overall functionality and philosophy of the Query tool interface was well understood by all users. Moreover, we have observed that the value of time spent to construct queries is independent of the domain expertise of users, validating such results with an Anova test. Finally, the questionnaires have highlighted that the user satisfaction to achieve the specific writing tasks is independent of the user domain experience; this aspect is a very strong point, because it demonstrates that the system can be used independently of the user domain expertise, confirming that the Query tool system is usable by both end users (domain-expert users) and non-domain expert users.

References

[1] Evaluation of existing tools, description of the user centered interface design – SEWASIE Deliverable D6.1 – April 2003.

[2] First tests and analysis of test results – SEWASIE Deliverable D9.3 – April 2004.

[3] Evaluation of the system usability – SEWASIE Deliverable D9.5 – March 2005.

[4] Query tool online – SEWASIE Deliverable D6.4 – August 2003 – http://dev.eurac.edu:8090/sewasie/Init.do.

[5] R.B. Miller – Human Ease of Use Criteria and their Tradeoffs – IBM Report TR.

[6] B. Shackel, D.J. Richardson – Human Factors for Informatics Usability – Cambridge University Press.

[7] ISO – ISO 9126: Software product evaluation – Quality characteristics and guidelines for their use.

[8] http://www.useit.com/papers/heuristic/.

[9] http://www.cc.gatech.edu/computing/classes/cs3302/documents/cog.walk.html.

[10] Y.Yagita, Y.Aikava, A.Inaba – A Proposal of the Quantitative Evaluation Method for Social Acceptability of Products and Services –1997.

[11] Ericsson, A., & Simon, H. (1984) - Protocol analysis: verbal reports as data Boston: The MIT Press.

[12] http://jthom.best.vwh.net/usability/thnkalod.htm.

[13] T. Catarci, and S.K. Chang, and M.F. Costabile, and S. Levialdi, and G. Santucci – A Visual Interface for Multiparadigmatic Access to Databases – IEEE Transactions on Knowledge and Data Engineering – 1996.

[14] Cuff R. N. – On Casual Users – International Journal of Man-Machine Studies – 1980 – Vol 12 – pp 163-187.