Combining and Uniting Business Intelligence with Semantic Technologies

Acronym: CUBIST

Project No: 257403

Small or Medium-scale Focused Research Project

FP7-ICT-2009-5

Duration: 2010/10/01-2013/09/30

Directives for the Evaluation of the Use Case Prototypes

Abstract: This deliverable will provide guidelines and directives for the evaluations of the use case prototypes (see use case prototypes D7.3.2, D8.3.2 and D9.3.2, and use case specific evaluations D7.4.1, D8.4.1 and D9.4.1)

Type	Report
Document ID:	CUBIST D1.4.1
Work package:	WP1
Leading partner:	SAP
Author(s):	Frithjof Dau (SAP)
Dissemination level:	PU
Status:	final
Date:	02. Apr. 2013 (v 1.0), 12. Nov 2013 (v1.1)
Version:	1.1

Versioning and contribution history

Version	Description	Contributors
0.1	Fist draft with overview over methodologies	Frithjof Dau (SAP)
0.2	First complete draft finished	Frithjof Dau (SAP)
0.3	Addressed feedback SAS	Frithjof Dau (SAP)
0.4	Addressed feedback and concerns from HWU	Frithjof Dau (SAP)
1.0	Final version	Frithjof Dau (SAP)
1.1	Addressed a comment from the CUBIST review meeting	Frithjof Dau (SAP)

Reviewers

Name	Affiliation
Emre Sevinç	SAS
Kenneth McLeod	HWU

1 INTRODUCTION 4

2 PURPOSE OF THE EVALUATION IN CUBIST 5

1.1 THE PROTOTYPE FROM A CONCEPTUAL POINT OF VIEW 5

1.2 EVALUATION STAKEHOLDERS 6

1.3 EVALUATION GOALS 7

2 METHODS 8

2.1 CRITERIA FOR THE CLASSIFICATION OF METHODS 8

2.2 METHODS 9

2.3 COMPARISON OF METHODS 16

3 CONSTRAINTS AND PREREQUISITES IN CUBIST 17

3.1 GIVEN CONSTRAINTS 17

3.2 PREREQUISITES FROM THE REQUIREMENT ANALYSIS 17

3.3 KIND OF INFORMATION NEEDED 19

4 GUIDELINES FOR THE EVALUATION 20

4.1 SELECTION OF EVALUATION METHODS FOR CUBIST 20

4.2 WORKFLOW OF THE EVALUATION 20

5 BIB 25

5.1 HEURISTIC EVALUATION 25

5.2 COGNITIVE WALKTHROUGH 25

5.3 INTERVIEWS 25

5.4 QUESTIONNAIRES 26

5.5 THINK ALOUD 26

5.6 GENERAL 27

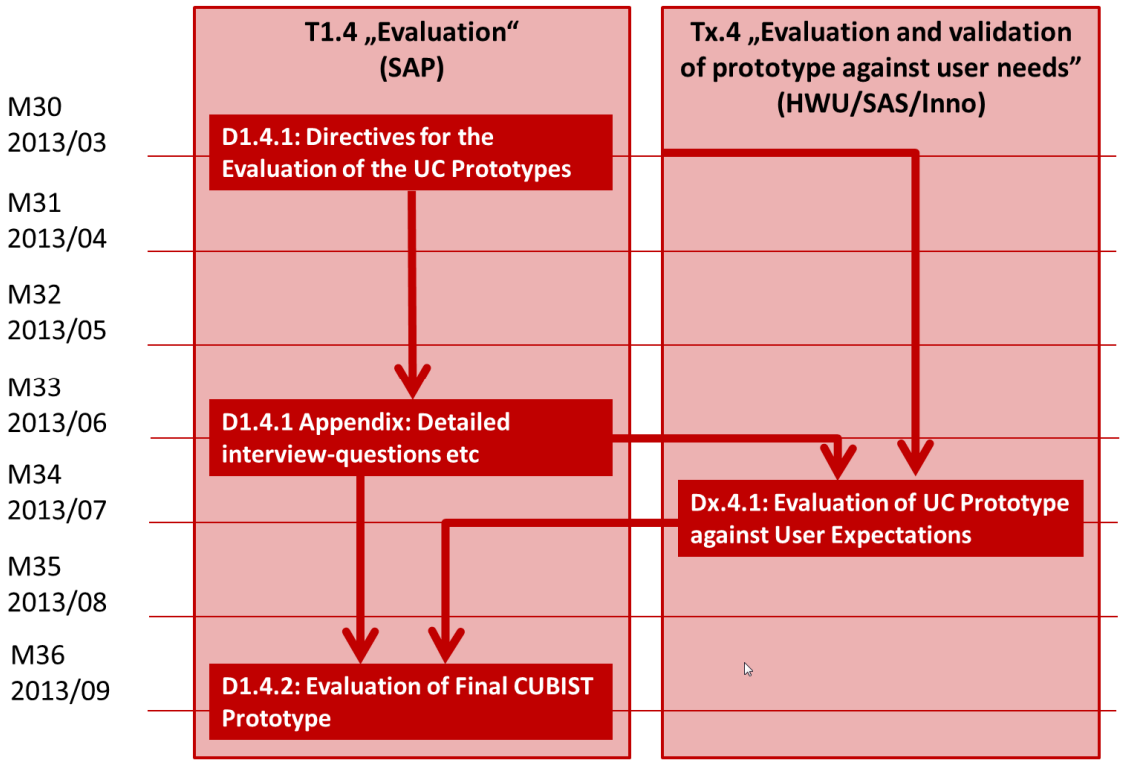
1 Introduction

This document provides the guidelines to be followed and the criteria to be to be observed during the evaluation of CUBIST Use Case prototypes (see use case prototypes D7.3.2, D8.3.2 and D9.3.2 "Use Case Prototype, v.2"). It describes the evaluation methodology and organization of all of the related use case specific evaluations. i.e. D7.4.1, D8.4.1 and D9.4.1 ("Evaluation of Use Case Prototype against User Expectations"). The outcome of these deliverables will in turn be summarized in a use case independent evaluation (D1.4.2 "Evaluation of Final CUBIST Prototype"). (This approach is similar to the requirement analysis, see D1.1.1 "Directives for the Requirement Analysis in the Use Cases).

In this document, we aim at:

1) Setting the general purpose of the evaluation
2) Setting the methods to be used in the evaluation
3) Describing the general methodology and approach of the evaluation in CUBIST.

This document **does not intend** to provide detailed interview-questions, detailed questionnaires or similar means which will actually be used in the UC evaluations. Instead, it serves as guidelines to create them, and the actual questions will be created as additional document latest in M33, so that they can be used in the actual UC evaluations. This appendix will be added to D1.4.2. The overall workflow of the CUBIST evaluation is given in the diagram below.



2 Purpose of the Evaluation in CUBIST

From a CUBIST-internal point of view, one can consider CUBIST as a software-development project with the goal to develop a CUBIST prototype which essentially meets the requirements which have been gathered in the starting phase of the project from the use case partners. This point of view as such is correct, but it does not reflect the intrinsic nature of the CUBIST project: it is a research project, and the CUBIST prototype is, as the name says, rather a prototype and not a final product, targeting a limited scope of the intended ideal solution. Thus the goal and the conduct of the project CUBIST is quite different from a product development, and the evaluation of the prototype has to address the address the research-specific aspects of CUBIST as well.

In the next section, we provide a short bird's eye perspective of the distinctive features of CUBIST, which serve as a first step on what has to be evaluated. In the following section, we provide the purpose of the CUBIST evaluation, which will turn out to evaluate the prototype from two different angles.

1.1 The prototype from a conceptual point of view

The CUBIST prototype aims at satisfying different kind of information needs:

1) **Factual Search:** Finding and exploring specific information, like a single fact or a list of objects which satisfy specific, information-need-relevant, criteria.
2) **Explorative Search:** Explore facts and entities when there is no precise information need, but the need to inform oneself about some facets of the domain of discourse
3) **Visual Analytics:** (Visually and interactively) analyze aggregated information to get a more condensed view on (subsets) of the data.

As described in the CUBIST Description of Work and as the project name already indicates, the focus of the research in CUBIST and the CUBIST prototype is on Visual Analytics. The evaluation will reflect this by considering all three kinds of information access, including their interplay, with an emphasis on evaluating the Visual Analytics.

In CUBIST, the main means for conducting analytics is based on a mathematical theory called *Formal Concept Analysis* (FCA). Analytics based on FCA are quite different from traditional BI analytics: The focus in FCA is not quantitative data analysis (aka "show me the numbers"), but on qualitative data analysis (aka "show me dependencies and meaningful clusters", so-to-speak).

Using FCA as main BI means in CUBIST has several impacts:

1) The workflow for the user in order to analyse data is quite distinct.
    a. In the first step, the user selects a set of entities (objects) to be analysed, and she selects some attributes (which are not necessarily attributes of the entities

to be analysed, but which can be attributes of different kind of entities) along the entities are to be analysed.

b. In a second step, the user prepares the Visual Analytics by defining for each attribute selected in the first step how its values shall be "clustered" in the Visual Analytics.
c. Finally, in a third step, different kind of FCA-based BI-graphs are shown to the user, which can be interactively and visually explored.

Due to the distinctive nature of this workflow, it has to be analysed on how effective and efficient users can deal with it, and on how well it is anticipated by the users.

2) The Visual Analytics means of CUBIST are quite distinct. CUBIST offers various kind of diagrams and metaphors which are uncommon in traditional BI-tools, like so-called Hasse-diagrams or sunburst diagrams for analysing the data, and matrices and radial diagrams for analysing implications within the data. Moreover, the Visual Analytics in CUBIST provide comprehensive interaction means like filtering data, drilling into data, linking-and-brushing facilities between different diagrams, and the like. Obviously, these Visual Analytics means have to be thoroughly evaluated.
3) Finally, the kind of (analytical) information which can be obtained from the Visual Analytics is quite different to what can be read of from traditional BI diagrams.[1] So it has to be evaluated for which kind of analytical questions CUBIST outperforms existing solutions, and for which kind of questions existing solutions are sufficient.

1.2 Evaluation Stakeholders

Generally, the evaluation of CUBIST aims at investigating whether CUBIST meets the needs of its stakeholders. For the evaluation, we can distinguish between two types of stakeholders:

1) The use case partners in CUBIST, being interested whether CUBIST meets their use case specific requirements, as they have been outlined during the requirement analysis phase in the beginning of the project.
2) Technical partners, as well as the EC, being interested on how CUBIST goes beyond the current state-of-the art, and being interested in the business relevance and exploitation opportunities of CUBIST.

1.2.1 CUBIST evaluation for Use Case Partners

From the perspective of a use case partner, the main goals of the evaluations are:

[1] A short discussion on this issue is provided in the external CUBIST Wiki, namely here: http://wiki.cubist-project.eu/index.php?title=Different_types_of_Analysis:_Bar_Chart,_Graphs,_FCA .

1) Providing an assessment of the of system's functionality extent. In particular, it has to be investigated whether CUBIST meets the requirements of the use case.
2) Answering the question on how the CUBIST user interface affects the user. It has to be investigated how efficient and effective users can deal with CUBIST, as well as the user's attitude towards the system.
3) Identify specific problems of the prototype. Such problems might be unexpected results, and causes of user confusion, and negative aspects of the UI design.

The evaluation from this perspective will focus on rather specific questions and tasks.

1.2.2 CUBIST evaluation for technical partners and EC

For the technical partners as well as for the EC (as funding) body, the main question for CUBIST is whether it goes beyond the current state of the art. Evaluating the progress beyond the state of the art in CUBIST is not a purely research question: the evaluation should lay the path for a prospective exploitation of the CUBIST results.

Compared to the evaluation from a use case partner's perspective, the focus of the evaluation from this perspective is rather general. E.g. instead of scrutinizing how users deal with given, CUBIST-tailored tasks, here it has to be evaluated for which *kind* of tasks CUBIST is best suited, and for given classes of task, how CUBIST compares with existing solutions.

1.3 Evaluation Goals

To summarize, the purpose of the evaluation is two-fold:

1) Evaluate CUBIST from a user's perspective, with a focus on CUBIST-specific requirements (as obtained in the beginning of the project) and tasks. This evaluation does only take CUBIST into account and can thus be considered an Intrinsic evaluation:
2) Evaluate CUBIST from a research- and even business-perspective, investigating where and how CUBIST goes beyond the current state-of-the art. This evaluation focuses on general tasks and will compare CUBIST with existing systems, thus it can be considered an extrinsic evaluation.

2 Methods

In this chapter, the most common evaluation methods are introduced, and common criteria for classifying them are provided.

2.1 Criteria for the classification of methods

In this section, some well-known classifications of evaluation methods are provided which will help to choose appropriate methods for the evaluation in CUBIST. Essentially, the methods are classified along three dimensions:

1) Who: From whom are the results of the evaluation obtained?
2) Where: Where does the evaluation take place?
3) When: For which phases in the development cycle of a product a method is appropriate?

These dimensions will be elaborated in the next subsections.

2.1.1 Who?

In an evaluation, data is gathered from people who reflects their judgment or their usage of the evaluated product. A first classification of methods takes into account whether those people are (prospective) end users of the product or experts.

- Usability testing methods are methods where the usability of the interface is evaluated by testing it on real users belong to the class of. Those methods provide information on how real users use the system and focus on on measuring a product's capacity to meet its intended purpose. Those techniques include: experimental methods, observational methods, query techniques (e.g., questionnaires and interviews), or physiological monitoring methods (e.g., eye tracking). They can be conducted in the laboratory and/or in the field.
- usability inspection methods are methods where some experts (i.e. usability experts, domain experts, cognitive psychologists) inspect and evaluate a user interface. Those methods can generally be used in early phases of the development process by evaluating prototypes or specifications for the system that can't be tested on users. They are generally considered to be cheaper to implement than testing on users.

Apart from these two categories, there is a third category of model-based evaluations (e.g. review-based evaluations), which is not discussed here.

2.1.2 Where?

The next classification is based on the place where an an evaluation is conducted.

1. Laboratory tests take place in dedicated laboratories for the evaluation. They have the advantage of being conducted in an uninterrupted environment on dedicated equipment. On the other hand, these tests lack often context. They are appropriate for if the system location is dangerous or impractical, for single user systems, and if a controlled manipulation of use is needed..

2. Field studies take place in the natural environment of the user, thus they have the advantage of the user context to be retained (though observation may alter it). On the other hand, if a user acts in his field, the danger of distractions from the evaluation is given.

2.1.3 When?

Finally, there are two types of evaluations[2], depending on the phases of the product development the evaluation method is appropriate for.

1. Formative usability evaluations are evaluations which test to answer UI questions during the design process.

2. Summative or acceptance usability evaluations, are evaluations which demonstrate that UI works when product is complete and which provide a final seal of approval on UI.

These types should not be considered to be exclusive: a number of evaluation methods can be used both in early and in late (even after product completion) phases of the product development.

2.2 Methods

There is a plethora of evaluation methods, and a comprehensive overview over existing methods goes beyond the scope of this document. In the next subsections, the most prominent methods are shortly introduced. Information about their conduct, about when, where and with whom they are carried out, and their pros and cons are provided.

---

[2] There exists moreover the notion of participatory design methods, which refers to the idea of the user being an active member in the design team. Thus it is context and work oriented rather than system oriented and has a collaborative and iterative nature.

2.2.1 Expert and review-based evaluation methods

Expert reviews, being probably the most informal method, is the first method we introduce here which belongs to the class of usability inspection methods. As the name indicates, this method relies on bringing in experts with experience in the field (possibly from companies that specialize in usability testing) to evaluate the usability of a product. In expert reviews, the experts just go through the prototype, try it out, and write down any problems. Thus there is no need to have any use tasks, or scripts (but having some data in the system helps). The experts compare the products to other products they have used, adopt a critical perspective, and detect any obvious problems. For this method, it can be extremely effective to have a UI expert and domain expert go through it together.

2.2.2 Heuristic evaluation

Heuristic evaluations belong to the class of usability inspection methods. In this method, few (e.g. five) experts (UI experts as well as domain experts) evaluate the interface and judge whether each element of it are compliant with recognized usability principles, namely the heuristics. It is similar to expert evaluations, but the heuristics aim to cope with the fact that in expert evaluations, the experts often forget to “look out” for common problems when doing an expert review. Heuristic evaluations “debugs” the design of an interface, as evaluators systematically evaluate it. They either go through every screen for one heuristic, then repeat for next heuristic, or for each screen, apply all heuristics. In each step, they rate the seriousness of problems identified.

There are different sets of heuristics; amongst them Jakob Nielsen's heuristics are probably the most-used usability heuristics for user interface design. The heuristics as published in Nielsen's book Usability Engineering are as follows :[3]

- Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- Match between system and the real world: The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- User control and freedom: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- Consistency and standards: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

[3] List taken from http://www.nngroup.com/articles/ten-usability-heuristics/

- Error prevention: Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- Recognition rather than recall: Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- Flexibility and efficiency of use: Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- Aesthetic and minimalist design: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- Help users recognize, diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Heuristic evaluations are fast and cheap to be conducted and deliver results of good quality, but as the experts are no real users, the heuristics will not capture all problems which might later occur when the system is used by real users.

2.2.3 Cognitive walkthrough

Cognitive walkthrough belongs to the class of usability inspection methods. It is based on the idea that users often prefer to learn a new system by accomplishing user-specific tasks instead of reading manuals. Due to this idea and in contrast to heuristic evaluation, this method is a *task-specific* approach, focusing on how easy it is for users to carry out tasks with the system.

Cognitive walkthrough as carried out by one or more experts (e.g. experts in cognitive psychology, software developers, or people from marketing) who inspect a user interface by going through a set of tasks and evaluate how easy the system is to learn and understand. The user interface is often presented in the form of a paper mock-up or a working prototype, but it can also be a fully developed interface. Though it can be best applied in the design phase of the product development, this method spans the whole development time of a product including coding, testing and deployment phases.

To perform a walkthough, the following four things have to be set in the preparation phase:

a) A description of the prototype of the system, focusing on its interface.

b) An indication of who the users are and what kind of experience and knowledge the evaluators can assume about them.

c) A description of the tasks the user is to perform on the system.

d) A complete written list of the actions needed to complete the task with the given prototype.

Given this material, in the analysis phase the experts walk through through the steps of a task as a group, asking themselves a set of questions at each step. These questions comprise:

- Will the users be trying to produce whatever effect the action has?
- Will users be able to notice that the correct action is available?
- Once users find the correct action at the interface, will they know that it is the right one for the effect that they are trying to produce?
- After the action is taken will users understand the feedback that they are given?

In the follow-up phase, a analysis is conducted which focuses on goals and knowledge and evaluates whether the design leads the user to generate the correct goals, and a final report with the analysis results and suggestions for improvements is created.

The advantages of cognitive walkthough are its ability to generate results quickly with low cost, especially when compared to usability testing, as well as the ability to apply the method early in the design phases, before coding has even begun. On the other hand, the walkthrough does not test real users on the system, and it will often identify many more problems than you would find with a single, unique user in a single test session.

2.2.4 Interviews

An **interview** is a guided conversation between an interviewer and individual users. This method is especially suitable for product innovation, as users can give detailed feedback on early drafts of the product, and the product acceptance in the target group can be tested very early. Moreover, the interview is also one of the best means to understand the current use of the product and to improve it. Due to the direct communication between the interviewer and the user, complex and detailed, qualitative information can be collected. Moreover, in contrast to other survey methods like questionnaires, the interviewer can directly address the user's individual concerns, and mistakes and misunderstandings can be quickly identified and cleared up. Interviews can be used in all phases of the product development. It is possible to conduct interviews in dedicated facilities (in a laboratory) or remotely via telephone or other telecommunication means. It should be possible to record the interview.

One can roughly distinguish between open and structured (or standardized) interviews. In open interviews, only minimal guidelines (in extreme cases, only the survey topic) are used. Everything else is left to the interviewer and the conversation process. In these interviews, the

information gathered is of qualitative natures. In completely structured interviews, all questions are defined beforehand, including a predetermined order, and sometimes even specific answer categories are provided. Here quantitative information is collected. Hybrid versions of interviews with a set of predefined questions as well with having open questions are often used.

In the planning phase of an interview, the required information is considered, and an interview schedule is prepared. This can be understood as a set of topics which is needed in order to obtain the requested information, and an order in which those topics will be covered. For each topic, one has to define on how the interviewer will ask for the information needed, and explanations of the topics have to be prepared (in case the user does not understand the interviewer's questions). The more the interview is structured, the more the topics are broken down into series of subtopics, and the more finegrained the questions are defined beforehand.

When an interview is conducted, there are often four typical phases:

1) The "nurturing" phase. This is an initial warm-up phase when the interviewer and the user introduce themselves and talk briefly about neutral topics to establish themselves.
2) The "energising" phase. Here the area of discourse, and any existing problems are identified.
3) The "body" of the interview. This is the peak phase of activity, where the interviewer is continually probing, ideally asking open-ended questions about issues to understand the range of responses the users produce. It is important at this stage for the interviewer to remain analytical and neutral, and to check before its ending whether all the topics have indeed been covered.
4) The "closing" phase. Summaries may be given as to what has taken place. Subsequent actions are noted, and future planning is made.

Finally, the results are the interviews are laid down in reports. Due to the unstructured nature of the information collected in interviews, this information is easily misinterpreted or censored. In order to cope with this, the interview responses or notes are split into a set of simple propositions, using the informant's own words as much as possible. These propositions can then become the input to a content analysis activity. Notes, recordings etc. can be added to an appendix of the reports.

Interviews are useful for identifying possible areas for more detailed analysis. The data gathered provides information directly from the users on general rules and principles and is faster than observational methods. With open interviews, one can gather open-ended results which are not possible with structured interviews or questionnaires, which is helpful for products with new and possibly surprising features. On the other hand, there is room for considerable bias in what questions are asked and how the answers are interpreted. Moreover, the interviewer must have sufficient domain knowledge in order to know what questions to ask. Finally, what people say often differs from what they really do, and due to the so-called

Hawthorn Effect, users tend to please the please the evaluator, which might distort the interview results.

2.2.5 Questionnaires

Questionnaires are one of the most common and popular tools to gather data from a large number of users. This method can be addressed to a larger group of users than interviews, and it can be undertaken and analysed relatively easily. A questionnaire is essentially a series of standard questions in a structured format. The more a questionnaire is standardized, the larger number of closed questions it contains. In most cases, the user is given predefined statements (descriptors) from which to choose, usually based on a given scale (see below). In case of a less standardised questionnaire, the respondent is free to formulate his or her answers as he or she wishes, as there are more open questions. In case of more formal questionnaires and in contrast to interviews, in this method quantitative data is gathered.

In formal questionnaires, scales are used to collect data. One can distinguish between

- nominal scales, where different values (attributes) of the scale are unordered,
- ordinal scales, where the values are meaningful ordered, and
- interval scales, where the values are meaningful ordered and clustered into intervals.

The number of possible values for ordinal scales can range from three to ten. 5-point scales are the the most common, though 7-point scales are becoming more prominent. Examples of scales are given below.

<table>
<tr>
<td>Acceptance (7-point ordinal scale)
Not at all acceptable — Neutral — Very acceptable
1 2 3 4 5 6 7</td>
<td>Ease (7-point ordinal scale)
Very Difficult — Neutral — Very Easy
1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Agreement (5-point ordinal scale)
1 (Strongly Agree) | 2 (Agree) | 3 (Neutral) | 4 (Disagree) | 5 (Strongly Disagree)
○ ○ ○ ○ ○</td>
<td>Marital status (5 point nominal scale)
□ married □ widowed □ divorced
□ separated □ never married</td>
</tr>
<tr>
<td>Yes/now (2 point nominal scale)
□ yes □ no</td>
<td>Age (4 point interval scale)
0-18 | 19-40 | 41-65 | 66-90
o o o o</td>
</tr>
</table>

Figure 1: examples for scales

A questionnaire can be undertaken remotely (e.g. by sending the survey to users via email) or in face-to-face interviews. Questionnaires are best suited to the observation of the results and

impacts of a product, thus they are likely to be conducted in the end phase (or even after finishing) of the product development.

A good questionnaire can be a powerful tool to inform your evaluation, but a poorly designed questionnaire can make life difficult for both the users who have to complete it as well as to the evaluators that have to analyse the data. Moreover, this method is characterized by small flexibility (for example, in contrast to interviews, concerns and questions of the users which show up during the execution of a questionnaire cannot be addressed). The most important issues can be omitted and disregarded if the questionnaire contains no questions referring to these particular issues.

2.2.6 User Testing/Thinking Aloud

Think-aloud is a usability testing method used to gather data in in product design and development. Essentially, in a think-aloud session, users are given a set of specified tasks. As they perform those tasks, the users are have to say whatever they are looking at, thinking, doing, and feeling, which enables observers to see first-hand the process of task completion. The observers have to note down in an objective manner everything that users say, without attempting to interpret their actions and words. Sessions re usually recorded, so that it is possible to go back and refer to what users did and how they reacted. The purpose of this method is to make explicit what is implicitly present in subjects who are able to perform a specific task.

For the utterances of the users, one can further divide this method into two different experimental procedures. In the concurrent think-aloud protocol, data is collected during the execution of the tasks, whereas in the retrospective think-aloud protocol, this is done after task execution.

In the preparation phase of think-aloud evaluation, the evaluator creates a set of tasks the user has to conduct. During execution, the user performs those tasks and thinks aloud. The execution is recorded and the evaluator is taking notes. Quite often the evaluator poses during the test additional interview questions in order to obtain additional information about the mental model of the test user. A complex question how strongly the evaluator can interact with the test person, e.g. in case of technical problems. Finally, the evaluators and the users discuss their findings, and it is analyzed which aspects of the product are to be positively judges, and which problems most often occured and how serious they are. The findings are taken down in a report.

According to Jakob Nielsen, the method has several advantages. It is cheap, as no special equipment is needed, it is robust, as it is not as strongly riped with methodology problems like other methods, it is flexible, as in can be used at any stage in the development lifecycle, it is convincing and easy to learn. Most importantly, it is a “window on the soul, letting you [the evaluator] discover what users really think about your design.” And one can “learn why users

guess wrong about some parts of the UI and why they find others easy to use." On the other hand, the methods does not provide quantitative data, thus it does not "lend itself to detailed statistics". The think-aloud situation is a unnatural situation for the evaluated users, and they tend to filter their statements.

2.3 Comparison of methods

Below a table-based summary of the methods is provided.

	Who	Where	When	Measure	Information
cognitive walkthrough	experts	lab	throughout	qualitative	low level
Expert review	experts	lab	throughout	qualitative	high level
Heuristic evaluation	experts	lab	throughout	qualitative	high level
Thinking-aloud	user and evaluator	lab or field	end	qualitative	low level & high level
Interview	user and interviewer	lab or telephone	throughout	qualitative	high level
questionnaires	user	field (or lab)	end	quantitative	low level

3 Constraints and prerequisites in CUBIST

In this chapter, we set the stage for choosing evaluation methods by summerising CUBIST-specific constraints, and by shortly discussing on which kind of information we aim at collecting during the evaluation.

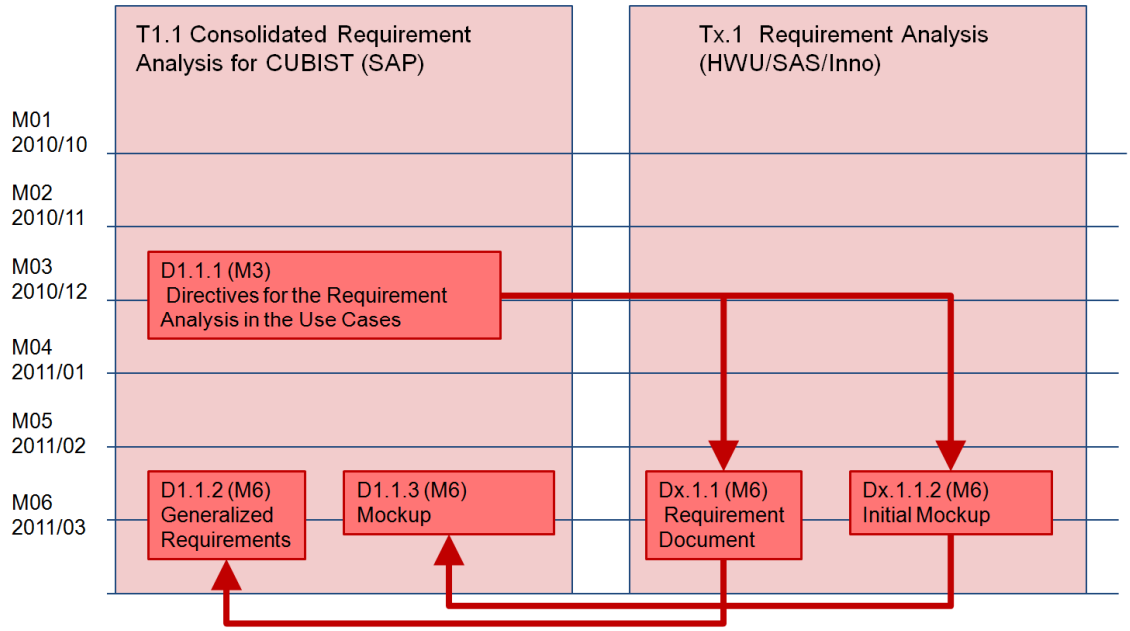
3.1 Given constraints

We face the following constraints for the evaluation in CUBIST.

1) **Available ressources (time, subjects, equipment, expertise):** The use case partners will provide test users with different levels of expertise, based on the personas as they have been gathered during the requirement analysis (see next section). As CUBIST is a web-based prototype, no special equipment (apart from a laptop with a browser) is needed. For remote evaluations and for recording evaluation sessions (see below), we can use SAPConnect, being an integrated conferencing service with audio and video for internal and external participants hosted by SAP.
2) **Development cycle when evaluation is conducted (design vs implementation):** The evaluation takes place at the end of the project, when use case instantiations of the second version of the CUBIST integrated prototype are available. It though should be taken into consideration that the CUBIST-prototype is a research-prototype with a limited scope of the intended ideal CUBIST solution. Thus even though the evaluation will be conducted at the end of the project, we still am at finding short-comings of the prototype, possible improvements, and the like.
3) **Evaluation style (laboratory vs field):** Due to the geographical spread of the users in the use cases (Uk, Uk, Belgium) and the evaluators (SAP, Germany) we cannot bring users to a dedicated test environment. Moreover, as we aim for comparative studies with existing solutions, which can be use-case-specific and thus only exist in the field. For this reason, we will conduct a field study.

3.2 Prerequisites from the requirement analysis

In the beginning of CUBIST, a requirement analysis has been conducted. Concerning the workflow, we first have provided in D1.1.1 Directives for the requirement analysis. Based on the directives, each use case partners has conducted a use case specific requirement analysis, which resulted in Deliverables “Dx.1.1 “Requirement Document” and Dx.1.2 “Initial Mockup” (x=7,8,9). These use case specific documents have been aggregated and generalized to a use case independent list of general, formal requirements in D.1.1.2 and a general CUBIST mockup in D1.1.3. This workflow is depicted in the next figure (repeated from D1.1.1).



Contentwise, we gathered personas, utilization scenarios, formal requirements, and mockups during the requirement analysis. It is obvious that these information assets should be taken into account during the evaluation. This is discussed next.

1. **Personas:** Personas are ficitional and prototypical persons which represent typical users of the CUBIST system. Per Use case, we have collected the following personas:
    - HWU: an biologist, an computational biologist, an bioinformatician and a software developer. For HWU it is important to note that biologist and computational biologists, as well as the software developer and bioinformatician, are resembling, thus HWU has provided two instead of four utlilization scenarios (see below).
    - SAS: a payload operator, a ground controller, an operations engineer, a prinipal investigator and a scientists. Among those, the payload operator is far the most important user for the SAS use case, and SAS has thus provided for this persona two different utilization scenarios.
    - INN: a user, a decision maker, an account manager and a super admin. Similar to HWU, INN has provided twoutilization scenarios, each of them for two personas simultaneously (one for the user and the decision maker, and one for the account manager and the super admin).

    As personas represent the typical users of the CUBIST system, we aim at evaluation real users which best resemble the personas. Anyhow, it is not feasible to have a one-to-one correspondence between personas and test users: First, as it can be best seen in

the SAS use case, not all personas are equally important to the use case partners, abd moreover, as can be seen in the HWU and INN use case, it is sometimes sufficient to provide for *pairs of* personas one corresponding test user.

2. **Utilization Scenario:** utilization scenarios represent typical days of personas. Each utilization scenario comes in two forms: An ”as-is”-scenario which describes the days in the life of a persona as they exist today without a CUBIST system, and an envisioned ”to-be”-scenario which reiterates the ”as-is”-scenario under the assumption that a CUBIST system is in place, thus showing the benefit CUBIST will provide. The utilization scenarios can be thus be used in order to define tasks to be perfomed by test users during the evaluation, and they can be used for the extrensic evaluation of CUBIST, i.e. comparing CUBIST to existing (BI-) solutions.
3. **Mockups:** Mockups have been a first step towards the envisioned user interface. The initial development of the prototype have been aligned to the mockup, but during the further course of CUBIST, it turned out that for the support of “self-service-BI” in CUBIST, a major revision of the frontend and the underlying CUBIST workflow was needed. Thus the initial mockups will not be used during the evaluation.
4. **Generalized requirements:** In D1.1.2 we collected a total of 67 formal and general requirements, amongst them 22 mandatory requirements, 38 desirable requirements, and 7 optional requirments. Most of the requirements can be directly checked (e.g. “CUBIST must be web-based). The list of generalized requirements will be checked directly with the use case partners in the evaluation and might further serve as one starting point for questionnaires.

3.3 Kind of information needed

1. **Qualitative and quantitative measures:** As described in section “Evaluation Goals”, the evaluation serves two purposes. For both purposes, In the evaluation of CUBIST from a user’s perspective, we will go both for qualitative and quantitative measures, whereas in the evaluation from the research /and business) perspective, we essentially target qualitative measures only. For this reason, the set of methods to be chosen have to cover both kinds of measures.
2. **Low-level or high level information:** The CUBIST system is a prototype and not a product, and it implements only a limited scope of the intended ideal solution. The evaluation does not target to detect the gaps to the ideal solution, but aims at answering research and business questions on whether the prototype (and even the envisioned ideal solution, as it can be anticipated from the implemented prototype) fulfills the intial hopes and promises, as described in the Dow, and how CUBIST compares to existing solutions. For this reason, even though we intend to gather some low-level-information, the focus will be gathering on high-level-information during the evaluation.

4 Guidelines for the evaluation

4.1 Selection of evaluation methods for CUBIST

As discussed in Section 3.1"Given constraints", we target at an evaluation with endusers in the field at the end of the CUBIST project. Thus the methods "cognitive walkthrough", "expert review" and "heuristic evaluation are not appropriate for the evaluation and are thus dismissed.

On the other hand, as discussed in Section 3.3"Kind of information needed", we have to collect both qualitative and quantitative measures, and we have to collect low level and –more importantly- high-level information. For this reason, we decided to utilize a combination of the following methods:

- **the task-based thinking aloud method** (this will inform both about low-level-task accomplishment in CUBIST as well as about high-level benefits of CUBIST),
- **interviews** (they will deepen the high-level evaluation of CUBIST), and
- **questionnaires** (which will serve for collecting quantitative measures, both for low-level and high-level information in the evaluation).

4.2 Workflow of the evaluation

4.2.1 Preparatory phase

4.2.1.1 Go through formal requirements

As stated in the last chapter, in a very first step, we plan to go through the list of formal requirements directly with the use case partners. For each requirement, it will collaboratively decided whether the requirment

1) is fullfilled (hence it needs not to be adressed during the further evaluation)
2) is meanwhile considered obsolete or unimportant (hence it needs not to be adressed during the further evaluation)
3) has to be checked during the evaluation.

The list of requirements which need to be checked will serve as one input for the questonnaires.

4.2.1.2 Find test users

Each UC partner has to provide two test users. As discussed in the past chapter, those test users must be based on the personas of the respective use case. More precisely:

1) HWU has to provide one test user being an HWU a biologist or a computational biologists, and one test user being a oftware developer or bioinformatician.
2) SAS has to provide at least one payload operator as test user, and choosing the second test user is left to SAS (it might be based on a different persona, or another payload operator).
3) INN has to provide two test users, one being a user or a decision maker, and anothe one being an account manager or a super admin.

4.2.1.3 Prepare Training Material and Provision of Access to Prototype

The test users are not expected to start from scratch with the CUBIST prototype: instead, we assume that they have gained some experience with the prototype beforehand. In order to support them with this, at least one month before the actual evaluation starts,

1) a training-video (duration: ca. 15 min) which explains the essential functionalities of CUBIST will be created and made accessible to the test users, and
2) access to the CUBIST prototype will be granted to the test users.

We assume that the test users will try to replicate the training-video using the prototype, and thus learn how to use it.

4.2.1.4 Prepare Tasks

The test users are asked to list three tasks/information needs which they consider to be easy (or relatively easy) accomplishable with CUBIST, but which are hard or impossible done with their current tools. These tasks should resemble tasks as taken down during the requirement analysis in the utilization scenario. In case those scenarios are dismissed and different kind of tasks are provided, a short a rationale for the departure from the utilization scenarios should be given. The test users are asked for alternative flows like unexpected events (e.g. information not found, dependencies with other resources/people), and they are requested to evaluate the effort/time needed for each task. In the .execution phase, those tasks will be conducted with the CUBIST prototype.

4.2.1.5 Prepare Interviews and Questionnaires

As stated in the last section, the interviews are intended to deepen the high-level evaluation of CUBIST and will be conducted after the tasks (see next subsection). Thus first of all, we will detail out the kind of information needed, and prepare a list of topics required to obtain that information. Moreover, we will decide on an order of the topics, i.e. an interview schedule will be created. For each topic, questions or question templates will be prepared, as well as explanations of the topics (in case the test users need them, e.g. because they do not understand the questions). The interviews will be used to collect knowledge about new insights that the test users they gained using the prototype). A trial run of the interview will be

conducted in order to check whether the envisioned interview schedule and timeframe is feasible.

Similarly to the interviews, for the questionnaires a list of topics have to be prepared first. As the questionnaires will be used to gain quantitative measures, both for low-level and high-level information in the evaluation, parts of the questionnaires will complement the topics of the interviews, and others will directly address specific features of the prototype. Moreover, the questionnaires will cover those formal requirements which have been selected in the first part of the preparation phase as still needed to be evaluated. Finally, we note that the user *satisfaction* with the prototype (compared to the users e*ffectiveness* –i.e. the accuracy and completeness on how users achieve certain goals- and *efficiency* –i.e. the relation between accuracy and completeness-, which can be measures when users conduct their tasks during the task execution of the evaluation) can be best measured with the questionnaires.

Both the interviews (topics, questions, explanations) and the questionnaires (topics, questions) will be delivered in the appendix to this deliverable.

4.2.2 Execution phase

Generally, we will conduct the following order during the evaluation: 1) the task-based thinking aloud method, 2) interviews, and 3) questionnaires.

The think-aloud method and the interviews will be conducted remotely using SAPconnect and recorded (audio and screencast).

1) For the think-aloud-method, the users are asked to perform the tasks they have provided in the preparation phase and find the information they already know using CUBIST and their data. The evaluators will push the test users to think aloud. Apart from the recording, the evaluators will take notes during this phase.

   We expect this part to take a maximum of 15 minutes.

2) After the think-aloud evaluation, the evaluators will conduct the interviews with the users, based on the prepared interview schedule. The interviews will be transcribed to written form, which will be later be approved by the test users.

   We estimate not more than 10 minutes for the interviews.

3) Finally, the questionnaires will be handed to the users, which they can fill in remotely and send them back via email to the evaluators.

   We estimate not more than 10 minutes for filling the questionnaires.

In total, the execution of the evalutation should take for each user between 30 and 45 minutes.

4.2.3 Analysis phase

According to the ISO Standard 9241-11 there are three primary attributes that comprise usability: effectiveness, efficiency, and satisfaction. The can be explained as follows:

- Effectiveness:
  - Accuracy and completeness on how users achieve certain goals
  - Indicators include: quality of solution and error rates
- Efficieny:
  - relation between Accuracy and completeness (again for given goals) and resources spent in achieving them
  - Indicators include: Task completion time and the learning time
- Satisfaction
  - Users comfort with and positive attitude towards the use of the system
  - Can be measured by attitude rating scales

These attributes can be understood to cover additional attributes such as learnability and retention which are considered by usability experts loke Nielsen or Shneiderman. The next table shows how some of the more popular definitions of usability map to one another.

Table 1: The three main criteria of usability

ISO 9241-11	**Nielsen 1993**	**Shneiderman 1998**
Efficiency	Efficiency Learnability	Speed of Performance Time to Learn
Effectiveness	Memorability Errors Safety	Retention over time Rate of errors by users
Satisfaction	Satisfaction	Subjective satisfaction

As the indicators of these criteria imply, they can be quantitatively measured (though qualitative measures form them are possible as well), and they directly act on the tasks of the think aloud method. It should be noted that we will have different tasks in the different use cases, thus an *objective* measurement of efficiency and effectiveness is not feasible. Instead, we will measure the *subjective* efficiency, effectiveness and satisfaction, e.g. via using Likert scales in the questionnaires.

Most of the information gathered during the evaluation will be qualitative and high-level information which will be used to evaluate CUBIST from a research- and even business-perspective. WE plan to device a small taxonomy for types of BI-questions in order to

systematically compare traditional "number-crunching" BI and the novel, FCA-based BI conducted in CUBIST, so that we can work-out the pros and cons of both approaches. As unstructured and semistructured information (i.e. the recordings and protocols of the think-aloud sessions, and the interviews) will be the base of the research evaluation of CUBIST, it seems reasonable to support the analysis with tools for qualitative data analysis, e.g. NVivo from QSR International[4].

We will conduct traditional statistical means (e.g. analysis of variance) as well as FCA-means to analyse the data, thus the analysis results will come in form of standard BI graphs (e.g. standard charts like bar charts) as well as FCA-based graphs. As a core part of the analysis will be of qualitative nature, we moreover anticipate that key insights and results will come in plain textual form.

The results of the analysis will be provided in the CUBIST deliverable D1.4.2 " Evaluation of Final CUBIST Prototype".

[4] http://www.qsrinternational.com

5 BIB

5.1 Heuristic Evaluation

Bias, R.G. and Mayhew, D.J. (Eds.). Cost justifying usability. Academic Press, 1994, pp.251-254.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. Proc. ACM CHI'92 (Monterey, CA, 3-7 May), pp. 373-380.

Nielsen, J. and Landauer, T. K. (1993). A Mathematical Model of Finding of Usability Problems. Proc. INTERCHI '93 (Amsterdam NL 24-29 April),

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1–5 April), 249-256

Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, Communications of the ACM 33, 3 (March), 338-348

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY

5.2 Cognitive walkthrough

Lewis, C. Polson, P, Wharton, C. and Rieman, J. (1990) Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces Chi ’90 Proceedings pp235–242.

Rowley, David E., and Rhoades, David G (1992). The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure. Proceedings of CHI '92, 389-395.

Spencer, R. (2000) The Streamlined Cognitive Walkthrough Method, Working Around Social Constraints Encountered in a Software Development Company CHI 2000 vol.2 issue 1 pp353–359.

Wharton, C. Bradford, J. Jeffries, J. Franzke, M. Applying Cognitive Walkthroughs to more Complex User Interfaces: Experiences, Issues and Recommendations CHI ’92 pp381–388.

5.3 Interviews

Blomberg, J., Giacomi, J., Mosher, A. and Swenton-Hall, P. (1993) Ethnographic field methods and their relation to design. In: Schuler, D. and Namioka, A. (eds.) Participatory Design: Principles & Practices. New Jersey: Lawrence Erlbaum.

Fowler, FJ (Jr), Mangione, TW (1990) Standardised Survey Interviewing. Sage Publications, Newbury Park.

French, John R. P., "Field Experiments: Changing Group Productivity," in James G. Miller (Ed.), Experiments in Social Process: A Symposium on Social Psychology, McGraw-Hill, 1950, p. 82.

French, John R. P., "Experiments in Field Settings," in Leon Festinger and Daniel Katz(Eds.), Research Methods in the Behavioral Sciences, Dryden Press, 1953, p. 101.

Wrege, Charles D.; Bedeian, Arthur G. (2008). ". . . thumbs down for the Hawthorne Effect". The Psychologist 21 (11): 990.

5.4 Questionnaires

Gräf, L. (2002): Assessing internet questionnaires: The online pretest lab. In: B.Batinic; U.-D. Reips; M. Bosnjak; A.Werner (Eds.). Online Social Sciences. Seattle, WA: Hogrefe & Huber Publishers. S. 73-93

Kirakowski, J. and Corbett, M., 1993. SUMI: The software usability measurement inventory. British Journal of Educational Technology, 24(3), S. 210-212.

Laugwitz, B., Schrepp, M. and Held, T., 2006. Konstruktion eines Fragebogens zur Messung der User Experience von Softwareprodukten. In Mensch & Computer 2006: Mensch und Computer im StrukturWandel? . München: Oldenbourg Verlag, S. 125-134.

Norman, K.L.; Friedman, Z.; Norman, K.; Stevenson, R. (2001): Navigational issues in the design of online self-administered questionnaires. Behavior and Information Technology, 20, S. 37-45

Richter, M. (1999): Online-Befragung als neues Instrument zur Beurteilung der Benutzerfreundlichkeit interaktiver Software am Beispiel einer Internet-Anwendung. In: U.-D. Reips; B.Batinic; W. Bandilla; M. Bosnjak; L. Gräf; K- Moser; A.Werner (Hrsg.). Current internet science ? Trends, techniques, results. Zürich: Online Press.
http://www.intuitive.ch/liz_98.pdf

5.5 Think aloud

Lewis, C. H. (1982). Using the "Thinking Aloud" Method In Cognitive Interface Design (Technical report RC-9265). IBM.

http://grouplab.cpsc.ucalgary.ca/saul/hci_topics/tcsd-book/chap-1_v-1.html Task-Centered User Interface Design: A Practical Introduction, by Clayton Lewis and John Rieman.

Ericsson, K., and Simon, H. (May 1980). "Verbal reports as data". Psychological Review 87 (3): 215–251. doi:10.1037/0033-295X.87.3.215.

Ericsson, K., and Simon, H. (1987). "Verbal reports on thinking". In C. Faerch and G. Kasper (eds.). Introspection in Second Language Research. Clevedon, Avon: Multilingual Matters. pp. 24–54.

Ericsson, K., and Simon, H. (1993). Protocol Analysis: Verbal Reports as Data (2nd ed.). Boston: MIT Press. ISBN 0-262-05029-3.

Kuusela, H., and Paul, P. (2000). "A comparison of concurrent and retrospective verbal protocol analysis". American Journal of Psychology (University of Illinois Press) 113 (3): 387–404. doi:10.2307/1423365. JSTOR 1423365. PMID 10997234.

5.6 General

Nielsen, J. (1994): Usability Inspection Methods. New York, NY: John Wiley and Sons.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T. (1994). Human-Computer Interaction. Reading MA: Addison-Wesley.

Macaulay, L.A. (1996). Requirements Engineering. Berlin: Springer Verlag Series on Applied Computing.

Courage, C. and Baxter, K. (2005): Understanding Your Users: A Practical Guide to User Requirements Methods Tools and Techniques (Morgan Kaufmann Series in Interactive Technologies). San Francisco: Morgan Kaufmann.

Sarodnick, F. and Brau, H. (2006), Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung. Bern: Huber.

Combining and Uniting Business Intelligence with Semantic Technologies

Acronym: CUBIST

Project No: 257403

Small or Medium-scale Focused Research Project FP7-ICT-2009-5

Duration: 2010/10/01-2013/09/30

Appendix to Directives for the Evaluation of the Use Case Prototypes

Abstract: This deliverable will provide guidelines and directives for the evaluations of the use case prototypes (see use case prototypes D7.3.2, D8.3.2 and D9.3.2, and use case specific evaluations D7.4.1, D8.4.1 and D9.4.1)

Type	Report
Document ID:	CUBIST D1.4.1
Work package:	WP1
Leading partner:	SAP
Author(s):	Frithjof Dau (SAP)
Dissemination level:	PU
Status:	Final
Date:	18. Sept. 2013
Version:	1.0

Versioning and contribution history

Version	Description	Contributors
0.1	First draft	Frithjof Dau (SAP)
0.2	Extended draft	Frithjof Dau (SAP)
0.3	Incorporated questions from K. Stefani, structuring doc	Frithjof Dau, Karolin Stefani (SAP)
0.4	feedback from K. Stefani	Karolin Stefani (SAP)
0.5	Incorporated feedback, new chapter “preparation document”	Frithjof Dau (SAP)
0.6	Reworked after first test run	Katja Pfeifer (SAP)
1.0	Final document	Frithjof Dau (SAP)

1 OVERVIEW 4

2 PREPARATION DOCUMENT FOR THE TEST USERS 5

2.1 SOME BACKGROUND 5

2.2 HOW THE EVALUATION IS CONDUCTED 5

2.3 WHAT YOU HAVE TO DO BEFOREHAND 6

2.4 WHAT YOU HAVE TO DO DURING THE EVALUATION 6

2.5 SOME TERMS 7

2.6 INTRODUCTION INTO CONCEPTUAL SCALING 9

3 INTERVIEW 13

3.1 FOR THE TASKS AS CONDUCTED: 13

3.2 FOR MORE TASKS: 14

3.3 COMPARING THE DIFFERENT CUBIST MEANS TO ACCESS INFORMATION 15

3.4 DISLIKED/UNNEEDED/MISSING FEATURES 16

3.5 FAMOUS LAST WORDS 17

4 QUESTIONNAIRE 18

4.1 FOR THE OVERALL PROTOTYPE 19

4.2 FOR THE "SEARCH AND SELECT" COMPONENT 20

4.3 FOR THE „NAVIGATE IN DATA"-COMPONENT 21

4.4 FOR THE "EXPLORE SELECTION" COMPONENT 22

4.5 FOR THE "ANALYSE SELECTION"-COMPONENT 23

1 Overview

This document comprises several “subdocuments” which are needed for conducting the evaluation in CUBIST.

- In chapter 2 “Preparation Document for the Test Users”, we provide the information which is needed for the users *before* (at least one week) they conduct the evaluation. This chapter will be send to the test users beforehand as separate document.
- Chapter 3 ”Interview” contains the questions to be posed during the (structured) interview. This chapter will used as separate document by the evaluating persons and not be send to the test users.
- Chapter 4 “Questionnaire” contains the questionnaire. It will be send as separate document to the test users *during* the evaluation.

2 Preparation Document for the Test Users

Dear CUBIST test user,

Welcome to the CUBIST evaluation! In this short handout, we want to provide some core information for you needed to successfully take part in the evaluation.

2.1 Some Background

As you probably have been told or as you already know, CUBIST is a research-project, funded by the European Commission, and conducted by seven partners (and you probably belong to one of them) coming from different European Countries (Germany, France, UK, Belgium, and Bulgaria). In CUBIST, we essentially investigated and implemented (in a prototype) novel ways to access information, e.g. searching for specific information, exploring the information space, or visually analyzing information with interactive, traditional and novel, diagrams. CUBIST lasted three years and will be finished by the end of September 2013, i.e. we are in the very final phase of this project. Unsurprisingly, there is an evaluation conducted in this final phase, and you are taking part in it. So, first things first:

Many thanks for your participation!

For the evaluation it is important to note that the CUBIST system you will test is a **research prototype**. That is, when we have developed the prototype, the focus on this prototype was on providing **new approaches** for searching (for), exploring and analyzing information, and not on developing a final piece of software being feature-complete, robust and reliable, and having sufficient performance. We, the CUBIST consortium and developers, are quite aware of the fact that this prototype sometimes lacks performance, that the prototype does not provide all features which one could envision, and that the prototype sometimes breaks. The evaluation does not aim at finding those gaps in order to close them. Instead, we want to investigate whether the **notions and ideas** to access information as they have been implemented in CUBIST work out for you, the user. In fact, you will find no questions during the evaluation which target e.g. the performance of the system; instead, you will be confronted with questions like “is this and that visualization **useful** for you”. Having said this we ask you to be both honest in your answers, but not too much distracted by performance- or robustness-issues. We, so-to-speak ask you to evaluate not the existing prototype as it stands, but an “ideal prototype” without performance- or stability-issues, but with the same functionalities.

2.2 How the evaluation is conducted

From a bird’s eye perspective, the following is about to happen:

1) Before the actual evaluation
   Before the evaluation, we provide you some material about CUBIST (including this information sheet). With this material, we expect that you familiarize yourself with the prototype and that you provide three tasks (tasks of your work) you want to carry out with CUBIST and show us.
2) During the evaluation

The actual evaluation will consists of three parts:

1) You will conduct the three tasks which have been defined by you beforehand, and we will watch and listen.
2) After this, we conduct a short interview with you.
3) Finally, we ask you to fill out a short questionnaire.

2.3 What you have to do beforehand

2.3.1 Watch tutorial video

First of all, we have created a quite comprehensive tutorial video which shows you – apart from a functionality called "conceptual scaling"- the functionalities of CUBIST. You can find the video here: http://www.youtube.com/watch?v=Kuu756nr1_I . Please watch the video.

2.3.2 Conceptual Scaling

If you belong to Innovantage or Space-Application-Services, there is a specific functionality called "conceptual scaling", which is not addressed by the tutorial video, but which is needed before you enter the "visual analytics" of CUBIST. After having watched the video, please read the introduction into conceptual scaling at the end of this document.

2.3.3 Provide Tasks

We ask you to list three tasks/information needs which you consider to be easy (or relatively easy) accomplishable with CUBIST, but which are hard or impossible done with your current tools. These tasks should resemble tasks as taken down during the requirement analysis in the utilization scenario (for more details, please ask your respective contact person per use case partner, e.g. Ken M^{c}Leod for HWU or Emre Sevinç for SAS). For those tasks, keep alternative flows like unexpected events (e.g. information not found, dependencies with other resources/people) in mind. In the .execution phase, those tasks will be conducted with the CUBIST prototype.

2.4 What you have to do during the evaluation

The evaluation as such will be conducted remotely. We will use a web conferencing solution hosted by SAP, which addresses a full range of online meeting needs, from simple screen sharing to mission-critical, real-time collaboration like chats or whiteboards. We will record all evaluation sessions.

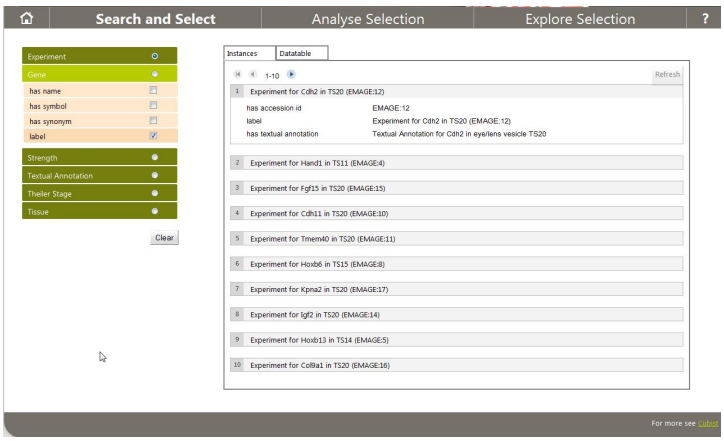
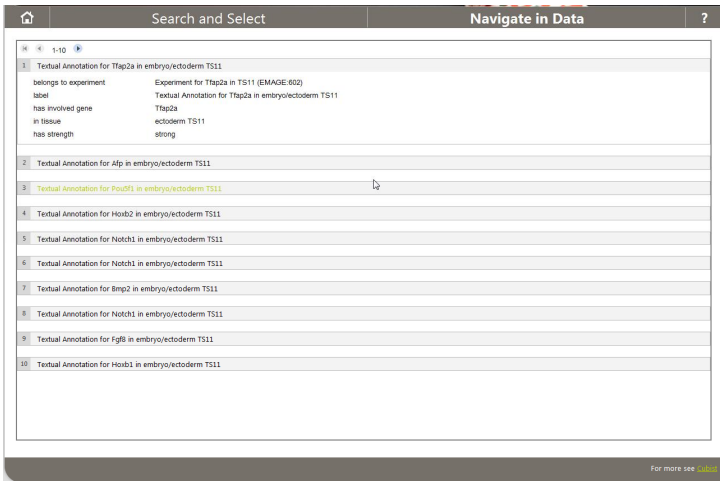
1) In the first part of the evaluation, we ask you to conduct live with the CUBIST system the tasks you have defined beforehand. We are interested in the *process* of the task being conducted, not simply the final result. Thus we want you to think aloud as you are performing a task, and we will ask you to say whatever you are looking at, thinking, doing, and feeling as you go about your task. And, btw, we will objectively take notes of everything that you say, without attempting to interpret your actions and words.
2) In the second part of the evaluation, we will conduct an interview with you. This interview will cover questions about the tasks you just conducted, questions about prospective other

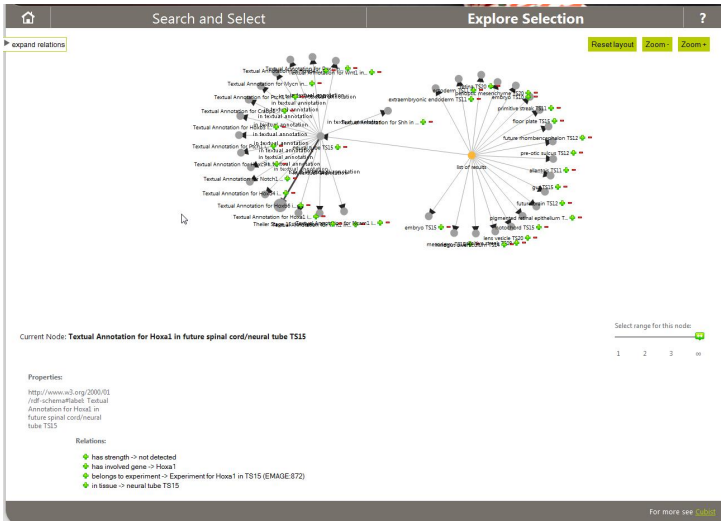
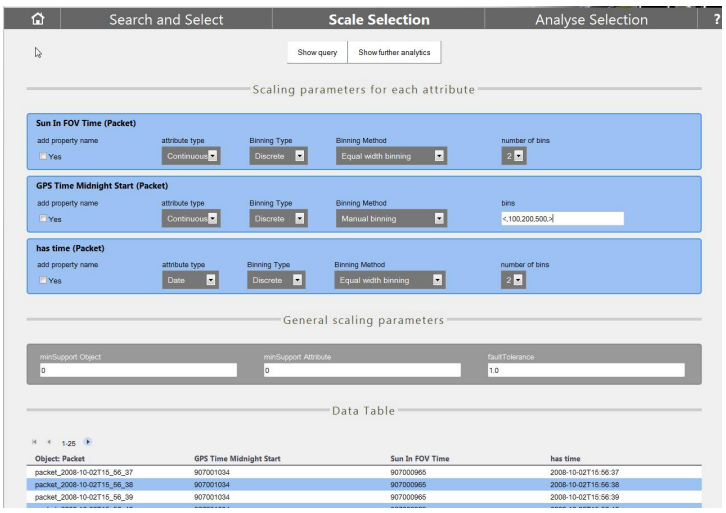
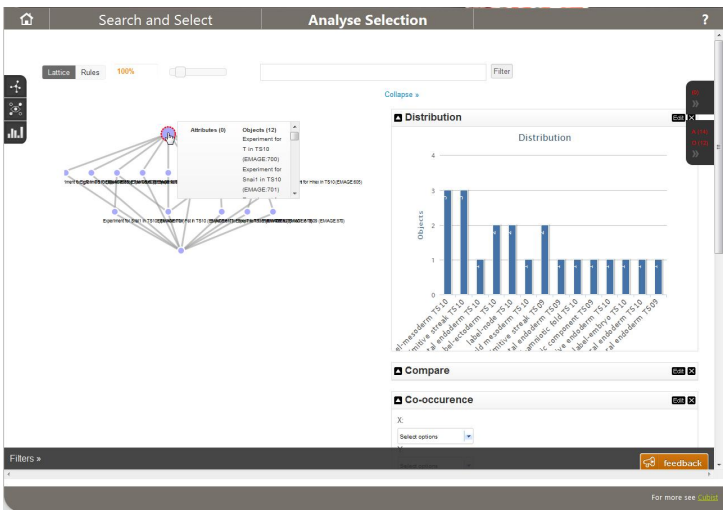
tasks, questions about the different "components" of the use case, and some questions about features you dislike or you miss.

3) Finally, in order to get some quantifiable results, we will send you a questionnaire (a word document) and we ask you to fill out his questionnaire and send it back to us.

2.5 Some Terms

In the overall CUBIST prototype, we have different components or panels which provide different means to access information. In this section, we aim at clarifying the terms "search and select"-panel (or "search and select"-component), "navigate in data"-panel etc by providing screenshots (from the HWU or SAS use case) from the respective panels.

"Search and Select". This is the first panel when you work with CUBIST. It allows you to search for data, and it allows you to define selections of data, which are used by other components.	
"Navigate in Data" This is a text-based panel, which allows you to navigate along the semantic relationships through the data.	

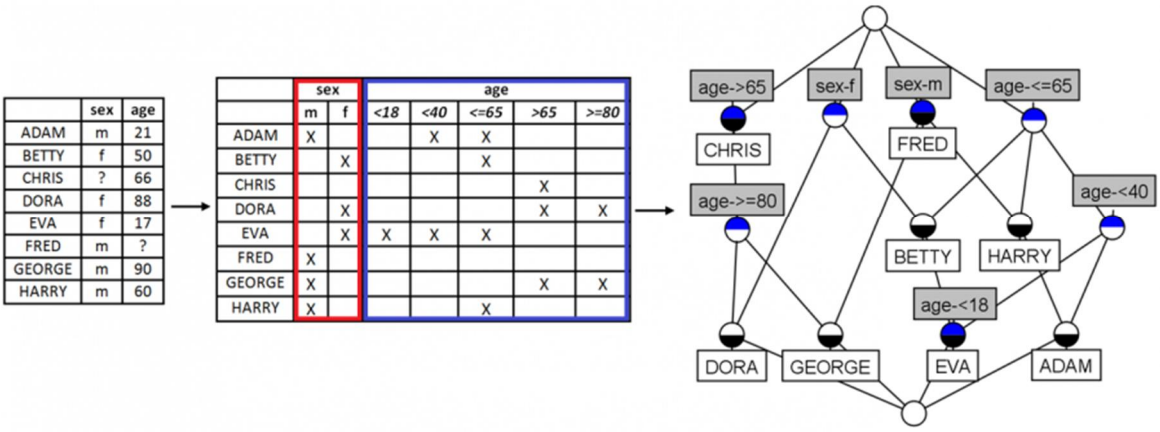
"Explore Selection" In this panel, data can be visually explored using a graph (a "node-edge-diagram").	
"Scale Selection" This panel allows you to replace attributes that are too fine-grained (e.g. replacing time attributes with the granularity of seconds by attributes with the granularity of minutes) by more coarse-grained attributes. This panel will not be covered by the evaluation.	
"Analyse Selection" This panel is the panel that allows you to visually and interactively analyse pre-selected entities.	

2.6 Introduction into Conceptual Scaling

2.6.1 Introduction

In CUBIST, we deal with numerous attributes, having types like “date”, “number” and “string”. Sometimes those attributes are too fine-grained in order to used directly in the “Analyse Selection” - Panel of CUBIST. Instead, one wants to replace the “raw”-attributes by more general, coarse-grained attribute. This process is called “conceptual scaling”. Scaling is done per (selected) attribute. Below is an example of scaling the sex and age of eight persons. The first table contains two attributes “sex” and “age”. The first attribute can be essentially taken as it is, whereas one might not be interested in the precise ages of all persons, but instead, one might consider only *ranges* of ages. To be more precisely:

- The first scale (second table, highlighted in red) contains the two possible values of sex, ‘m’ for male and ‘f’ for female.
- The second scale (second table, highlighted in blue), age, has potentially many values as it is a continuous attribute, so rather than using individual values, a scale was created using boundary values: ‘<18’, ‘<40’, ‘≤65’, ‘>65’ and ‘≥80’. ‘ADAM’ is 21 years old, so has the formal attributes ‘<40’ and ‘≤65’. ‘GEORGE’ is 90 years old, so he has the formal attributes ‘>65’ and ‘≥80’.
- The two scales are then merged to form the complete formal context representing ‘the sex and age of eight persons’ (second table).
- The resulting lattice can be seen on the right.



	sex	age
ADAM	m	21
BETTY	f	50
CHRIS	?	66
DORA	f	88
EVA	f	17
FRED	m	?
GEORGE	m	90
HARRY	m	60

	sex		age				
	m	f	<18	<40	<=65	>65	>=80
ADAM	X			X	X		
BETTY		X			X		
CHRIS						X	
DORA		X				X	X
EVA		X	X	X	X		
FRED	X						
GEORGE	X					X	X
HARRY	X				X		

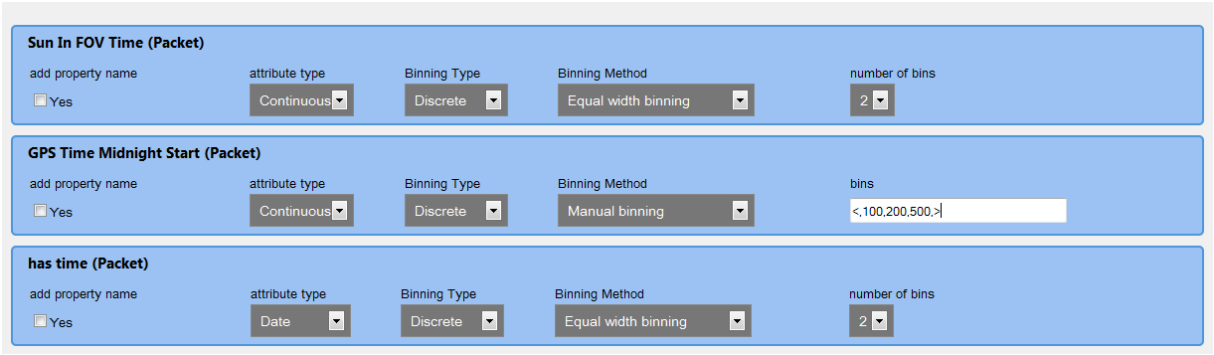
Some information that can be read from the lattice:

- The oldest persons are Dora and George, being the only ones having the *age-≥80* attribute.
- The youngest person is Eva, being the only one having the *age-<18* attribute. Interesting fact with Eva is to notice the hierarchy in the lattice; as she is younger than 18, she is also younger than 40 and younger than 65.

- Chris is the only one whose gender is unspecified. Take a closer look at the original data (table 1). His gender is marked as '?' implying unknown. Since it's unknown, in table 2 he has neither *age-m* or *age-f*. If we would like to include this in the analysis, we could have added an extra value for the age scale in table 2: *age-?*.

2.6.2 Conceptual Scaling in CUBIST

Below, we repeat a part of the ”Scale Selection”-Panel screenshot provided in the last section. We see that for each attribute, we can essentially select an attribute type, a binning type, and a binning method. These three parameters are explained below.



2.6.2.1 **Attribute Types**

In CUBIST, we deal with numerous attribute types including dates, numbers and strings. Each attribute type requires it’s own approach to convert and analyze it. The attribute types encountered in CUBIST, along with a brief description of how each type is scaled, are mentioned below.

- **Categorical**
  - The predominant attribute type is categorical. This is the typical many-valued attribute, e.g. ‘Color’ can have multiple values such as red, green, blue, black, yellow, etc.
  - Categorical attributes are converted by creating one formal attribute for each of the attribute categories, e.g. Color-red, Color-green, Color-blue, etc.
  - In the conversion process, each value read in from the data is compared with the category values listed for the corresponding attribute. A match is recorded as a true value in the formal context.
- **Boolean**
  - A Boolean attribute can be interpreted as a single formal attribute. Typically, a Boolean attribute in a dataset is one that has two categories that represent *true* and *false*.
  - During the conversion process, the corresponding value read in from the data file is compared with the true value. If they match, this is recorded as a true value in the formal context.
  - Treating a Boolean attribute as categorical can also allow for both the true and false values to appear in the formal context.

- **Continuous**
  - Continuous attributes are dealt with by discretising the data using user-defined ranges (e.g., *0–10, 10–20*...), or by hierarchical scaling (e.g., *> 0, > 10, >= 20*...).
  - During the conversion process, a continuous value read in from the data file is compared with the appropriate boundary values, assigning a true value to the corresponding formal attribute in the formal context.
- **Ordinal**
  - Ordinal attributes are categorical attributes where the ordering of the attribute values is important.
  - Taking as an example an attribute Education containing higher education degrees, it is important that a foundation course comes before a Bachelors degree, a Doctorate comes after a Masters degree and so on.
  - Ordinality also allows the creation of sensible ranges to summarise values for non-continuous attributes. Using the same example, creating ranges for the Education attribute can be used to group education levels to undergraduate and postgraduate: *Foundation-Bachelors*, *Masters-Doctorate*.
  - Capturing the order of non-numerical data is a feature that allows analyses that are not easy using traditional means of data analysis.
  - When an ordinal attribute has been declared, a numerical value is assigned to each of the attribute values. During the conversion process, an attribute value is read, it's numerical counterpart is found and compared with the appropriate boundary values, assigning a true value to the corresponding formal attribute in the formal context.
- **Date**
  - The most common date formats are supported. As an example, creating ranges for a dates attribute can be used to group months into seasons: 01/12/2010 to 28/02/2011, 01/03/2011 to 31/05/2011, 01/06/2011 to 31/08/2011 and 01/09/2011 to 30/11/2011.
  - Declaring ranges for date attributes is possible using the same process used for declaring ranges for continuous attributes. Adding hours, minutes and seconds to a date is also possible – 23/09/2011 01:55:40 AM is valid as a date value.
  - During the conversion process, a date value is read and compared with the appropriate boundary values, assigning a True value to the corresponding formal attribute in the formal context.

The categorical and Boolean attributes are straightforward as to how they are scaled: for a categorical attribute, each of its values is used to create the formal attributes, while for Boolean only the true (or even false) value is used to create the formal attribute (in effect, declaring a Boolean attribute as Categorical is valid: both True and False values will appear in the formal context). Continuous/Dates attributes need some explanation, see below.

2.6.2.2 Binning Types

Let's how the two continuous/date scaling options work with an example: Let's assume that we have ***manually*** defined the **boundaries** of the ranges of an ***age*** attribute to be *"0,20,35,50"*. Discrete scaling will result in this:

	age-0to<20	age-20to<35	age-35to<50
Person 1 (25 yrs)		X	
Person 2 (43 yrs)			X
Person 3 (17 yrs)	X		
Person 4 (75 yrs)			

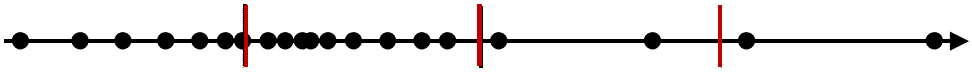
…and Progressive scaling will result in this:

	age-0to<20	age-0to<35	age-0to<50
Person 1 (25 yrs)		X	X
Person 2 (43 yrs)			X
Person 3 (17 yrs)	X	X	X
Person 4 (75 yrs)			

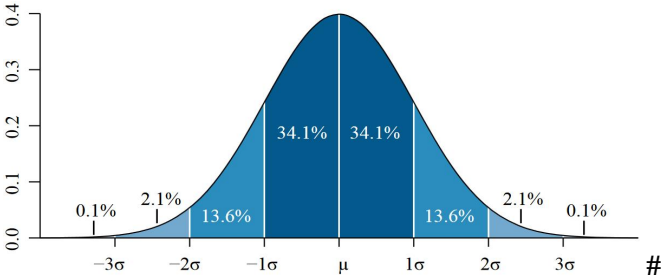
2.6.2.3 **Binning Methods**

As demonstrated in the previous examples, continuous (and date) attributes are dealt with by producing discretized values of a continuous attribute and replacing it with the new values. This approach makes continuous attributes easier to handle. The ranges created when discretizing a continuous attribute are also called *bins*. In the data binning process, the original data values which fall in a given small interval (the so-called *bin*) are replaced by a value representative of that interval. I have implemented three binning methods:

- **Equal width binning**: the attribute is scaled using ranges of equal width



- **"Progressive" standard deviation binning:** let's say we have a continuous attribute with a mean (denoted by μ) value of 50 and a standard deviation (denoted by σ) of 10. We abbreviate Standard Deviation with the acronym StD (not to be confused with *Sexually Transmitted Diseases* [STDs], which are actually renamed to *Sexually Transmitted Infections* [STIs] now ☺)



- **Manual binning:** the borders of the ranges (bins) are entered manually. Moreover, one can enter two special borders "<" and ">", meaning "all values below the smallest border" and "all values above the largest border".

#

3 Interview

- Name: Click here to enter text.
- Use Case (HWU; SAS; INN): Click here to enter text.
- Age: Click here to enter text.
- Gender: Click here to enter text.
- Profession: Click here to enter text.
- Computer Usage per day in hour: Click here to enter text.
- Date of Test: DD/MM/2013 Click here to enter text.
- Location of Test: [City, State] Click here to enter text.
- Please rate your overall computer skills?

  ☐ Very good (e.g. programming, security, data modeling, …).
  ☐ Good (e.g. frequently using spreadsheet applications, advanced in office tools, analysis tools,…)
  ☐ Standard (e.g. surfing, e-mail and writing simple documents, but not much more).
  ☐ Bad.

3.1 For the tasks as conducted:

1. Please shortly describe the tasks you conducted with CUBIST:

   Click here to enter text.

   What do you expect from a system to fulfill these tasks?

   Click here to enter text.

2. Did the system offer you the right information to fulfill your analytical tasks?
   a. If yes, what kind of information and system functionality provided did you find especially helpful?
      Click here to enter text.
   b. If no, what was missing from your point of view (regarding data provided/visualization possibilities/ interaction possibilities) to derive the desired information?
      Click here to enter text.

3. Did you discover new facts during your analysis tasks that you had not expected to discover at all before?
   a. If yes, what kind of new facts did you discover that were most surprising for you?
      Click here to enter text.

b. No.

4. If the tasks fulfilled are typical for your daily work, do you think the tool can enrich your daily work by offering new ways to analyze your data?
   a. Yes, because
      Click here to enter text.
   b. No, because
      Click here to enter text.
   c. Partly, because
      Click here to enter text.

3.2 For more tasks:

These questions can be omitted, if the interview takes too long!

5. Which analytical systems do you currently use in your daily work?
   a. The tools are for example:
      Click here to enter text.
   b. None.

6. From your point of view …
   c. Please shortly describe what is missing in current systems to use them effectively for your daily tasks:
      Click here to enter text.
   d. Do you think CUBIST fills an analytical gap or provides functionalities that better fit your analytical tasks? Why do you think so?
      Click here to enter text.

7. Next to the data/use case currently implemented in the system, do you see any content from your daily life (private and professional) to be integrated in the system in future?
   Click here to enter text.
   Why do you think this would be benefit?
   Click here to enter text.

8. For which kind of tasks from you daily work do you believe the system can be especially useful? Please describe the tasks and the possible benefit shortly:
   Click here to enter text.

9. For which kind of tasks from your daily work do you believe the system is annoying / ineffective? Please describe the tasks and the possible drawbacks shortly:
   Click here to enter text.

3.3 Comparing the different CUBIST means to access information

Which of CUBIST's analysis components did you find most valuable for your tasks and why?

- „Search and Select"-Panel
  Click here to enter text.
- Instance View
  Click here to enter text.
- „Explore Selection"-Panel
  Click here to enter text.

a. „Scale Selection"-Panel
   Click here to enter text.

- „Analyse Selection"-Panel
  Click here to enter text.
- Traditional Visualization (e.g. bar charts):
  Click here to enter text.
- Other – Function:
  Click here to enter text.

10. For each of the following components, what do you think for what kind of information need it is suited best (e.g. number analysis, …)?
    a. „Search and Select"-Panel
       Click here to enter text.
    b. „Navigate in Data"-Panel
       Click here to enter text.
    c. „Explore Selection"-Panel
       Click here to enter text.
    d. „Scale Selection"-Panel
       Click here to enter text.
    e. „Analyse Selection"-Panel
       Click here to enter text.
    f. Traditional Visualization (e.g. bar charts):
       Click here to enter text.
    g. Other – Function:
       Click here to enter text.

11. How easy was it for you to choose the most appropriate analysis approach and visualization for your needs from the overall functionalities offered:
    - Very easy
    - Easy

- Difficult
- Very difficult

Because, Click here to enter text.

12. What (functionalities and/or system in general) was the major drawback of CUBIST for you and why?
    Click here to enter text.

13. Do you think you understood how all the different means offered by CUBIST to access information interact? Comment if necessary.
    - Yes, Click here to enter text.
    - No, Click here to enter text.
14. How did you like the guidance offered by the system to navigate through the available information? Please comment your decision.
    - Very good guidance
    - Good guidance.
    - Ok. Not to mention
    - Poor guidance
    - Very poor guidance.

    Because, Click here to enter text.

15. Did you immediately understand how to read the visualizations and use the analysis functionalities in the tool?
    - Yes: If yes, can you describe what made it easy for you to understand the tool?
      Click here to enter text.
    - No: If no, what made it difficult?
      Click here to enter text.

3.4 Disliked / Unneeded / Missing Features

In this part, we aim at finding out disliked, unneeded or missing features. We do this per component.

16. For each of the following components, which features do you dislike most? And Why?
    a. „Search and Select“-Panel
       Click here to enter text.
    b. „Navigate in Data“-Panel
       Click here to enter text.
    c. „Explore Selection“-Panel

Click here to enter text.

d. „Scale Selection“-Panel
   Click here to enter text.
e. „Analyse Selection“-Panel
   Click here to enter text.

17. For each of the following components, which features are not needed from your point of view?
    a. „Search and Select“-Panel
       Click here to enter text.
    b. „Navigate in Data“-Panel
       Click here to enter text.
    c. „Explore Selection“-Panel
       Click here to enter text.
    d. „Scale Selection“-Panel
       Click here to enter text.
    e. „Analyse Selection“-Panel
       Click here to enter text.

18. And finally, each of the following components, which features missing?
    a. „Search and Select“-Panel
       Click here to enter text.
    b. „Navigate in Data“-Panel
       Click here to enter text.
    c. „Explore Selection“-Panel
       Click here to enter text.
    d. „Scale Selection“-Panel
       Click here to enter text.
    e. „Analyse Selection“-Panel
       Click here to enter text.

3.5 Famous last words

Do you have any more comments or remarks?

Click here to enter text.

4 Questionnaire

- Name: Click here to enter text.
- Use Case (HWU; SAS; INN): Click here to enter text.

4.1 For the overall prototype

	strongly agree			neutral		strongly disagree		n/a
The CUBIST software was easy to use and work with.	☐	☐	☐	☐	☐	☐		☐
Neglecting the currently prototypic character, I would like to use the CUBIST software in future again.	☐	☐	☐	☐	☐	☐		☐
In future, I would prefer CUBIST to other analytical tools I currently use.	☐	☐	☐	☐	☐	☐		☐
Using CUBIST software could make my work more effective and efficient.	☐	☐	☐	☐	☐	☐		☐
The integration of different components (used to access, explore and visualize information) was helpful for fulfilling my tasks.	☐	☐	☐	☐	☐	☐		☐
The different components and the visualizations in CUBIST are well integrated.	☐	☐	☐	☐	☐	☐		☐
It is clear how the different components interact.	☐	☐	☐	☐	☐	☐		☐
The navigation/interaction functionalities were easy to understand and apply.	☐	☐	☐	☐	☐	☐		☐
It was easy to follow the steps performed by the system when using the interaction functionalities.	☐	☐	☐	☐	☐	☐		☐

4.2 For the "Search and Select" Component

The purpose and function of the component is clear.	strongly agree				neutral				strongly disagree			n/a	
	☐	☐	☐	☐	☐	☐						☐	
The component is easy to understand and use.	strongly agree				neutral				strongly disagree			n/a	
	☐	☐	☐	☐	☐	☐						☐	
The interface is appealing and attractive.	strongly agree				neutral				strongly disagree			n/a	
	☐	☐	☐	☐	☐	☐						☐	
The component is useful.	strongly agree				neutral				strongly disagree			n/a	
	☐	☐	☐	☐	☐	☐						☐	
For some kinds of information needs or queries, particularly this component (or similar components based on the same approach) is useful.	strongly agree				neutral				strongly disagree			n/a	
	☐	☐	☐	☐	☐	☐						☐	
I have similar functionalities in the tools I usually use.	strongly agree				neutral				strongly disagree			n/a	
	☐	☐	☐	☐	☐	☐						☐	

4.3 For the „Navigate in Data”-Component

The purpose and function of the component is clear.	strongly agree			neutral			strongly disagree			n/a
	☐	☐		☐	☐		☐	☐		☐
The component is easy to understand and use.	strongly agree			neutral			strongly disagree			n/a
	☐	☐		☐	☐		☐	☐		☐
The interface is appealing and attractive.	strongly agree			neutral			strongly disagree			n/a
	☐	☐		☐	☐		☐	☐		☐
The component is useful.	strongly agree			neutral			strongly disagree			n/a
	☐	☐		☐	☐		☐	☐		☐
For some kinds of information needs or queries, particularly this component (or similar components based on the same approach) is useful.	strongly agree			neutral			strongly disagree			n/a
	☐	☐		☐	☐		☐	☐		☐
I have similar functionalities in the tools I usually use.	strongly agree			neutral			strongly disagree			n/a
	☐	☐		☐	☐		☐	☐		☐

4.4 For the "Explore Selection" Component

The purpose and function of the component is clear.	strongly agree			neutral			strongly disagree			n/a		
	☐	☐		☐	☐		☐	☐		☐		
The component is easy to understand and use.	strongly agree			neutral			strongly disagree			n/a		
	☐	☐		☐	☐		☐	☐		☐		
The interface is appealing and attractive.	strongly agree			neutral			strongly disagree			n/a		
	☐	☐		☐	☐		☐	☐		☐		
The component is useful.	strongly agree			neutral			strongly disagree			n/a		
	☐	☐		☐	☐		☐	☐		☐		
For some kinds of information needs or queries, particularly this component (or similar components based on the same approach) is useful.	strongly agree			neutral			strongly disagree			n/a		
	☐	☐		☐	☐		☐	☐		☐		
I have similar functionalities in the tools I usually use.	strongly agree			neutral			strongly disagree			n/a		
	☐	☐		☐	☐		☐	☐		☐		

4.5 For the "Analyse Selection"-Component

	strongly agree		neutral		strongly disagree		n/a
The purpose and function of the component is clear.	☐	☐	☐	☐	☐	☐	☐
The component is easy to understand and use.	☐	☐	☐	☐	☐	☐	☐
The interface is appealing and attractive.	☐	☐	☐	☐	☐	☐	☐
The component is useful.	☐	☐	☐	☐	☐	☐	☐
For some kinds of information needs or queries, particularly this component (or similar components based on the same approach) is useful.	☐	☐	☐	☐	☐	☐	☐
I have similar functionalities in the tools I usually use.	☐	☐	☐	☐	☐	☐	☐
The visualizations were easy to understand.	☐	☐	☐	☐	☐	☐	☐
There are visualizations available that did fit my tasks very well.	☐	☐	☐	☐	☐	☐	☐
The integration of different visualizations was helpful for fulfilling my task.	☐	☐	☐	☐	☐	☐	☐
It is clear how the different visualization interact.	☐	☐	☐	☐	☐	☐	☐