Degree project

Design and Development of a Decision Making Application Using Analytic Hierarchy Process

Author: Igor Kovbasiuk
Supervisor: Welf Löwe
External Supervisor: Mathias Hedenborg, Linnaeus University

Date: 2014-09-24

Course Code: 5DV00E, 30 credits
Level: Master

Department of Computer Science
Abstract

Every action a person takes is determined by the result of some decision. Making simple decisions is natural and does not require additional considerations. However, in case of multiple alternatives and criteria to be considered, decision-making technique is required. The most studied and developed technique is the Analytic Hierarchy Process (AHP).

This thesis is focused on the practical implementation of AHP. Firstly, it defines a set of features that are necessary for the decision-making process involving several experts and additional non-functional requirements to be met. Feature comparison has shown that none of the existing applications implements all required features. Therefore, a new application is designed. Further on, the engineering process is described, including transforming functional requirements into features, features into use cases, use cases into activities diagrams. After the developed application is described screens corresponding to each use case are presented.

Non-functional requirements are portability, free availability, and usability. Compliance of the developed application with these criteria is checked with tests and two user experiments.

The main results of this thesis are: (i) the extension of AHP theory with external consistency check to improve the quality of final results and (ii) the developed application which meets all requirements.

Keywords: decision making, analytic hierarchy process.
Contents

1 Introduction ............................................. 1
    1.1 Background ........................................ 1
    1.2 Previous research .................................. 2
    1.3 Problem definition .................................. 2
    1.4 Scope ............................................. 3
    1.5 Scientific approach ................................ 4
    1.6 Ethical considerations .............................. 4
    1.7 Report layout ...................................... 5

2 Theory and background ................................. 6
    2.1 AHP theory description ............................ 6
    2.2 Notions and definitions ............................ 7

3 Problem definition refinement ....................... 9
    3.1 Refining goal ...................................... 9
    3.2 Refining goal criteria ............................. 9
    3.3 Evaluating competitors against refined goal criteria .. 10
    3.4 Conclusion ....................................... 11

4 Architecture and design ................................. 12
    4.1 Use cases ......................................... 12
    4.2 Architecture ...................................... 13
    4.3 Design description ................................ 15

5 Implementation details ................................. 24
    5.1 Use cases implementation .......................... 24
    5.2 Usability measures implementation ................. 30

6 Evaluation .............................................. 33
    6.1 Portability goal criterion .......................... 33
    6.2 Free available goal criterion ...................... 36
    6.3 Usability goal criterion ........................... 36
        6.3.1 Experiment setup description ................ 36
        6.3.2 Execution of experiment ..................... 37
        6.3.3 Discussion of results ....................... 38
    6.4 Conclusion ....................................... 39

7 Conclusion, future work ............................... 40
    7.1 Conclusion ....................................... 40
    7.2 Future work ....................................... 40

Reference literature ...................................... 42

A Appendix .............................................. 1
    A.1 AHP introduction .................................. 1
    A.2 Decision criteria tree ............................ 2
    A.3 Task ............................................. 3
1 Introduction
In this chapter the background of the thesis is given and the previous research in decision making with the AHP is summarized. Thesis goal and goal criteria are defined and it is determined how they can be refined and checked. After that, thesis scope is outlined to limit research field and describe scientific approach used in the thesis for problem solving. In the ethical considerations section risks involved with the usage of the application are described.

1.1 Background
People are all decision makers. Everything we do is the result of some decision. When decision process involves a lot of alternatives, or when a goal is complex and consists of many criteria, it is very difficult to make a decision. For example, when a student considers which occupation to pursue – PhD or to work in the industry, several criteria, such as income, fun, experience, should be considered. It is impossible to make a decision while keeping all these criteria in mind. However, it is possible to compare pairs of alternatives with respect to each criterion. For example, PhD vs. work with respect to income, PhD vs. work with respect to fun. Now special decision-making technique is required to combine these pairwise judgments together and calculate priorities of criteria and alternatives.

Such technique formalizes thinking process thus allowing for transparent decisions to be made. Furthermore, it allows not only to find out the best decision, but to form a ranking of alternatives [1]. Such ranking can be useful in case of limited resources to be allocated, allowing efficient distribution of available resources.

Cooperation of experts is also an important issue. People tend to have different opinions, and it is difficult to find a solution that satisfies all experts. Hence, transparent incorporating of different experts opinions is also required.

The AHP is an effective decision-making approach applied in dealing with this kind of decision problems. According to this technique, decision is decomposed into a hierarchy of easily comprehended sub-problems, each of them can be analyzed independently. Thus, complex decision-making problem is simplified to many small comparison tasks. The AHP is flexible, straightforward and provides a rational and consistent way for decision making [2]. It was selected because it provides a simple interface for both decision owner and experts to understand decision making and participate in it. Even users without AHP knowledge can use AHP based tools. Description of this method and its application in the project is done in chapter 2.

Thesis is focused on the practical application of the AHP. The AHP is extended in order to allow consolidation of experts' opinions. The implementation is focused on both PC and mobile platforms, so portability is an important property to be achieved. Compliance with this property will be checked with tests.

Furthermore, it is important to cover internal calculations and complexity of the AHP from users, allowing them to focus on the decision. Thus, usability property is valuable. Compliance of the program with this property will be checked with the help of the user experiment.
1.2 Previous research

Decision-making software is based on multi-criteria decision-making processes. They include Aggregated Indices Randomization Method (AIRM), Analytic Hierarchy Process (AHP), Analytic Network Process (ANP). However, only the AHP provides a transparent way to incorporate the opinions of experts. Moreover, the AHP is simple to use for people without any prior knowledge of the process. That is why the AHP is most dominant in modern decision-making software [15].

The AHP was developed by Thomas L. Saaty in the 1970s and since then has been extensively studied and refined [2]. Thus, no additional theoretical research is required. The only significant refinement in the AHP done in the thesis was introducing an external consistency check (cf. 2.2). However, it does not influence results calculation technique. It was added to improve the consistency of the consolidated opinion.

Known applications implementing the AHP, are either expensive (MakeItRational [16], TransparentChoice [17]) or complex and hard to master (Super Decisions [18]). The functionality of these tools is not sufficient. Furthermore, mobile devices are common now, but none of these applications is designed to work on a mobile device. Detailed comparison of the competitors is done in chapter 3.3.

To sum up, there exist applications implementing the AHP, but none has all requested features implemented. Thus, to solve the abovementioned problems of the decision-making process, a new application was developed in this thesis.

1.3 Problem definition

The main problem addressed in the thesis is consolidating opinions of multiple experts participating in decision making. Additional requirements are portability and usability of the solution. Existing tools do not satisfy these requirements (chapter 3.3).

The goal of the thesis is to find or design and develop portable, usable decision-making application using the AHP. There can be determined functional and non-functional requirements, which describe what we expect from the application. In chapter 3.3 the evaluation of competitors is performed, which shows that none of the existing tools satisfies specified requirements. The thesis is focused on design and development of such application.

This section is focused on a general description of functional requirements, according to decision making using the AHP. The detailed refinement is performed in chapter 3.1.

According to the AHP, at the beginning of the decision-making process, a decision is created, and its goal is specified. Then criteria (cf. 2.2) and alternatives (cf. 2.2) are added. Experts (cf. 2.2) participate in the decision by performing pairwise comparisons (cf. 2.2). These two activities: adding criteria or alternatives and performing pairwise comparisons should be possible in parallel. After all comparisons are performed, results - priorities of criteria, alternatives - are calculated.

Unit tests were developed to ensure correct implementation of the above mentioned functional requirements. Checking of this functionality working together (integration testing) is done via portability tests and usability user experiment.

This section is focused on a general description of the non-functional requirements. The detailed refinement is performed in chapter 3.2.
Mobile devices are common now, but none of the currently existing applications supporting the AHP is designed to work on mobile devices. At the same time, portability must not hamper usability.

The portability requirement also influenced development technologies. Web technologies were selected and they provide the following advantages:

- access to the application from any device connected to Internet, both PCs and mobile devices, thus web-based solution facilitates both usability (user can use any device) and portability;
- unlike native applications, web applications do not require installation of software to a computer/device. User can open browser to access the application and use it anytime he wants;
- reduce development cost, as we develop only one application that is accessed from various devices with different operating systems;

Thus, web-based application is the best way to solve the problem. The portability requirement influenced selected web technologies to use. To develop portable application the Google Web Toolkit (GWT) [6] will be used, which allows to provide versions for both PC and mobile users.

Compliance of the program with the portability characteristic will be checked via series of tests. All use cases must be available from both PC and mobile.

Working with complex decision involving large criteria tree and several alternatives can be difficult for a person not familiar with the AHP. Thus, a program should be user-friendly and simple to use for such users. They should clearly understand which comparisons are left to be performed by them. In the case of inconsistent judgments, the error message should be clear, so it would be obvious what to change in order to make judgments consistent. Furthermore, during results browsing, users should understand the final result and see their contribution to it.

At the same time, users familiar with the AHP should be able to create a decision, add/remove experts and develop decision trees.

Therefore, usability is vital for both new and experienced users.

Experiments will be conducted to check usability. Experiments will be also focused on features that are not provided in competitor tools. To improve usability of developed application implementing required features, usability property will be refined into measures, which will be implemented (cf. 3.2). User experiment will show whether those measures were successful, and whether the application is usable (cf. 6.3). During the experiment, such usability indicators will be analyzed: time to perform experiment task, number of questions asked and feedback from users.

1.4 Scope
Thesis is not focused on theoretical refinement of the AHP, but on tuning of the AHP and implementing it to make usable, mobile-accessible decision-making application.

Theoretical basis for the AHP has been extensively studied and developed since 1970th. Hence, there is no need for further theoretical development of the AHP. Only two extensions to classical AHP process were introduced in this thesis: consolidation of experts' opinions and external consistency check of experts' opinions.

According to the group decision-making definition [2], aggregation of individual judgments from a group into a single representative judgment for the entire group is
done by calculating geometric mean of these judgments. Moreover, the result of combining consistent judgments will always be consistent.

However, the consistency of the result does not imply semantic value of this result. In the case of opposite or very different opinions, the results will differ from all initial judgments. Though obtained result is mathematically correct, it is not useful for decision making. In order to avoid such situations, an external consistency check was introduced (chapter 2). It ensures that experts first agree on general terms with their judgments and then their opinions are taken for consolidated result calculation.

The thesis is mainly focused on practical implementation of the AHP using web technologies and assessing the usability and the portability of the developed application.

1.5 Scientific approach
The scientific approach of this thesis uses a combination of design and natural science. The design science [3] approach to thesis’ problem includes:

1) problem definition: specification and refinement of the goal;
2) finding existing solutions: feature comparison of existing tools, which shows none of the existing tools satisfies the goal;
3) designing and implementing an application satisfying the goal. The completeness of the implementation is assured by mapping of feature definitions to use cases, use cases to activity diagrams and, finally, to code;
4) checking whether the application satisfies the goal and the goal criteria.

Since the goal is refined into criteria, the checking of compliance with the goal is also separated for each goal criterion. For portability we perform tests, the criterion of free availability is just observed.

Experimental hypothesis testing (controlled experiment) [9] is used to check compliance with usability. This part is an approach from natural sciences. The user experiment is performed in order to check the usability of the application. The experimental setup and its description are done in chapter 6.2.

1.6 Ethical considerations
The AHP performs transparent structuring of a decision problem according to experts' knowledge. It does not guarantee “correct” decision. It helps decision makers find result that best suits their goal and their understanding of the problem. The outcome is defined solely by experts' judgments. Common problems concerning experts' judgments are those of consolidation and inaccurate judgments.

The main problem during consolidation of different experts' judgments occurs when the final result neither satisfies experts nor represents their initial judgments. For example, during the comparison of items A and B one expert claims that A is 5 times more important than B (A-5 - B-1) and other expert has an opposite opinion (A-1 - B-5). Then the average of their opinions is A-1 B-1, which though is mathematically correct, is not semantically valuable. To avoid this situation an external consistency check was introduced. With help of it, the decision owner will be informed about such problems, and he can organize additional meeting for discussion, instead of just calculating result which will be confusing for experts.
Moreover, if initial experts' judgments were inaccurate, AHP result would also be inaccurate. The usage of the AHP allows to analyze the decision-making process from experts' judgments to final consolidated result, so it is transparent enough for finding errors.

Another great advantage of the AHP is that it allows to focus on one small part (comparison) of a decision problem at a time. Attempts to take into account all criteria simultaneously can lead to long, boring and unproductive discussions. Moreover, performing pairwise comparisons on refined criteria tree is quick and simple. Thus, pairwise comparisons reduce the risk of an inaccurate result.

1.7 Report layout
The rest of the thesis is organized as follows. In chapter 2, an introduction to the AHP is provided, which is sufficient to understand the remainder of the thesis. Furthermore, some notions related to the AHP and used throughout this thesis are defined. In chapter 3, a refinement of the goal into functional and nonfunctional requirements is given. In chapter 4, use cases, activities diagrams, application's architecture and data storage are described. In chapter 5, implementation decisions are motivated and application's figures, corresponding to use cases, are shown. In chapter 6 the evaluation of the application is performed with respect to the goal criteria. In chapter 7 the conclusions obtained from the thesis are presented, and plans for future work are outlined.
2 Theory and background
In this chapter an introduction to the AHP is provided. It is sufficient to understand the remainder of the thesis. Furthermore, some notions related to the AHP and used throughout this thesis are introduced.

2.1 AHP theory description
According to the AHP, the decomposition of a decision is done in these steps [2]:

1. Define the problem and gather information related to it.
2. Structure the decision hierarchy from the top with the goal of the decision, then the objectives from a broad perspective (criteria with subsequent dependent sub-criteria) to the lowest level, which usually is a set of alternatives.
3. Construct a set of pairwise comparison matrices. Each element in the upper level is used to compare the elements in the level immediately below with respect to it. Thus, there are two types of comparisons: children criteria with respect to the parent criterion (branch criteria) and alternatives with respect to bottom level criteria (leaf criteria).
4. For each element, use the priorities obtained from the comparisons to weight the priorities in the level immediately below. Then for each element in the level below add its weighted values and obtain its overall or global priority.

<table>
<thead>
<tr>
<th>Intensity of Importance</th>
<th>Definition</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Equal importance</td>
<td>Two activities contribute equally to the objective</td>
</tr>
<tr>
<td>1.1 - 1.9</td>
<td>If the activities are very close</td>
<td>Such small numbers will not be too noticeable compared to other weights</td>
</tr>
<tr>
<td>2</td>
<td>Weak or slight</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Moderate importance</td>
<td>Experience and judgment slightly favor one activity over another</td>
</tr>
<tr>
<td>4</td>
<td>Moderate plus</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Strong importance</td>
<td>Experience and judgment strongly favor one activity over another</td>
</tr>
<tr>
<td>6</td>
<td>Strong plus</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Very strong or demonstrated importance</td>
<td>An activity is favored very strongly over another</td>
</tr>
<tr>
<td>8</td>
<td>Very, very strong</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Extreme importance</td>
<td>The evidence favoring one activity over another is of the highest possible order</td>
</tr>
</tbody>
</table>

Table 2.1: The fundamental scale of absolute numbers

To make comparisons, the AHP suggests a scale of numbers that indicates how much more important one element is compared to another with respect to comparison basis (cf. 2.2), against which they are compared. Table 2.1 presents the scale [2].
In addition, the AHP incorporates a useful technique for checking the consistency of the expert’s evaluations – internal consistency check (cf. 2.2). This check is done after each expert enters pairwise comparisons. Internal consistency check ensures that judgments of an expert are mutually concordant thus reducing the bias in the decision-making process.

To incorporate experts' opinions, adding corresponding step to the AHP is proposed. During this step check for the consistency of the opinions (external consistency) is performed. This check ensures that experts have rather agreed opinions and leaves room for some oscillations in opinions (within estimated boundaries).

After both consistency checks are done, consolidated results are calculated. These results include prioritized criteria tree and ranking of alternatives.

### 2.2 Notions and definitions

Now we will introduce main notions and definitions used in the thesis.

![Decision tree example](image)

Decision tree - data structure that contains information related to the decision. It consists of a goal, subgoals (criteria) and sub-criteria (all combined into criteria tree) and alternatives on the bottom level. Example of a decision tree is on figure 2.1.

Owner - a user who creates the decision and then accesses it by an administration link to add alternatives, criteria.

Expert - a user who participates in the decision by a participation link.

Goal – the main problem being analyzed in the decision, a root of the criteria tree.

During criteria tree construction, a goal is decomposed into a hierarchy (criteria tree) of easily comprehended subproblems (criterion), each of which can be analyzed independently.

Criterion - sub-problem of the goal or parent criterion. If it is not possible to compare alternatives with respect to the criterion, it should be decomposed into several sub-criteria, which will be children criteria of the parent criterion.
Criteria tree - hierarchy of goal and sub-goals (criteria). It is a decision tree without alternatives. Goal is a root of hierarchy and criteria are branches and leafs.

Alternative - item which is compared with other similar items (alternatives) with respect to the goal. Since the goal is decomposed into the criteria tree, alternatives are compared with respect to each bottom level criteria.

Leaf criterion - criterion on the bottom of the criteria tree.

Branch criterion - criterion, which is not a leaf criterion. Includes the goal and all criteria that have at least one child criterion.

Pairwise comparison - way for expert to input his judgments. According to the AHP, during pairwise comparison expert sets weights for comparates with numbers from the fundamental scale of absolute numbers (table 2.1). Pairwise comparison consists of comparison basis and two comparates.

Comparate - one of the two items being compared together in a pairwise comparison. Depending on comparison basis comparates can be alternatives (leaf criterion as a basis) or criteria (branch criterion as a basis).

Comparison basis - criteria with respect to which comparates are compared.

Leaf comparison - comparison whose comparison basis is leaf criterion. Comparates of such comparison are alternatives.

Branch comparison - comparison whose comparison basis is branch criterion. Comparates of such comparison are criteria.

Consistency - state of weights being proper enough for further AHP processing. Can be internal (within one expert’s judgments with common comparison basis) and external (among judgments of all experts within one comparison).

Inconsistency - state of weights being not proper enough to get useful results from the AHP.

Internal inconsistency - in simpler terms, it ensures that judgments of an expert are mutually concordant. In details, it is defined by notion of the consistency index. Consistency index is calculated based on a principal eigenvalue of a comparison matrix. Later this index is compared with the same index obtained as an average over a large number of reciprocal matrices of the same order whose entries are random. If the ratio of the consistency index to that from random matrices is significantly small (10% or less), we accept comparison matrix. Otherwise, we attempt to improve consistency [2].

External inconsistency - external consistency check ensures initial agreement of experts on a comparison. Without this step representativeness of consolidated judgment can be hampered.

For example, combining judgments 1-9 and 9-1 with a geometric mean will provide judgment 1-1, which is correct from the mathematical point of view, but does not represent any of experts' initial opinions. To avoid this situation, external consistency check was introduced. It ensures that experts' opinions agree enough to get meaningful resulting priorities. External consistency check is done by ensuring that opinions of experts differ within defined distance between each other (it is set by default to 3). For example judgments \(\{1-2, 1-4, 1-5\}\), \(\{1-1, 1-2, 1-3\}\), \(\{2-1, 1-2, 1-1\}\) are considered consistent, and judgments \(\{1-2, 1-4, 1-6\}\) are considered inconsistent because of outlier 1-6. In judgments \(\{3-1, 1-2, 1-3\}\) 3-1 is an outlier, in \(\{3-1, 1-2, 1-3\}\) - both 3-1 and 1-3 are outliers.
3 Problem definition refinement

In this chapter the refinement of the goal is provided. It is refined into functional and nonfunctional requirements. Functional requirements consist of features. Non-functional requirements consist of portability, free availability, usability. Then competitors are evaluated against features as well as portability and free available criteria. Since competitors do not satisfy these requirements, there is no point in assessing their usability. Development of a new tool is required. The usability of this new tool will be later assessed with experiment.

3.1 Refining goal

According to the AHP, steps of the decision-making process and corresponding required features for an application are:

F1. Decision creation, goal specification.
F2. Decision tree (cf. 2.2) development: add, remove, modify alternative; add, remove, modify criterion.
F3. Multiple experts can perform pairwise comparisons: comparing alternatives with respect to leaf criterion (leaf comparison - cf. 2.2); comparing criteria with respect to parent branch criterion (branch comparison - cf. 2.2).
F4. Describe the contribution of each user into the final result.
F5. Work without alternatives - developing and prioritizing criteria tree.
F6. Consolidation of different experts opinions on the results stage: automated consolidated results calculation and owner controlling the consolidation process.
F7. External consistency check before final results calculation to improve accuracy and transparency of the results.

Steps F2, F3 can be done in parallel: a decision owner (cf. 2.2) can add new criteria or alternatives to the decision while experts perform pairwise comparisons on existing parts of the decision tree.

Unit tests were developed to ensure correct implementation of the above mentioned functional requirements. These features are the basis for use cases (cf. 4.1) so application is focused on implementing them. Checking of this functionality working together (integration testing) is done in portability tests and usability user experiment.

3.2 Refining goal criteria

The portability [12] goal criterion tests use cases on mobile and PC platforms. Thus, it can be decomposed into two refined criteria:

P1) access from PC web-browser;
P2) access from mobile devices;

These criteria can be observed by tests without additional experiments.

Free available goal criterion defines whether the application can be used without paying a fee. Possible values are: not - cannot be used without fee; trial - trial version is available; free - can be used without fee.

The usability [10] goal criterion defines the understanding of the usability used in the thesis. It consists the list of measures taken and implemented to improve the usability. In evaluation chapter, an user experiment will be conducted to check whether these measures were successful and whether developed application is user-friendly and
tool-enabled decision-making process is better than manual. Each usability measure also
lists features, influenced by this usability improvement.

- U1) (F4) final result should be clear and users should see their contribution to it;
- U2) (F2, F5) application should not force users to add alternatives and allow
prioritization of standalone criteria tree;
- U3) (F5) users should clearly see which comparisons are left to be performed by
them in the decision;
- U4) (F3, F5, F6, F7) error message about inconsistent judgments should be clear
and help the user make corresponding changes;
- U5) (F6, F7) decision owner should be able to have vision and control of experts'
opinions consolidation process;

User experiments will be conducted to check whether these measures improved

### 3.3 Evaluating competitors against refined goal criteria

Evaluation of competitor applications with respect to features and portability and free
available criteria is done here. Since they do not provide required features, they are not
suitable for solving thesis problem. Thus, there is no point in assessing their usability.
Results are presented in Table 3.1.

<table>
<thead>
<tr>
<th>Competitor</th>
<th>Features</th>
<th>Portability</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>MakeItRational</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Transparent Choice</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Super Decisions</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.1: Competitors evaluation results

MakeItRational [16] is the best solution of analyzed competitors and the most
expensive. Free trial mode is available, but it is not accessible from mobile phones. This
solution provides almost all required features, though consolidation of experts' opinions
is done manually (“if you cannot agree - use the average value”). Provides good
statistics for analyzing results.

Transparent choice [17] is also web-accessible but has less functionality compared
to MakeItRational. No error message is provided when weights are inconsistent. No
consolidation of experts' opinions is available, they are regarded separately.

Super Decision [18] is not web-accessible, is only installable as a desktop
application. It is more academic oriented than previous two. It provides fewer features
and is free.
3.4 Conclusion
As it was shown, existing solutions do not provide required functionality: criteria tree prioritization without alternatives, automated consolidation of different experts opinions with external consistency check. Moreover, none of them provides mobile adapted version of the application.

None of the existing applications satisfies defined criteria. Thus, new application needs to be developed. To check whether the developed application complies with goal criteria, tests will be performed, and usability user experiment will be conducted.
4 Architecture and design
In this chapter use cases, derived from functional requirements (features), are described. General structure of the Model View Presenter architecture is outlined. Then application of this architecture in developed tool is presented. Further data storage structure is described. Mapping of features, use cases and activities diagrams is done.

4.1 Use cases
Mapping of the use cases [13] to the features is done in Table 4.1.

Two roles in the system are: Owner - creates and administers decision, Expert - participates in the decision by performing pairwise comparisons.

Owner use cases are: Create decision, Modify decision, which includes editing alternatives, criteria and expert judgments, View results. Owner cannot perform pairwise comparisons. To do so, accessing through participate link is required. In that case owner will be regarded as one of the experts.

Expert use cases are: Perform pairwise comparisons, which implies comparing alternatives with respect to leaf criterion and comparing criteria with respect to parent branch criterion. After perform comparisons stage, expert can view results and observe his contribution to final priorities.

Figure 4.1: Use case diagram
4.2 Architecture
Frontend is designed using the Model View Presenter (MVP) [4] GUI design pattern. Diagram depicting MVP pattern is on figure 4.2.

Figure 4.2: MVP pattern diagram, [4].

MVP pattern is engineered to facilitate automated unit testing and improve the separation of concerns in presentation logic. Pattern components are:

1) model is an interface defining the data to be displayed or otherwise acted upon in the user interface;

2) view is a passive interface that displays data (the model) and routes user commands (events) to the presenter to act upon that data;

3) presenter acts upon the model and the view. It retrieves data from repositories (the model), and formats it for display in the view. It contains UI business logic for the view.

Main advantages of MVP pattern are:

1) view is more loosely coupled to the model. The presenter is responsible for binding the model to the view and handling user events;

2) easier to unit test because interaction between the presenter and the view is through an interface. This decoupling allows mocking of the view in a unit test.

Since portability is an important goal criterion, separation of view and presenter is extremely useful in providing different views for mobile and PC users while reusing same presenter. Moreover, decoupled view allows developing and enhancing it independent from presenter and model, which improves usability of the application.

MVP pattern implementation in the application is done according to [5] and is presented on figure 4.3.
A model represents business objects, and in the application those are: Decision, Alternative, CriteriaTreeNode, Person, Expert, Comparison, ExpertOpinion (cf. 4.3).

A view contains all of the UI components that make up our application. These components are tables, labels, buttons, textboxes, etc. Views are responsible for the layout of the UI components and have no notion of the model. Views implement Display interface of corresponding Presenter. Switching between views is tied to the history management within the presentation layer.

A presenter contains all of the logic for the application, view transition and data interactions with server RPCs.

To handle logic that is not specific to any presenter and instead resides at the application layer, AppController component is introduced. This component contains the history management and view transition logic. View transition is directly tied to the history management.

Application events (like changing view, RPC call result) are transmitted via Event Bus. Presenters add handlers on user events (button click, list item selection), and transform them into application events, which are sent to Event Bus. Handles to events from Event Bus are added in AppController. History management is also implemented by AppController.
AppController handles application events from Event Bus and history change events. In handlers to both application events and history events corresponding presenters are initialized, and view transitions are performed.

Using MVP design pattern allowed to increase the separation of concerns and facilitated unit test coverage.

4.3 Design description

Since Google App Engine (more about this decision in chapter 5) is used for hosting of the application, Google Datastore [8] is used as a storage for the application.

![DB structure diagram]

The list of data entities:

1) Owner - represents decision owner. It is the user who created the decision and administers it by administration link.

2) Decision - represents decision. Stores ownerId, title and decision id.

3) Alternative - decision alternative. Stores id of the decision, alternative id, name and calculated priority.

4) CriteriaTreeNode - decision criteria. Stores id of the decision, the node id and parent node id, node name and node local priority (with respect to the parent).

5) Comparison - comparison (both leaf and branch). Stores id of the decision, comparison basis id (id of criteria tree node) and ids of comparates.

6) Expert - user that participates in the decision by participation link. Stores decision id, user name and expert id.
7) ExpertOpinion - stores pairwise comparison of an expert, specified by expertId field. Comparison is defined by comparisonId field.

To model workflow, activity diagrams are used. Mapping of features (cf. 3.1), use cases (cf. 4.1) and activities diagrams (below) is described in table 4.1:

<table>
<thead>
<tr>
<th>Features. General functional requirements without roles</th>
<th>Use cases. Define Owner and Expert roles</th>
<th>Activities diagrams. Describe how roles are implemented in the work flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1. Decision creation, goal specification</td>
<td>(role: Owner) Create decision</td>
<td>Figure 4.7 - Decision creation block</td>
</tr>
<tr>
<td>(part of F3) Enable multiple experts participation</td>
<td>Access by participate link. Role: Expert</td>
<td>Figure 4.6 - Decision participation</td>
</tr>
<tr>
<td>F2. Decision tree development: add / remove / modify alternative / criterion. F5. Enable work without alternatives.</td>
<td>(role: Owner) Modify decision: add / remove / modify alternative / criterion.</td>
<td>Figure 4.7 - Decision administration. Figures 4.9 - 4.11 - Add / Remove / Modify alternative or criterion.</td>
</tr>
<tr>
<td>F3. Multiple experts performing pairwise comparisons</td>
<td>(role: Expert) Pairwise comparison of alternatives or criteria</td>
<td>Figure 4.8 - Pairwise comparisons</td>
</tr>
<tr>
<td>F4. Describe users contribution to result. F6. Consolidation of experts opinions. F7. External consistency check.</td>
<td>(role: both) View results</td>
<td>Figure 4.12 - View results</td>
</tr>
</tbody>
</table>

Table 4.1: Features, use cases, activities diagrams mapping
Activities diagrams [14] are constructed from such shape types:

- rectangles represent actions;
- rectangles with double-struck vertical edges represent routines - operations that are described with separate activity diagram.
- diamonds represent decisions, text on edges represent possible continuations of workflow;
- bars represent the start or end of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
- an encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen. Application activities diagrams are presented further.

![Application entry point activity diagram](image)

**Figure 4.5: Application entry point activity diagram**

Owner accesses decision by the administration link. Expert accesses decision by the participation link. Decision participation and Decision administration are described on figures 4.6 and 4.7 respectively:
Figure 4.6: Decision participation activity diagram
Expert can perform Pairwise comparisons (figure 4.8) and View results (figure 4.12).

Figure 4.7: Decision administration activity diagram
Decision administration involves Decision creation, Add/Remove/Modify alternative/criterion and View results.
Activity diagram is the same for both comparisons of criteria and comparisons of alternatives. Internal consistency check is performed after weights input.
Figure 4.9: Add criterion (left), Add alternative (right) activity diagrams

During the addition of a criterion to a parent, it is important to check whether it is first child of the parent criterion. If it is, then the parent criterion is turning from leaf criterion into branch criterion. Then comparison of alternatives with respect to the parent should be removed. If it is not the first child, then branch comparisons of other children criteria with respect to the new criterion should be added. Then priorities on other children of the parent should be removed, and new leaf comparisons of all alternatives with respect to the new criterion should be added.

Add alternative activity is simpler. First, new leaf comparisons of the new alternative with each of existing alternatives with respect to each leaf criteria are added. Then priorities on other alternatives are removed.
Figure 4.10: Remove criterion (left), Remove alternative (right) activity diagrams

Remove criterion is only allowed for leaf criteria. Removing branch criterion would require more steps and will remove many comparisons from decision, which may be confusing for the user.

Remove leaf criterion removes comparisons involving this criterion. Those are branch comparisons with respect to the parent criterion and leaf comparisons with respect to removing leaf criterion. After that priorities of alternatives are recalculated.

Remove alternative also removes comparisons involving this alternative. Since those are only leaf comparisons, only priorities on alternatives are recalculated.
Modify activity is the same for a criterion and an alternative. First, a new name is specified. Then user selects whether it is renaming or replacement. Renaming just changes name. Replacing removes selected item and adds new with the new name. Replace involves Remove and Add activities, described earlier.
In View results check for completeness of data is performed first. This involves checking whether all experts performed their comparisons and whether each expert comparisons are internally consistent. If some statement of abovementioned is false, message describing the problem is shown. External consistency check is an advisory mechanism. It does not directly influence results calculation. Consequently, it is not automatically checked before results calculation. It can be done by owner at any state of the decision to observe general progress of experts on the comparisons and to notice outliers or disagreements early.
5 Implementation details
In this chapter some implementation decisions are motivated and figures of the application implementing use cases and usability measures are shown.

GWT [6] was selected for building browser-based application. Google Application Engine (GAE) [7] was selected for hosting and deployment of the application because it allows simple one-click deployment and provides free hosting. Since GAE was selected as deployment platform, Google Datastore [8] as the persistence mechanism was required. Moreover, it provides local implementation of Datastore API, which allows local execution of JUnit tests.

Example decision being solved is Analytic hierarchy process — Leader example from en.wikipedia.org.

5.1 Use cases implementation
Section shows figures of the application implementing the use cases.

Use case 1: (Owner) Create decision.

![New decision data](image)

<table>
<thead>
<tr>
<th>Decision title*</th>
<th>Wiki example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision goal*</td>
<td>Choose a leader</td>
</tr>
<tr>
<td>Your name*</td>
<td>Igor</td>
</tr>
<tr>
<td>E-mail address*</td>
<td><a href="mailto:Igor@example.com">Igor@example.com</a></td>
</tr>
</tbody>
</table>

Create

Figure 5.1: Creating new decision

After a decision is created, participation and administration links are generated.
Use case 2: Access by participate link (multiple experts participation). Expert and Administrator roles are implemented in expert and owner web pages respectively. These pages are accessible by participation and administration links, generated in during decision creation.

![Decision owner page (left), expert page (right)](image)

Use case 3: (Owner) Decision modification: add/remove/modify an alternative or a criterion.

![Decision owner page criteria tab](image)

Adding a criterion is performed by entering criteria name, selecting parent and clicking ‘Add’ button. Let’s add ‘Experience’, ‘Education’, ‘Charisma’, ‘Age’ criteria to parent ‘Goal’:

![Decision owner page criteria tab with criteria tree](image)

Criteria can be removed by clicking ‘Remove’ button.
Alternatives are managed on alternatives tab:

![Alternatives tab](image)

**Figure 5.5: Decision owner page alternatives tab**

To add an alternative user should write alternative name and click ‘Add’.

Alternatives tab after adding Tom, Dick, Harry alternatives:

![Alternatives with alternatives](image)

**Figure 5.6: Decision owner page alternatives tab with alternatives**

Alternatives can be removed by clicking ‘Remove’ button.

**Use case 4: (Expert) Pairwise comparison.** Comparison page contains list of comparisons, combined in groups by comparison basis, with internal inconsistency calculated.

New expert can be added in participation page, by entering expert name and clicking ‘Perform comparisons’ button:

![Experts page](image)

**Figure 5.7: Experts page before and after adding new expert**
By clicking on ‘Edit’ button near Expert name, expert comparisons are shown. Comparisons are grouped by common comparison basis and groups are sorted from goal to leaf criteria. After clicking on ‘Edit’ button pairwise comparison window is opened:

![Figure 5.8: Expert comparison page (left); pairwise comparison page (right)](image)

Status bar on the bottom of each comparison group contains results of consistency check for this group. Possible results of consistency check are presented on figure 5.9 and include:
1. (red) Please perform comparisons.
2. (red) CR>0.1 - comparisons are not internally consistent.
3. (green) CR<0.1 - comparisons are consistent.

![Figure 5.9: States of comparison group](image)

**Use case 5: View results.** Results are prioritized criteria tree and ranking of alternatives. Description of each expert's contribution to the result is also shown.

Weights for comparisons are entered according to the example. Now experts tab of the decision administration page shows that all comparisons are performed:
Figure 5.10: Decision experts tab after comparisons performed

Now we can click on ‘Results’ button and see results: prioritized criteria tree and alternatives rankings:

Figure 5.11: Results: criteria tree (left) and alternatives (right)

Comparisons tab of decision administration page implements the external consistency check and also gives a nice overview and control over experts' opinions. Before the external consistency check is performed, a consistency step is specified (Figure 5.12). After click on ‘Check consistency’ button, decision experts' opinions are shown. Experts' opinions are grouped by comparison, and external inconsistency is defined for each group. Comparisons are grouped by common comparison basis. Comparison bases are sorted tree-wise, from goal and branches (branch comparisons) to leaves (leaf comparisons).

Figure 5.12: Comparisons tab
To see the external consistency check in action, new expert is added to the decision. After performing comparisons, experts tab shows two users with all comparisons done:

<table>
<thead>
<tr>
<th>Experts</th>
<th>Criteria</th>
<th>Alternatives</th>
<th>Comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Comparisons</td>
<td>Edit</td>
<td>Remove</td>
</tr>
<tr>
<td>Igor</td>
<td>Comparisons: 18/18</td>
<td>Edit</td>
<td>Remove</td>
</tr>
<tr>
<td>Welf</td>
<td>Comparisons: 18/18</td>
<td>Edit</td>
<td>Remove</td>
</tr>
</tbody>
</table>

Figure 5.13: Experts tab with two experts

Now comparisons tab shows opinions of both experts. States of the comparison group are similar to states of expert comparisons: Consistent, Inconsistent, Not enough data. When facing external inconsistency, owner can increase step or talk with experts and resolve differences in opinions or manually update expert opinions by clicking ‘Edit’ button.

Figure 5.14: Comparisons tab

When two experts are present in a decision, consolidated results are calculated. They are shown after clicking on each leaf criterion or alternative panel in results page.
5.2 Usability measures implementation

Usability measures being implemented were derived during usability refinement (cf. 3.2). Implementation of these measures with corresponding figures is shown below.

U1) Final result should be clear, and users should see their contribution to it. This measure is implemented at ‘Results’ page, shown on figure 5.15.

U2) Application should not force users to add alternatives and allow prioritization of standalone criteria tree. Owner can skip the addition of alternatives at all or remove existing alternatives (figure 5.6). Then ‘alternatives’ tab of decision will be empty (figure 5.5), comparisons view for expert will contain only branch comparisons and results view will only show prioritized criteria tree and no alternatives tab (figure 5.16).
U3) Users should clearly see which comparisons are left to be performed by them in the decision. Each expert observes all comparisons to be performed in expert comparisons page (figure 5.8). On this page, comparisons are grouped by comparison basis and state of each group is indicated (figure 5.9). Participate home page (figure 5.7) shows how many comparisons each expert has performed.

U4) Error message about inconsistent judgments should be clear and help the user to make corresponding changes. When comparisons are inconsistent, it is indicated with the corresponding color of the comparison group. After click on the info button, a message is shown (figure 5.17). This message can be upgraded with suggestions what comparisons to change in order to get the consistent result. This feature is not implemented yet and is now in future work section.
U5) Decision owner should be able to have vision and control of experts' opinions consolidation process.

Vision and control over consolidation process are implemented in ‘Comparisons’ tab of owner decision page (figure 5.14). Vision implies an overview of each expert opinion on each comparison. Control implies an external consistency check, which assists with finding outliers and disagreements in opinions. Step for the external consistency check is manually specified. Thus, owner can control the precision of checking.
6 Evaluation
In this chapter the description of each goal criterion's check is given. The portability is checked with tests; the free available criterion is simply observed. The usability criterion is checked via the user experiment by observing usability indicators (number of questions, time to finish the task) and comparing them with manual work.

6.1 Portability goal criterion
The portability criterion implies program availability for both mobile platforms and PC web browsers. It is tested by performing use cases on a chosen platform. Figures of the application working on the web platform are in chapter 5. Figures of application working on the mobile platform are below.

1. (Owner) Create decision

![New decision data](image)

**New decision data**

- **Decision title**
- **Decision goal**
- **Your name**

[Create]

Figure 6.1: Creating new decision page

After decision creation participation and administration links are generated.
2. Implementation of Expert, Administrator roles - access by participation and administration links, generated in previous step.

3. (Owner) Decision modification: add/remove/modify alternative/criterion.

![Pairwise comparison](image1)

**Figure 6.4: Pairwise comparison**

5. View results. Prioritized criteria tree and alternatives rankings.

![Decision results](image2)

**Figure 6.5: Results: criteria (left) and alternatives (right)**
6.2 Free available goal criterion
The application is free to use. Both decision administration and participation parts do not require application users to register or to log in.

6.3 Usability goal criterion
Measures intended to improve usability were defined in chapter 3.2. The implementation of these measures was presented in chapter 5.2. In this chapter, user experiments will be conducted, and their results will indicate whether taken usability measures were effective.

The usability is twofold: for experts performing comparisons - experts experiment (EE) and for owners and experts working together - owners-experts experiment (OEE).

EE is focused on checking the usability of users working with features F4 (describe the contribution of each user) and F5 (working without alternatives - criteria tree prioritization). OEE is focused on checking the usability of users working with features F6 (owner control on opinions consolidation) and F7 (external consistency check). Thus, experiments cover all new features, which were not present in competitor tools.

Both EE and OEE were designed. However, due to lack in time and resources, only EE was conducted.

Experiments goal description according to [9] is below.

EE goal:
Analyze the process of experts participating in criteria tree prioritization, both manual and with the tool (designed application).

For the purpose of comparing manual and automated criteria tree prioritization.
With respect to usability.
From the point of view of the decision participants - experts.
In the context of a decision-making problem.

OEE goal:
Analyze the process of owner collecting experts' opinions, consolidating them and deriving final priorities from them, both manual and with the tool.

For the purpose of compare manual and automated experts opinions consolidation.

With respect to usability.
From the point of view of owner and experts.
In the context of decision-making problem.

6.3.1 Experiment setup description
EE setup:
1) hypothesis - tool-enabled criteria tree prioritization is more user-friendly;
2) subjects - people (2 or more) with no AHP knowledge or experience;
3) variables:
  3.1) input (independent) variables:
   - object: decision criteria tree;
   - treatment: pairwise comparisons of criteria with respect to their parent;
   calculation of priorities of criteria derived from comparisons using the AHP.
  3.2) result (dependent) variables:
   - prioritized criteria tree and description of each user contribution to the result, derived using the AHP;
number of questions asked during task execution;
- time to finish the task;
- user feedback about the task complexity.

4) experiment schedule includes execution of two tasks. Manual task is expected to be done in 20 - 40 minutes. Tool-enabled task is expected to be done in 5 - 10 minutes.

OEE setup:
1) hypothesis - tool-enabled experts opinions consolidation is more user-friendly;
2) subjects - people with some or without AHP knowledge or experience;
3) variables:
3.1) input (independent) variables:
- object: decision goal;
- treatment: dividing goal into criteria and sub-criteria, forming criteria tree; adding alternatives; forming pairs of criteria to compare with respect to parent criterion and pair of alternatives to compare with respect to leaf criteria; experts performing pairwise comparisons; processing of experts input; calculation of priorities of criteria and alternatives derived from comparisons using the AHP.
3.2) result (dependent) variables:
- decision tree, criteria and alternatives priorities as AHP results;
- number of questions asked during task execution;
- time to finish the task;
- user feedback about task complexity.

4) experiment schedule includes execution of two tasks. Manual task is expected to be done in 30 - 60 minutes. Tool-enabled task is expected to be done in 10 - 20 minutes.

6.3.2 Execution of experiment
Execution plan of both experiments is described below. Since only EE was conducted, only EE results are analyzed.

EE execution:
1) operation
Subjects are given a short introduction to the AHP (Appendix A.1) and the decision criteria tree (Appendix A.2). Subjects are asked to perform required pairwise comparisons and (in the case of manual work) calculate the priorities of criteria and determine each user contribution to the final result (Appendix A.3). Subjects' questions, time to finish the task and general feedback about task complexity are measured.

2) results
Participants were provided with an introduction to the AHP, the decision criteria tree example and the task to be done (all in Appendix A). One group of participants performed the task with the help of a spreadsheet program, other with the help of the tool.

First group finished task in 15 minutes, second in 12 minutes. Thus, there is some time improvements using tool, however not substantial since features being under focus in experiment were not very time-consuming.
First group during experiment:

1) stated that the scale is rather big and confusing. During comparisons they had to come back to scale, to see what scale numbers mean. As a result, they could not focus on comparisons;

2) asked how to combine together weights (there were three experts in the group, so they had to use cubic root for calculations);

3) had some disagreements over final results. Users with very low or very high initial opinions were not satisfied with the final result since it was significantly different from their initial opinions. In program, this process is automated, and users see final results and each expert contribution to it. It is almost the same as in a spreadsheet program, so tool does not make much difference here. It makes a difference during the external consistency check which allows to find disagreement before result calculation. However, this was not the focus of this experiment.

Second group did not have any additional questions, only some disagreements over final results. In the tool, text corresponding to each scale number is displayed, and consolidated weights calculation is automated. Consequently, this questions did not rise during work with the tool.

Thus, in comparison with a semi-manual work with a spreadsheet, the number of question while performing task with tool is reduced.

Both groups provided some general feedback about the AHP:

1. Numbers in scale can be misleading. They correspond to different text descriptions like ‘slightly favoured’, ‘strongly favoured’ (cf. Table 2.1). At the same time, the user can understand them as ‘times more important’, which is incorrect. This issue can be solved by spending more time for the explanation of the scale. As an alternative solution, numbers can be eliminated at all from comparison stage. Only corresponding text should be presented to the users. This approach would be suitable for inexperienced users. This idea was added to future work section.

2. Not every user was satisfied with program UI appearance. Some buttons were too small; text labels unaligned. Since the application was still in development, corresponding changes were implemented in subsequent versions.

OEE planned execution:

1) operation

Owner is given a decision goal. In collaboration with the group of experts, he constructs criteria tree and add alternatives. After decision tree construction, comparison pairs are formed, and experts perform pairwise comparisons. The owner collects this input and calculates result priorities of alternatives and criteria.

2) results

Subjects questions, time to finish the task and general feedback.

6.3.3 Discussion of results

EE results interpretation:

Usage of the application allowed to reduce the time to perform the task by 3 minutes. Almost none additional questions were asked during work with the application. Complex calculations of priorities are automated. Thus, we can conclude that usage of developed application improved users' AHP experience.
EE validity threats:

Experiment was conducted only with one pair of participant groups (first group - manual part, second group - work with the tool). Thus, results cannot be generalized.

EE conclusion:

Main advantages of the usage of the tool for decision making are: covering AHP details and complex calculations from users, automated calculation and clear presentation of each user contribution to the result. This resulted in faster decision making; fewer questions asked and more positive feedback from users. Thus, tool-enabled decision making is more user-friendly that manual process.

6.4 Conclusion

Application development was based on use cases. Use cases were derived from features. This development process allowed to develop application satisfying required functional requirements. It is portable (6.1), free available (6.2), and usability measures implemented ensured that developed application is usable (6.3).
7 Conclusion, future work

In this chapter conclusions obtained from the thesis are described and plans for future work are outlined. These plans consist of experiments, features and integration sections.

7.1 Conclusion

In first chapter the background of the thesis was analyzed, and the goal was defined and refined into functional and nonfunctional constituents. Thesis scope was limited to mostly practical implementation of the AHP with small adjustments to increase accuracy of multiple experts' consolidated opinions. Main source for possible errors is errors in initial experts' judgments, this was elaborated in the ethical considerations part.

Second chapter provided a short introduction to the basics of the AHP theory. Later some notions related to the AHP and used throughout this thesis were defined.

In the third chapter, the complete refinement of the goal was performed. Competitors were evaluated, and it was established that none satisfies the goal completely, so design and development of a new tool is necessary.

Fourth chapter described how application was designed from functional features to use cases and then to activities diagrams. MVP design pattern was used to form the architecture of the application, and the Google Datastore was used for persistence.

Fifth chapter provided additional implementations details and figures of application corresponding to use cases.

After the application had been developed, it was insured in the previous chapter that it implements the use cases, thus satisfying functional requirements, non-functional requirements were tested in chapter 6. Different testing methods were used for each goal criteria: tests for the portability, observation for the free availability and user experiment for the usability. It was established that the application satisfied required goal criteria; thus, thesis goal is achieved.

Main differences between existing tools and the developed tool are extending the AHP with the external consistency check and providing a mobile version of the application.

7.2 Future work

Future work can be divided into next subsections: experiments, features, integration with other tools.

Experiments. The usability experiment is just a study, and more extensive one should be performed to improve application assessment. Alternatively, instead of performing costly extensive experiment, application can be released for users and their feedback should be analyzed as the main indicator of application usability.

Features to be included into future versions of the application are:

1. Internal inconsistency suggestion. Provide not only a message showing which comparison group is inconsistent, but also suggest which comparisons should be changed and how, in order to get consistent results.

2. Users accounts. Users can create accounts, where they can see address book with recently used emails, and the list of their decisions.

3. Experts participating in decision tree construction. Allow experts to add/remove criteria and alternatives.
4. “Don’t know” option in comparisons. If an expert does not answer to comparison, instead of guessing, “don’t know” option can be selected. Then priorities will be calculated from other experts opinions.

5. To avoid “don’t know” situation, experience area can be attached to each expert. Experience area is a subtree of criteria tree. If an expert is assigned with experience area, then he will only participate in comparisons from that area.

6. Manual definition of priorities. Sometimes priorities are already defined in external application or documents. Then it is a good idea to allow the user to set priorities manually instead of forcing them to perform comparisons.

7. Absolute rating scale as an alternative to pairwise comparisons. According to absolute rating scale approach, first an absolute scale is defined and then comparates are compared against the scale. Advantage in this case is fewer efforts when working with decision involving many comparates.

8. Show only text description of the scale number during comparison. This can be easier to understand and to work with for an inexperienced user.

Integration with other tools: application functionality can be useful in other areas, for example, resource allocation or software quality monitoring. Thus, appropriate API for automated integration should be available.
Reference literature

A Appendix
Contains documents related to the Expert Experiment setup and execution: AHP introduction, decision tree, experiment task. Topic is highly dependent on personal preferences; thus, different priorities are expected. However, external inconsistency check is not involved in this experiment, so this is not a big issue. Here I check usability (cf. 3.2) for a single expert participating in the decision.

A.1 AHP introduction
The AHP is a structured technique for organizing and analyzing complex decisions. According to the AHP, decomposition of a decision is done in following steps:

1. Define the problem and gather information related to it.
2. Structure the decision hierarchy from the top with the goal of the decision, then the objectives from a broad perspective (criteria with subsequent dependent sub-criteria) to the lowest level, which usually is a set of the alternatives. Hierarchy example:

   
   ![Decision hierarchy example](image)

   Figure A.1: Decision hierarchy example

3. Construct a set of pairwise comparison matrices. Each element in the upper level is used to compare the elements in the level immediately below with respect to it. Thus, we have two types of comparisons: children criteria with respect to the parent criterion (branch criteria) and alternatives with respect to bottom level criteria (leaf criteria).

4. For each element use the priorities obtained from the comparisons to weight the priorities in the level immediately below. Then for each element in the level below add its weighted values and obtain its overall or global priority. Continue this process of weighting and adding until the final priorities of alternatives in the bottom most level are obtained.

   To make comparisons, the AHP suggests a scale of numbers that indicates how much more important one element is compared to another with respect to comparison basis, against which they are compared. Table A.1 presents the scale
<table>
<thead>
<tr>
<th>Intensity of Importance</th>
<th>Definition</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Equal importance</td>
<td>Two activities contribute equally to the objective</td>
</tr>
<tr>
<td>2</td>
<td>Weak or slight</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Moderate importance</td>
<td>Experience and judgment slightly favour one activity over another</td>
</tr>
<tr>
<td>4</td>
<td>Moderate plus</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Strong importance</td>
<td>Experience and judgment strongly favour one activity over another</td>
</tr>
<tr>
<td>6</td>
<td>Strong plus</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Very strong or demonstrated</td>
<td>An activity is favoured very strongly over another; its dominance</td>
</tr>
<tr>
<td></td>
<td>importance</td>
<td>demonstrated in practice</td>
</tr>
<tr>
<td>8</td>
<td>Very, very strong</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Extreme importance</td>
<td>The evidence favouring one activity over another is of the highest</td>
</tr>
<tr>
<td></td>
<td></td>
<td>possible order of affirmation</td>
</tr>
</tbody>
</table>

Table A.1: The fundamental scale of absolute numbers

AHP detailed example:
http://en.wikipedia.org/wiki/Analytic_hierarchy_process_%E2%80%94_Leader_example

A.2 Decision criteria tree
Decision criteria tree being prioritized in the experiment.

![Decision criteria tree](image)

Figure A.2: Decision criteria tree
A.3 Task

Manual work
Each participant gets a link to a spreadsheet document for priorities calculation:
https://docs.google.com/spreadsheets/d/1Rft21Q8ukFQ6h76OXPDQot0ELR70Uj
QAQCf5ivRPNJ0/edit?usp=sharing
This document does not provide precisely correct AHP calculations, but it is
suitable for the experiment.
Each participant is supposed to:
1) duplicate ‘Base sheet’;
2) rename new copy to a participant name;
3) perform comparisons by selecting a value for cells with green background from
drop-down list. Please confirm to the AHP and provide weights like N-1, 1-1, 1-N;
where N is one of the numbers in fundamental scale table;
4) get result priorities in the bottom of the spreadsheet from cells with blue
background.
After that participants are asked to combine their results by geometric mean and
determine each participant contribution to the result. This is done in results sheet of the
spreadsheet.

Automated work
Each participant gets a link to the participate page of the application.
On this page, participant can view results (‘Results’ button) or perform
comparisons (enter name in text box and click ‘Perform Comparisons’ button).