

Comparing AHP and CBRanking: an Experiment

Anna Perini¹, Filippo Ricca², Angelo Susi¹ and Cinzia Bazzanella¹

¹Fondazione Bruno Kessler - IRST
Via Sommarive 18, I-38050, Trento, Italy

²CINI at DISI Laboratorio Iniziativa Software
Università di Genova
Viale Dodecaneso, 35, I-16146, Genova, Italy

susi@itc.it

October, 16th, New Delhi

A controlled experiment to compare two different tool supported requirements prioritisation techniques based on pair-wise comparisons

- ▶ AHP (supported by JAHP)
- ▶ CBRanking (supported by SCORE)

with the objective of understanding differences in *accuracy* from the point of view of the decision maker

An accurate prioritization approach is one that produces a priority order which reflects the decision maker opinion

- User Preference: binary $\{-1, 1\}$ (CBRanking), ..., in a range $\{1, 3, 5, 7, 9\}$ (AHP), ...
- Preference to be acquired
- Symmetric Preference

Requirement B	
Descr.
Cost	50
Risk	high
Value	medium

Requirement A	
Descr.
Cost	100
Value	low

	A	B	C	D	E	F	G
A							
B	v						
C	v	?					
D	?	v	v				
E	v	?	?	?			
F	?	?	?	?	v		
G	v	?	v	?	?	?	

$N(N-1)/2$
comparisons

- JAHP*: it implements the *pair-wise comparison* approach based on AHP method (web-based)

JAHP

Java Analytic Hierarchy Process



Requirement A	Back	Requirement B
R3	Identifier	R6
Search a compilation in the compilation repository, by compilation name	Description	Single song download

How important is Requirement A compared to Requirement B?

A>>>>B
 A>>>B
 A>>B
 A>B
 A=B
 A<B
 A<<B
 A<<<B
 A<<<<B

Submit

Range of Values

Attention: after the pair elicitation, it is not possible to change the preference.

* <http://www.di.unioi.it/~morge/software/JAHP.htm>

- SCORE implements the pair-wise comparison based on CBRanking method (web-based)

SCORE - Single User

Supporting **C**ase-Based **O**riented **R**ank **E**licitation

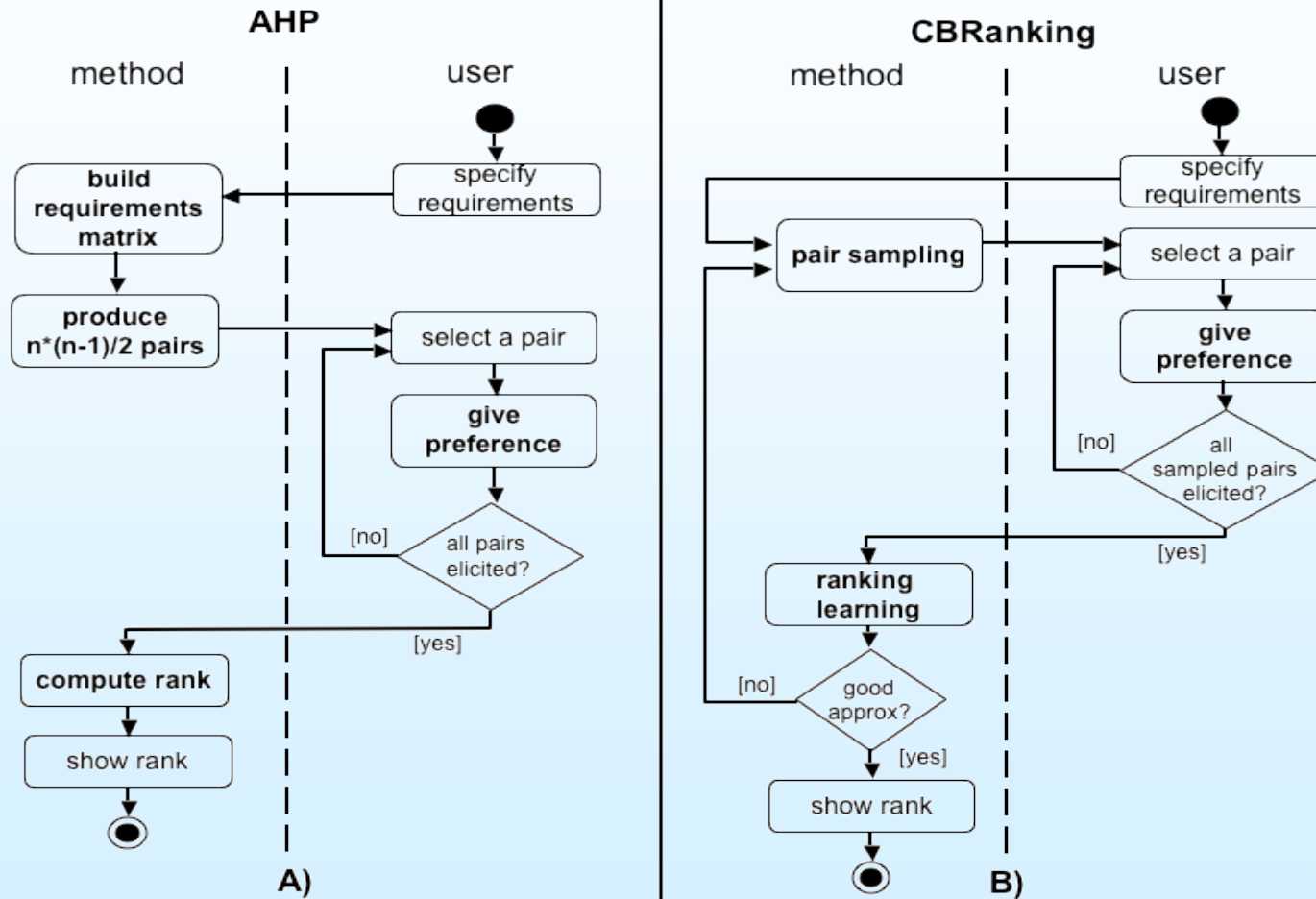


Requirement A	Back	Requirement B
R7	Identifier	R13
Inserting and deleting of a song in/from a compilation	Description	List of the top 10 downloaded compilations
<input type="text" value="More important requirement"/>		<input type="text" value="More important requirement"/>

Attention: after the pair elicitation, it is not possible to change the preference.

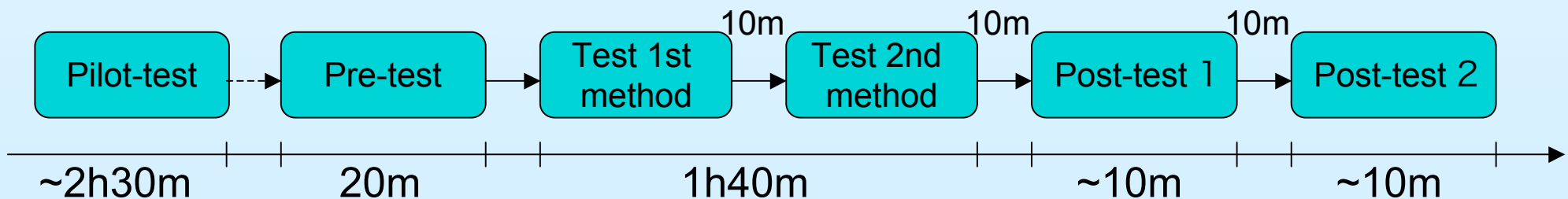
Binary values

The Prioritization Processes



Experiment

- 18 Subjects (PhD students, researchers)
- 20 Requirements from the Compilation Compiler Advisor (CoCoA) system (“Save a user defined compilation” or “Search a song in the songs repository, by title”)
- Criterion:
 - ▶ The Value criterion corresponds to how important and valuable the subject find the requirement
- To avoid the influence of the order in which the methodology can be exploited
 - ▶ Half of the set started with AHP, the other half with CBRanking



Null hypothesis:

- H_{0all} The **accuracy** is considered to be equal for both techniques, AHP and CBRanking by all subjects

The alternative hypothesis:

- H_{1all} The **accuracy** is not considered to be equal for both techniques, AHP and CBRanking by all subjects

Further null hypothesis:

- H_{0PHD} The **accuracy is considered to be equal** for both techniques, AHP and CBRanking **by PhD students**
- H_{0Res} The **accuracy is considered to be equal** for both techniques, AHP and CBRanking **by Researchers**

And related Alternative hypothesis

We also have detected on other hypothesis related to:

- **subjects with and without experience in requirements**
- **subjects with and without industrial experience as Analyst**

○ *Independent variables*: the techniques AHP and CBRanking

○ *Dependent variable*:

- ▶ *Accuracy*: a questionnaire on the “expected” accuracy in post-test 1 and a second question in the post-test 2 (few minutes after the experiment)

○ Pilot Test:

- ▶ *Two people* tested the whole experimental setting
- ▶ Evaluating the 20 CoCoA requirements with the two methods

○ Subjects were exposed to

- ▶ a questionnaire related to their knowledge about the prioritization methods exploited in the experiment
- ▶ a questionnaire related to their knowledge about CoCoA system features

- 2 and half hours experiment (4 groups in different days)

- AHP:
 - ▶ For 20 req => 190 comparisons (exhaustive)

- CBRanking:
 - ▶ For 20 req => ~55 comparisons (28% of the total pairs)

○ Post-test 1

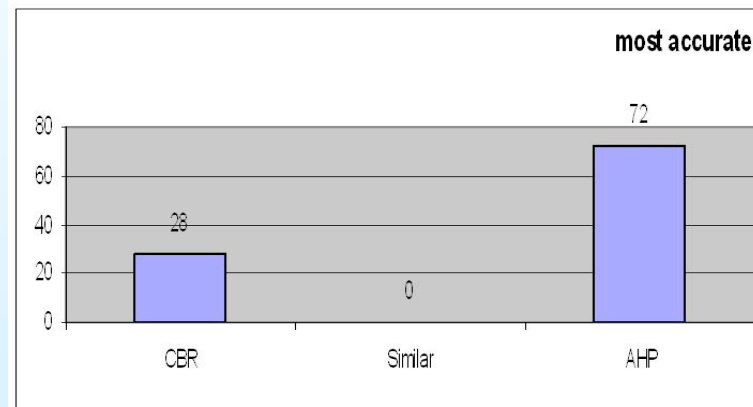
- ▶ Questions related to the “expected” accuracy (using 3 classes “more than”, “equal”, “less than”)

○ Post-test 2 (after few minutes)

- ▶ Users were shown the two lists of requirements produced by the two methods, one for AHP and one for CBRanking; (blind-test)
- ▶ Users were asked to select the rank that best fit their ideal rank

○ Questionnaire in post-test 1:

- ▶ 5 found CBRanking most accurate than AHP, nobody found them equally accurate 13 stated that AHP was more accurate (*not statistically significant: p -value=0.059*)
- ▶ The null H_{0all} cannot be rejected (H_{0all} The accuracy is equal for both techniques, AHP and CBRanking)

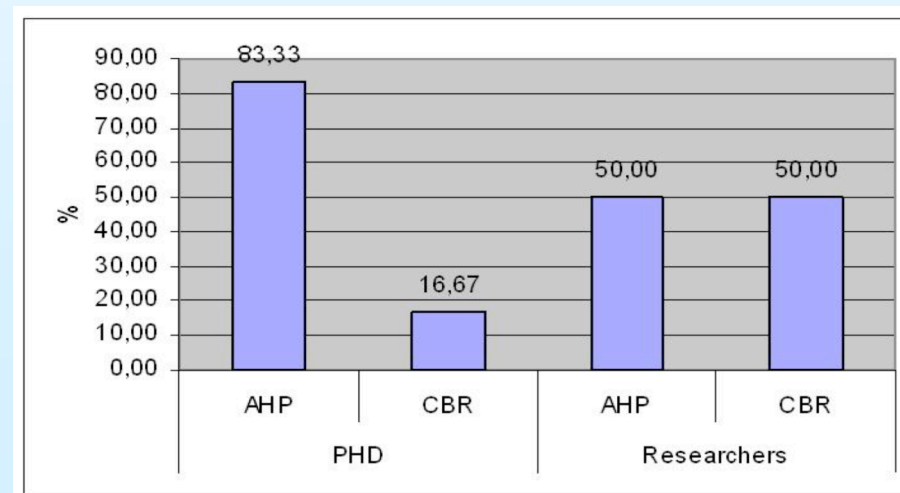


○ Questionnaire in post-test 2, after few minutes, 100% chosen the rank produced by AHP

- ▶ The null H_{0all} can be rejected (p -value=2.209e-05)

○ Questionnaire in post-test 1:

- ▶ Among the 12 PhD students: 2 found CBRanking most accurate than AHP, nobody found them equally accurate 10 stated that AHP was more accurate (*statistically significant: p-value=0.02*)
- ▶ Among the 6 researchers: 3 found CBRanking most accurate than AHP, nobody found them equally accurate 3 stated that AHP was more accurate (*not statistically significant*)
- ▶ H_{0PhD} can be rejected while the The null H_{0Res} cannot be rejected



- *Internal validity* Threats can be due to the:
 - ▶ fatigue effect. breaks during the experiment
 - ▶ learning effect. Measures of the interval of time for the elicitation seems to be constant (not influenced by the order of the methods)
- *Construct validity*
 - ▶ subjects not evaluated on their performance in the experiment
 - ▶ subject were not aware of the experimental hypothesis
- *External validity* Threats are always present when experimenting with students and researchers/programmers. Since the selected subjects represent a population with good knowledge on requirements, experience (78% of the subjects declared to have worked as programmer/analyst in the industry), and trained on prioritization methods we expect similar results for industrial developers

- We couldn't determine which technique has the *expected "highest accuracy"* except for the PhD students category (we can't statistically reject the null hypothesis) while, after the post-test 2, the null hypothesis for H_{0all} *could be rejected*
- How closed are the rankings produced by the two techniques? We measured the agreement among the rankings produced by each subject with the two methods and we found that the two rankings are generally very closed to each other

- Experiment in a real environmental setting where:
 - ▶ The number of requirements is higher
 - ▶ The requirements description more complex

- Experiment with AHP adopting the stopping rule

Thank you !