

# A Model-based Heuristic Evaluation Method of Exploratory Search

Emilie Palagi  
Inria, Université Côte d'Azur  
Sophia Antipolis, France  
emilie.palagi@inria.fr

Fabien Gandon, Alain Giboin  
Inria, Université Côte d'Azur  
Sophia Antipolis, France  
first-name.name@inria.fr

Raphaël Troncy  
EURECOM  
Sophia Antipolis, France  
raphael.troncy@eurecom.fr

**A shortcoming of current evaluation methods of exploratory search engines is that they cannot be used to determine whether users' behaviours and exploratory search tasks are usefully supported. One main reason is that these methods rely on a loosely defined model of the exploratory search process. The aim of this paper is to propose a new model-based heuristic evaluation method of these systems. We present the approach used to design the heuristics of exploratory search and a checklist form for using them. The evaluation of the heuristics and its procedure demonstrates that they significantly help the users achieve an exploratory search system assessment.**

*Exploratory Search Tasks, Heuristics, Evaluation methodology, Model-based method.*

## 1. INTRODUCTION

Exploratory search (ES) is a particular information seeking activity in terms of “problem context and/or strategies employed” (White & Roth, 2009). In this paper we will use White’s (2016) definition: “[the term] *exploratory search can be used to describe both an information-seeking problem context that is open-ended, persistent, and multifaceted, and an information seeking process that is opportunistic, iterative, and multi-tactical. [...] Although almost all searches are in some way exploratory, it is not only the act of exploration that makes a search exploratory ; the search must also include complex cognitive activities associated with knowledge acquisition and the development of cognitive skills*”.

Evaluating ES systems is still an open issue. A main point is the capacity of the evaluation methods to effectively assess whether users' ES behaviours and tasks are actually supported by the ES systems. One of the reasons is that these methods rely on a model of ES which is still loosely defined, or at least on a definition which is not yet clear and stable. The few existing evaluations based explicitly on a model proceed at a too low level avoiding the analysis of a user's exploration process in its entirety (Bozzon, Brambilla, Ceri, & Mazza, 2013; Wilson, schraefel, & White, 2009). Thus, the evaluation methods of ES are still incomplete as they are not fully based on a suitable ES process model.

Our main goal is to design a set of user-centered methods based on a suitable model of ES which provide a better understanding of the user, such as her needs or her behaviors when performing an ES. The method we present here is an inspection method in line with Nielsen's heuristic evaluation (Nielsen, 1994); it allows to evaluate ES systems

without users. The method is based on a model of ES emphasizing the *transitions* between the ES steps, so-called ES “*features*” hereafter, and aim to verify if the evaluated system supports ES behaviours. The method consists of a set of heuristics based on this model, and, of their associated procedure of use (including the heuristics presented in a checklist format).

## 2 THE MODEL'S FEATURES AND TRANSITIONS

The ten features of the model (from A to J) express typical ES *behaviours* such as having an evolving information need or a serendipitous attitude (Palagi, Gandon, Giboin, & Troncy, 2017): A. Define the search space; B. (Re)Formulate the query; C. Gather information; D. Put some information aside; E. Pinpoint result(s); F. Change goal(s); G. Proceed backward/forward; H. Browse results; I. Analyse results; J. Stop the search session.

Table 1 presents a non-exhaustive list of possible transitions between these 10 model's features. The transitions were elicited by comparing the features of the model with actual behaviours of three information-seekers performing an ES task. Two transitions were not observed in the records analysis but inferred from basic behaviours occurring commonly in an ES task. Users can do in their ES process the two following transitions:

**G → F:** When the user does backward or forward steps, she can have an idea and change her goal(s) of search. The ES process is evolving, and in fact, a user should be able to change her goal any time in the ES session.

**I → G:** When the user analyses the relevance of a result, she also analyses the pathway that took her there. If she is not satisfied, she can do backward

or forward steps to restore a satisfactory state. This situation is not specific to ES and can occur in all type of information seeking. Indeed, a user can analyse an irrelevant result and come back to the result list for example.

One will notice that a search session always starts with **A**. *Define the search space*, and ends with **J**. *Stop the search session*.

**Table 1:** Non exhaustive list of possible transitions in an exploratory search session. The transitions with asterisks were not observed but inferred.

Previous Features	Features	Next features
NA	A	B;J
A;F	B	G;H;I;J
D;E;I	C	D;E;F;G;H;J
E;I	D	C;F;G;J
G;H;I	E	C;D; F;G;J
C;D;E;G*;H;I	F	B;H;I;J
B;D;E;H;I*	G	E;F*;H;I;J
B;F;G;I	H	E;F;G;I;J
B;F;G;H	I	C;D; E;F;G*;H;J
All	J	NA

### 3. HEURISTICS OF EXPLORATORY SEARCH

Heuristic evaluation helps experts identify potential issues in the system's interface. We propose a list of ES heuristics (see details in Table 2) that present interfaces or interactions recommendations that help and facilitate users to carry out their exploration.

These heuristics are classified according to user interface elements: the home page, the list of results (or data), elements' description, the browsing history (including bread crumb and other similar features), and *any screen*. This classification helps the expert and makes the evaluation easier. Indeed, the expert can focus on a specified (element of the) interface and then split the evaluation for a better attention and a lower cognitive load.

#### 3.1 Heuristics' design procedure

The heuristics of ES are based on the model's features and transitions. However, in the design process of the heuristics, we realised that one heuristic is not necessarily linked to only one feature or to only one transition. So we had to cluster features and transitions. First of all, we listed the different ways to facilitate the transitions. For example, in Table 2, transition F (*Change goal(s)*) → B (*(Re)Formulate the query*) can be

realised in different ways and implies different needs in terms of interface elements: the user must be able to change easily her search at any time during her search session. This includes different situations, and that is why two heuristics are proposed:

- The search bar or the bookmarks are accessible anytime during the search session.
- The system should enable starting to start a new search from the search bar but also from an element of the interface (e.g. a result, a world/element on the result description...).

After that, we cluster the transitions corresponding to the same action with the same interface elements. For example, we cannot cluster the previous heuristics, but the transitions E (*Pinpoint result*) → D (*Put some information aside*) and I (*Analyse results*) → D can both be facilitated by using bookmarks.

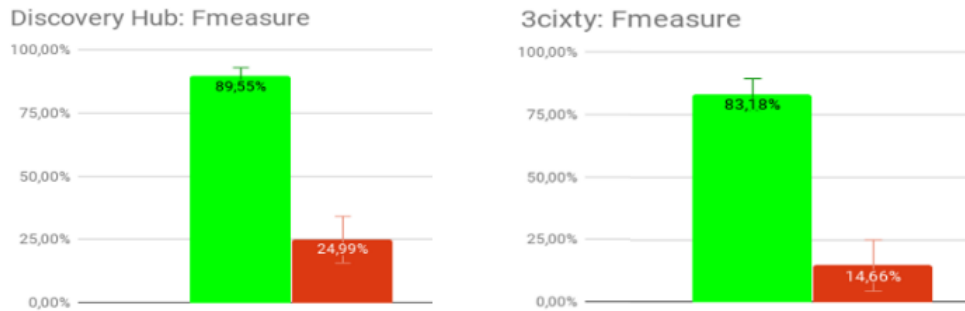
Note that C, D, E, G, H, I → F does not have linked heuristics because there are no interface elements that facilitate the transition to F. Indeed, this feature cannot be translated by a physical action: it occurs when the user decides to change her search goal.

Heuristics for facilitating the transitions between features are not sufficient to support the whole process of ES. Indeed, some model's features need to have their own heuristics. For example, feature A (*Define the search space*) implies a description of the system and tools to help the user to (1) understand what kind of information she will find by using the system, and (2) define her search space (e.g. with query auto-completion feature, suggestions...). The heuristic corresponding to the transition A → B (*(Re)Formulate the query*) is more focused on how to facilitate the query (re)formulation after learning more about the system at the beginning of the search session. That is why the heuristics that facilitate some model's features are also necessary, and we added heuristics that support these features. Note that heuristics "A & B" and "E & I" are also clustered, sometimes more than once.

For an easy use, the heuristics are classified into different screens. To this end, sometimes, we need to split some heuristics in two, e.g. the previous E, I → D:

**List of result/data:** The system should allow the user to bookmark elements from the result list.

**Elements' description:** The system should allow the user to bookmark elements from the elements' description.



**Figure 1:** F-measure of the tests on Discovery Hub and 3cixty. The left bar refers to Group A (with the evaluation checklist) and the right one to Group B (without the evaluation checklist).

The evaluator is able to use these classified heuristics without understanding or knowing the model of ES process or the transition to evaluate the ES system. She only uses the heuristics and verifies if the system satisfies all of them.

All the heuristics provide elements that support at least one feature or transition. Thus, if a system satisfies all the heuristics, it supports the ES process and behaviour. This evaluation of ES process should be performed by persons familiar with the ES process (e.g. ES systems designer, or UX designer). The level of knowledge on ES can vary but a certain level of experience is required. The evaluators will use their own experience and the heuristics to analyse and improve their search system.

### 3.2 Heuristics' use: an evaluation checklist

For an easy use of the heuristics of ES, we proposed to present them to the evaluators in a form format. Indeed, first tests of the heuristics showed that they are more understandable when phrased in an interrogative form. The evaluation checklist is at: <http://bit.ly/evaluation-checklist>

## 4. EVALUATION OF THE HEURISTICS OF EXPLORATORY SEARCH

In order to determine if an ES system supports users' ES behaviours, our heuristics intend to identify system's features, interface elements or possible actions that are required for the proper achievement of the ES task. We assume that an evaluation of the system with our heuristics of ES allows a better identification of these elements. We want to evaluate the following hypotheses derived from our research questions:

(H1) users following the evaluation checklist identify more provided or missing features than users not following the evaluation checklist

(H2) users following the evaluation checklist identify more systematically than users not following the evaluation checklist

### 4.1 Methodology and protocol

Twenty persons participated to the evaluation and all of them are computer scientists. They did not know any of the evaluated systems before. Ten participants evaluated Discovery Hub<sup>1</sup>: five with the heuristics (Group A) and five without the heuristics (Group B). The ten other participants evaluated 3cixty<sup>2</sup>: five with the heuristics (Group A) and five without the heuristics (Group B). In the users tests we proposed to Group A to use the heuristics in their form format mentioned in Section 3.2. We also provide Group B paper and a pen. Group A and B first familiarized themselves with their respective system through a scenario-based demo. Then Group A (evaluation with the checklist) evaluated the system by following the evaluation check list, whereas Group B replayed the scenario-based demo of the familiarization phase and formulated all negative and positive aspects that can facilitate or compromise the exploration.

### 4.2 Metrics

We measured the effectiveness of our heuristic method in terms of precision, recall and F-measure of the participants' answers:

**Group A:** each time a participant has to check "Yes" or "No" in a multiple choice list.

**Group B:** each time a participant's comment refers to an answer to the evaluation form.

We created a gold standard by asking two usability experts, who possess a good knowledge of ES process, to evaluate independently the two systems with the evaluation checklist. After a discussion about the understandability of the questions, the two experts reach a perfect consensus.

### 4.4 Results and conclusion

Figure 1 shows the F-measures for the two experiments with the two ES systems: for (H1)

<sup>1</sup> <http://discoveryhub.co/#>

<sup>2</sup> <https://nice.3cixty.com/>

considering the F-measure of 83.18% vs 14.66% and 89.55% vs 24.99% we can say that the users with the evaluation checklist identify more provided or missing system's features than users not following the evaluation checklist. For (H2), with a standard deviation of 3.35 vs 10.32 and 3.45 vs 9.26, we can say that users following the evaluation checklist identified features more systematically than users not following the evaluation checklist.

This evaluation demonstrates that the method significantly help the evaluators identify more and more systematically the presence or the absence of elements or possible actions required to effectively support ES behaviours than without the heuristics (Mann-Whitney test,  $p < 0.02$ ). Our experiment also shows that a more complete evaluation of these systems requires a combined evaluation with well-known usability and user experience heuristics such as Nielsen and Bastien's and Scapin's ones (Bastien & Scapin, 1993; Nielsen, 1994).

## 5. ACKNOWLEDGMENT

This work was partly funded by the French Government (National Research Agency, ANR) through the "Investments for the Future" Program reference #ANR-11-LABX-0031-01.

**Table 2:** Model-based Exploratory Search Heuristics. (Heuristics highlighted in bold italic are related to transitions between features.)

Home page
<ul style="list-style-type: none"> <li>From the very start, the user should understand what kind of information she will find by using the system (the explored field, the mediatype...). She may know what kind of query or search she can do. (A)</li> <li>The system use tools to help the user to define her search space (e.g. query auto-completion feature, suggestions, automatic correction or filtering tools, etc.). (A &amp; B)</li> <li><b>The search bar (or bookmarks, filtering tools, etc.) and the system's description are always nearby. (Transition A → B)</b></li> </ul>
List of results / data
<ul style="list-style-type: none"> <li><b>The system should present the results list after a query (re)formulation and a change of goal(s). (Transitions B, F → H)</b></li> <li>The system should associate to the title of the result a picture and, if it is possible, a result snippet. The idea is helping the user to understand and evaluate the results list faster. (I)</li> <li>The system should present the result/data in a relevant way according the type of the result/data and the main activity (list, graph, carousel, parallel interfaces...). (I)</li> <li><b>The system should allow the user to bookmark elements from the result list. (Transitions E, I → D)</b></li> <li><b>The system should show in the results list the way to access to the elements' description (e.g. more details button). (Transitions H, I → E)</b></li> <li><b>The system should first display the most relevant</b></li> </ul>

<p><b>results (e.g. a top selection list, the similarity or even the distance between the query and the result, etc.). (Transition B → I)</b></p> <ul style="list-style-type: none"> <li><b>The system should make available filters features that help the user to understand and analyse the results' list. These filters can be proposed by the system and/or chosen by the user (e.g. common facets). (Transition F → I)</b></li> <li><b>The system should explain why a result is proposed from the results' list. This explanation should help the user to analyse its relevance. (Transition H → I)</b></li> </ul>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Elements' description (including results' details and query description – if there is one)
<ul style="list-style-type: none"> <li>If there is query formulation, the system should provide a description of the query. Sometimes the user may explore unknown subject and need to learn more about it. (E &amp; I)</li> <li>The system should give enough information to provide a better understanding of the result. Every result should propose a description (text, picture(s), video, maps, sources...), the associated tags, characteristics, links, etc. These information enable the user to analyse the relevance of the result. The user mustn't need to go elsewhere (another search engine or website) to search the explanation. (E &amp; I)</li> <li>The system should explain why the result or element is proposed (why the user would find it relevant). Indeed, the link between two elements should be described (e.g. the link between the query and the result: the characteristics in common, a chosen filter...). (E &amp; I)</li> <li>The system should offer comparison features between at least two elements (including the query, different results...). This point is really important, especially if the user is often led to compare several element of information in her goal task or exploration. (E &amp; I)</li> <li><b>The system should allow the user to bookmark elements from the elements' description. (Transitions E, I → D)</b></li> </ul>

Browsing history / Breadcrumb
<ul style="list-style-type: none"> <li><b>In the browsing history or the bread crumb, the system should show the elements' descriptions the user already read. (Transitions G → E, and G → I)</b></li> <li><b>In the browsing history or the bread crumb, the system should show where the user was browsing the results' list. (Transitions G → H, and G → I)</b></li> </ul>

Any screen
<ul style="list-style-type: none"> <li><b>The search bar or the bookmarks are accessible anytime during the search session. (Transition F → B)</b></li> <li><b>The system should allow to launch a new search from the search bar but also from an element of the interface (e.g. a result, a world/element on the result description, etc.). (Transition F → B)</b></li> <li><b>The system should propose an easily (or always) accessible browsing history breadcrumb and back/next buttons. (Transitions B, D, E, H, I → G)</b></li> <li>The system should help the user to keep in mind her goals of search (e.g. the user can see, at any time, the query and the exploratory search pathway throughout bread crumb or browsing history). (G)</li> </ul>

## 6. REFERENCES

- Bastien, J. M. C., & Scapin, D. L. (1993). *Ergonomic criteria for the evaluation of human-computer interfaces* (No. RT-0156) (p. 79). Inria. Retrieved from <https://hal.inria.fr/inria-00070012>
- Bozzon, A., Brambilla, M., Ceri, S., & Mazza, D. (2013). Exploratory search framework for Web data sources. *The VLDB Journal*, 22(5), 641–663.
- Nielsen, J. (1994). Usability inspection methods. In *Conference companion on Human factors in computing systems* (pp. 413–414). ACM.
- Palagi, E., Gandon, F., Giboin, A., & Troncy, R. (2017). A Survey of Definitions and Models of Exploratory Search. In *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics* (pp. 3–8). ACM.
- White, R. W., & Roth, R. A. (2009). Exploratory Search: Beyond the Query-Response Paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1), 1–98.
- Wilson, M. L., schraefel, m. ., & White, R. (2009). Evaluating advanced search interfaces using established information-seeking models. *Journal of the American Society for Information Science and Technology*, 60(7), 1407–1422.