Towards Exploratory Search Mashups based on Strategic Knowledge

Oliver Mroß, Carsten Radeck and Klaus Meißner

Faculty of Computer Science
Technische Universität Dresden
Dresden, Germany
Email: {oliver.mross, carsten.radeck, klaus.meissner}@tu-dresden.de

***Abstract*—Exploratory search is a complex, long-lasting and highly iterative process. Users may have only a vague or an open-ended information need that is likely to change during the search process. Besides the insufficient domain knowledge, most users lack experience in efficient information search. Hence, they have to be guided during the search process by strategic recommendations. In this work, we present an overview of our strategy-oriented search platform that derives appropriate composite web applications from recommended and preferred search strategies with respect to the current user and search context. Moreover, we give a glimpse into our meta-model of search strategies, which represent best practices on how to solve search problems and are described using hierarchical task models.**

Keywords–Mashup; End-User Development; Exploratory Search; Search Strategy; Strategy Recommendation.

I. INTRODUCTION

Considering today's vast amount of digital information and it's increasing availability, information search is an omnipresent human activity [1]. Traditional Web search engines follow the same retrieval paradigm "query and response". Thereby, the user's information need is represented as a keyword-based query that is processed by the search engine. Corresponding results are returned as a ranked list of entities containing additional meta-data, e. g., title, content fragments and the data source. However, this lookup-oriented information seeking model does not fully represent the human search behavior in real life scenarios. Users may have a vague or open-ended information need, which furthermore is likely to change during the search process. Satisfying his/her information need is additionally hindered if the user's research or domain expertise are insufficient [2]. Consequently, *exploratory search* characterizes information seeking as a highly iterative, often long-lasting and complex process [3]. As pointed out by [4], assisting users during the search process is crucial. We argue the searcher has to be guided by *search strategies*, which are consolidated, verified and composable best practices on how to solve search problems. To this end, search strategies are recommended by the search environment and serve as foundation for generating dashboard-like search applications with respect to the current context (i. e., user's profile, search history, research task description and problem domain).

A promising approach for generating context-aware search dashboards are *composite web applications* (CWA) consisting of loosely coupled *mashup components*. The latter encapsulate arbitrary web-services and resources like domain-specific business logic as well as widgets. Moreover, each component is characterized by a semantically enriched interface providing information according to their data provision and capabilities [5]. It has the advantage that a component's interface as well as their capabilities can be matched more precisely on a semantic level with an abstract application description including required capabilities, e. g., a business process model, and allows to generate corresponding mashups.

In our vision, a *search strategy* is a formal description of the planned use of information gathering activities leading effectively and efficiently to relevant search results. An information gathering activity is performed by the user, e. g., providing an author's name, or by the search system, for instance, presenting the author's books or papers relevant to the user's information need. A search strategy is effective because the resulting entities (documents, terms, domains etc.) are highly relevant compared to the user's information need. It is efficient because it reduces the user's cognitive load by providing a predefined order of search activities and it reduces the amount of time in finding relevant information.

The idea is to utilize strategy models for assisting users throughout search processes. Thereby, strategies are recommended context-sensitively, giving advice on efficient search activities. Additionally, they serve as a foundation to derive adequate CWA, taking advantage of composition knowledge and the semantic description of component interfaces.

Typically, strategies are associated to a critical situation and should result in valuable information related to the initial search problem. Consider the following two strategy examples:

1) "Define the information need more precisely by selecting refined concepts of domain X using archive services $S_1, S_2, \dots, S_n$ if precision level of current query is low."
2) "Reformulate a query more precisely by searching for concepts in top-10 relevant papers and slides of domain experts $E_1, E_2, \dots, E_m$ if the user's search experience is low and the current result list is almost empty."

Such search strategies are defined explicitly by search experts or can be derived semi-automatically from usage and feedback data of experienced searchers.

Our vision of a guided exploratory search experience comes with various requirements. First, a sufficiently expressive *search strategy meta-model* for describing stepwise information gathering processes usable in a multitude of scenarios is needed. It should address the following aspects: (a) Specifying *information request and provision activities*, concrete domain concepts or placeholders, each referencing a domain concept, a document type or a human informant (e. g., the data analysis expert) are needed. (b) *Context criteria* to define valid usage situations of search strategies are required considering the temporal availability and validity of multidimensional context data. Secondly, the meta-model concepts are the basis for *strategy recommendation* and *mashup generation*, i. e., the concepts should largely support the component selection and

 ISBN: 978-1-61208-563-0

composition process to reduce the configuration effort from the end-user perspective. Considering that there is a semantic gap between the abstract process-oriented strategy meta-model and the fine-grained, function-oriented model of mashup components, there is a need for an *efficient mapping algorithm* that should be executed transparently during the search process.

Regarding the previous requirements, the contributions of this work are the following: First, we propose a meta-model for search strategies that allows to specify arbitrarily complex information gathering activities and which features semantic annotations. Second, we present a novel reference architecture of an exploratory search platform. It supports unexperienced users in highly iterative information search with help of guidance mechanisms leveraging the strategy meta-model. Especially users are assisted by strategy recommendation and provision of adequate search mashups. The remainder of this paper is structured as follows. In Section II, we discuss related work. Then, Section III presents our search platform based on mashup concepts. Finally, Section IV concludes the paper.

II. Related Work

Approaches like query suggestion [4], [6] and facet recommendation [7], [8] try to compensate the limited domain and search expertise of users. Such techniques are assisting users in specifying an information need more precisely. However, users often have complex information needs requiring them to perform multiple search steps [9]. Such approaches fall short of expectations in supporting users on a strategic level.

Kangasrääsiö et al. [10] propose a search front-end showing estimates of the user's search action effects allowing the user to anticipate consequences and to direct his/her exploratory search. Thus, the user actively influences the information search. Musetti et al. [11] use topological knowledge patterns extracted from Wikipedia to deliver relevant and filtered search results. Each result is represented semantically and visualized based on concepts provided by DBpedia. Complementing meta-data are retrieved from additional sources such as Twitter. Anyway, for users without extensive domain knowledge, it is challenging to navigate to relevant information. We follow the motive of [10], but we argue that users without guidance explore information rather in a trial and error manner. Thus, information search still is time consuming and cumbersome. Compared to our solution, both approaches lack concepts for active user guidance.

Bates describes four levels of search activities [12], whereby higher levels build upon lower ones. *Moves* are the basic unit of her model. *Tactics* are composed of moves in order to improve search activities. *Stratagems* are larger complexes of several moves or tactics and they are typically domain-specific. A *strategy* can be considered as a plan for a whole search process, incorporating all other types of activities. Bates emphasizes the importance of supporting users on strategic levels, which is one of the main goals of our approach. In contrast to [12], we formally describe search activities of all levels. Belkin characterize *information seeking strategies* (ISS) [13] according to the dimensions: *method* of interaction, *goal* of interaction, *mode* of retrieval, and resource considered. They identify 16 relevant combinations and thus strategies. Our model is partly inspired by this work, as we describe user's situation and search activities. However, it lacks formalism and a detailed search process description. In [14] Belkin et al. propose script-based user guidance, whereby scripts represent effective interaction sequences for ISS. Such scripts serve as prototypical dialogs between system and user and can be combined to more complex ones. In the case of the *MERIT* platform [14], scripts guide users during the search process. Besides initially provided scripts, case-based reasoning is applied to derive scripts. Similarly, our approach allows to model circumstances when a strategy can be applied. However, we take the information need and user group into consideration. In addition, we describe search processes, yet in a semantically enriched and user-oriented way.

According to Sutcliffe and Ennis [15], strategies represent information searching skills and are determined by the type of information need. The latter is categorized according to aspects like the knowledge a user has about the information, whether the need is rather fix or likely to change, and if the target is precisely known or rather general. They propose a search process model that features strategy selection rules, which govern behavior within the process model. Such rules differ in their preconditions, incorporating information need types and other context parameters, as well as action clauses that, e. g., alter the query and invoke an action. Strategy rules are used to determine suitable strategies with respect to the information need type and current search process. Our approach is influenced by the work of Sutcliff and Ennis. For instance, we adopt their concepts for describing information needs and context-sensitively selecting suitable strategies in a rule-based manner. However, they provide no model of strategic process knowledge as we require it.

The *FIRE* system presented in [16] offers strategic help to users in form of suggestions, which partially correspond to Bates' classification. They apply reasoning on user actions and the search context to derive applicable suggestions. Selecting suggestions is based on rules describing necessary context conditions and the consequences as actions. Our approach not only allows to provide suggestions in context of a predefined search application, but also uses strategic knowledge to derive suitable applications, which is out of scope of FIRE. In addition, our strategy model can describe whole search processes and strategies can be composed. Kriewel et al. present *DAFFODIL* [9] that uses case-based reasoning techniques for determining appropriate strategic suggestions considering the current user's context based on the tactics and stratagems according to [12]. Tacke and Kriewel [2] extend DAFFODIL by providing tools enabling guided information search. They differentiate between macro and micro-level guidance. The former support unexperienced users in specifying his/her information need and explaining steps of a complex search task. A disadvantage is a fixed set of generic features and tools for the different search process phases. Domain-specific visualizations and user preferences are considered in a very limited fashion only. Our platform strives to provide strategic knowledge and a construction kit to reflect it in suitable CWA.

III. Strategy-oriented Search Platform

In this section, first we present an architectural overview of our CWA platform for exploratory search and its strategy-based functionalities. Afterwards, details on our proposed strategy model are discussed. Finally, we describe some of the novel features, which utilize strategy models.

A. Architectural Overview

We claim that assistance mechanisms based on formalized strategy descriptions, for instance, context-aware strategy recommendations, generation of appropriate applications and query suggestions, are supporting and substantially simplifying the user's overall search process. As result of our investigation of related solutions (see Section II) and to the best of our knowledge, there is a need for a novel information search platform providing such assistance, which we present afterwards.

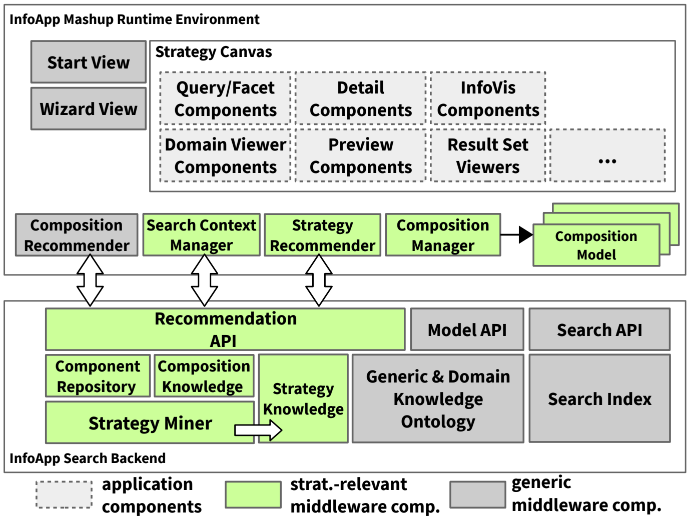

Figure 1. Architectural overview of our platform for exploratory search.

As illustrated in Figure 1, the search platform's front-end is based on a mashup runtime environment. The *composition manager* implements the life-cycle of CWA, which are represented by a *composition model*. Exploratory search mashups are rendered in the *strategy canvas* and build up on a set of components that provide capabilities to cover all phases of a search process, for instance, components for textual or graphical query construction, result lists and diagrams as well as charts for visual analytics. In order to provide several starting points into the research process, the front-end provides a *start view* and a *wizard*, c. f. Section III-C. The platform features a component-oriented recommender system (*composition recommender*) and assistance mechanisms for live development of CWA [5]. They allow to recommend, select and compose components as required. To this end, *composition knowledge*, that holds information about mashups and recurring composition patterns, and a *component repository*, that stores information about semantically annotated components, are utilized. Both are accessible via the *recommendation API*. In addition, the latter provides the following context-aware search-oriented recommendation functionalities: (a) suggest domain concepts and facets, (b) suggest query reformulations, (c) recommend related documents, domains and their inherent concepts, and (d) recommend search strategies each associated with at least one adequate CWA. These functions incorporate domain knowledge represented in *ontologies*. Furthermore, they pay attention to the current search context, which comprises a users' research task, queries, information need and skills. It is maintained by the *search context manager* and analyzed by the recommender in order to fulfill the above mentioned functionalities. *Strategy knowledge* serves as a further crucial data source for our recommendation and assistance features. Therein, formal models of search strategies according to our meta-model, see Section III-B, are stored and maintained. A *strategy miner* is responsible for semi-automatically detecting recurring work-flows in the *composition knowledge*. By comparing the current context with the purpose and use cases of a search strategy, the *strategy recommender* derives and presents suitable strategies, as detailed in Section III-C.

In order to answer user queries, search mashups leverage the *search API*, which grants access to our hybrid *search index*. The latter combines the efficiency of an inverted index together with the expressiveness of an ontology.

B. Strategy Model

As a prerequisite, we briefly outline our *user and search context model*. Therein, user profiles model skills that include search and domain expertise, and interests based on semantic concepts and quality levels. Users have certain *roles*, that additionally imply specific skills and can group users. Furthermore, the current search context describes the search task featuring a textual description, research goals (adopting [13]) associated with semantic domain concepts, a classification of the information need (based on [15]). Further, current research activities including a history of queries and gathered feedback of users in association to relevant documents, concepts or strategies is represented.

Search strategies describe effective, proven practices for fulfilling certain information needs, for instance, searching patents by navigating in a classification or by querying companies in a sector. According to our meta-model depicted in Figure 2, each *search strategy* is formally characterized by the following attributes.

- *core meta-data* like a name and description
- circumstances (*cases*) under which a strategy is useful. Such cases are basically tuples of *search task*, user *roles* and *rating*, reflecting the suitability of a strategy in a given context. To describe target groups of users for that a strategy is suitable, we utilize user *roles*, that group users with respect to their skills. Furthermore, a case is associated with a model of a CWA. There are two types of cases differing in their origin:
  - *reference purposes* are defined by search experts,
  - *community feedback* on the suitability of strategies.
- hierarchical task model (*search strategy* and subclasses) formally specifying a procedural description including
  - place holders carrying selection rules for dynamic expansion using other strategy models (see *isTemplate*),
  - composite and/or conditional activities (expressed in sequential or parallel order or as alternatives). Hierarchically defined *conditions* allow to further restrict when strategies are applicable. Therein, arbitrary context parameters are addressable using a selector language, like SPARQL property paths.
  - besides user actions, activities can model system actions and thus configure platform features similar to [15], like the recommender system and the component selection during CWA generation.

As can be seen, our model is influenced by cases and scripts [14], task models and capabilities [5]. With regard to Bates' categorization, our model covers all levels, i. e., it is capable of describing moves, tactics and arbitrarily complex stratagems and strategies.

C. Strategy-based Platform Features

As indicated in Figure 3, we distinguish two roles interacting with our platform. *Users* with little or no strategic

 ISBN: 978-1-61208-563-0

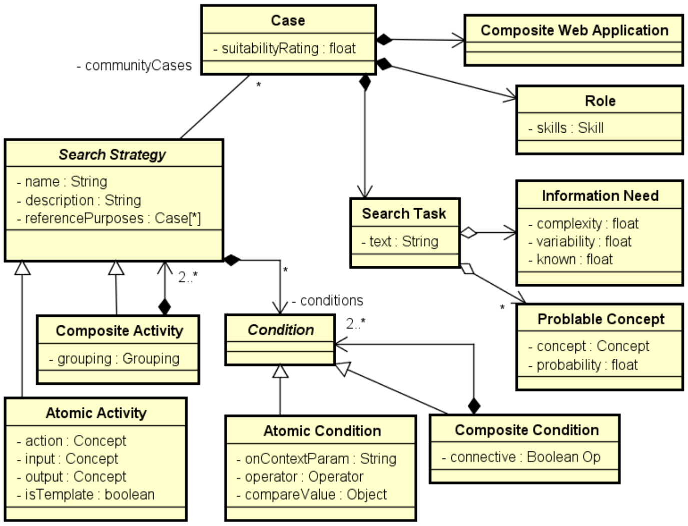

Figure 2. Overview of the search strategy meta-model.

expertise and with limited domain knowledge utilize our platform to fulfill their information need. *Search experts* are experienced information seekers with profound knowledge about efficient search strategies, which they apply as required. To this end, they create or modify CWA on demand. We assume, that experts are interested in sharing strategic knowledge by explicitly modeling their strategies and contributing them to the platform's strategy knowledge.

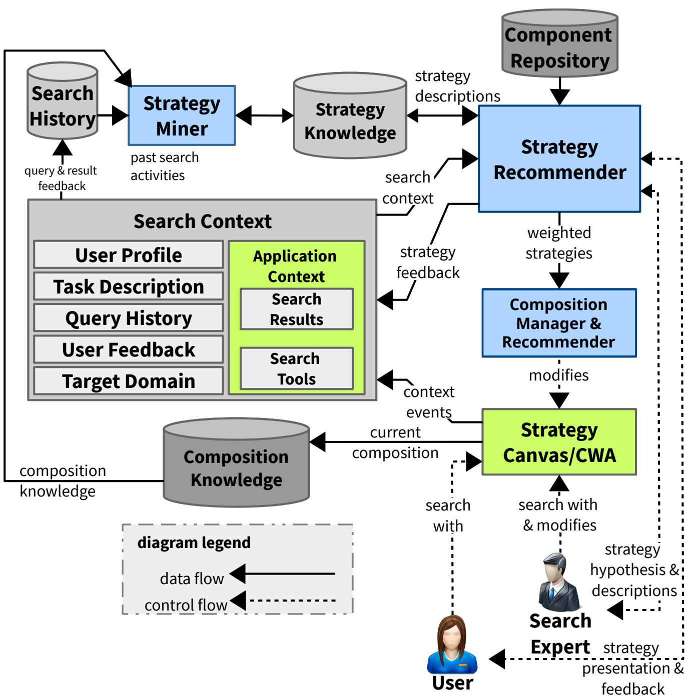

Figure 3. Strategy Recommendation Overview

In our approach the search platform provides following main features, which we explain in more detail afterwards:

Strategy assignment: Search strategies are assigned to matching CWA automatically by the *strategy miner* prior to the user's information search. Furthermore, the *search expert* can specify the association between a strategy and its representing application semi-automatically at runtime.

Strategy recommendation: Search strategies are recommended by the *strategy recommender* at the *beginning* or *during* the search process. For this, meta-data, e. g., the name and human readable description, are visualized by the *wizard* or as part of the *search canvas*.

Strategy usage: After strategies were recommended, the user activates the most appropriate strategy. At this point, the association between the selected strategy and its referenced CWA is resolved. Thereafter, we consider following integration cases: (a) Initial setup of the CWA, (b) extend or (c) replace the current composition.

Strategy feedback: From the user's perspective a strategy leads to more or less suitable search results. Reusing a valuable strategy or to filter out unsuitable ones the user can give feedback.

Next, we discuss these features considering the relations between actors and platform entities depicted in Figure 3.

Strategy assignment: We distinguish between *automatic* and *semi-automatic* strategy assignments. The latter is performed by *search experts*. In the first case, the *strategy miner* compares each registered strategy model with available CWAs managed by the *composition knowledge base* in association with historical context parameters provided by the *search history*, e. g., the research task or user profile. When comparing the strategy and CWA, the strategy's context condition are matched with context parameters associated to the current mashup. Moreover, the strategy's activities are compared with capability descriptions of each component referenced in the mashup's composition. Both values—the *context* and *activity* matching degree— result in the overall strategy-application similarity. After a strategy has been matched with available CWAs, the applications list is ranked with respect to the matching degree of every strategy-application pair. The top-*k* applications are associated with the current strategy using the *case* concept introduced in Section III-B and stored in the *strategy knowledge base*. In summary, each strategy is compared with available CWA and the best matching applications per strategy are assigned. The associated applications are considered as the strategy's manifestation on the application layer and are advertised during the user's search time.

In the case of *semi-automatic* strategy assignment, while the *search expert* is composing an information search mashup the *strategy miner* calculates probable strategies at runtime. At this, the expert's mashup as well as his/her search context are compared to each strategy model similarly to the algorithm described above. As result, probable strategies are calculated and visualized as *hypothetic candidates* to *search experts*. The latter evaluate strategy-application pairs and can modify strategy models and CWA, for instance, add domain and knowledge conditions as well as domain-specific mashup components. Thus, the validity and relevance of strategies is ensured by explicit feedback and the expertise of search professionals.

Strategy recommendation: Essentially, our platform provides two entry points into the search process: In the *start view* users can browse recommended strategies, which are filtered with respect to the user profile, or is guided by a *wizard*. The latter supports users in formulating research goals, captures relevant topics and assists users by recommending related search strategies. In addition, default search CWA for generic strategies can quickly be accessed. Initial strategy recommendations

 ISBN: 978-1-61208-563-0

take into account the user context with skills, roles and history, and the search context including task description, information need and queries. The *strategy recommender* derives suitable strategies leveraging semantic filtering techniques. For this, context conditions of every strategy are evaluated and matched with current context parameters. For example, the strategy "find chemical patents by formular" includes following conditions: $((domain \simeq \text{chemistry}) \vee (userrole \simeq \text{patent officer}))$. As fallback solutions there are general purpose strategies and corresponding CWA featuring generic search tools.

Further, strategies are recommended at *runtime*. To enable guidance throughout search processes, the current context is continuously monitored. Upon relevant context changes, e. g., modified query, selected target domains and facets etc., new loops of the recommendation procedure are triggered. Consider the following example: After the user has selected several documents and topics from the chemistry domain, the platform offers matching strategies. One of them—strategy "precise chemical search queries"—suggests to use a chemistry-specific query formulation tool based on chemical formulas. As soon as the user accepts, assigned CWAs are presented as sorted list in the strategy canvas. The applications order depends on the similarity of each strategy-application pair and on the feedback of other users or search experts. The *case's suitability rating* introduced in Section III-B represents both aspects.

Strategy usage: After recommendation, a user can choose from several strategies and at least a CWA per selected strategy is generated. At this point, we differentiate between following integration cases. At the *beginning* of the search process, the *strategy canvas* only includes generic search tools such as a query editor, a facet browser and a search results viewer. Per selected strategy the user can activate most appropriate mashups and decides whether the current composition will be extended or a new mashup is created in the *strategy canvas*. After the user has chosen an option, components and communication relations between them are integrated as defined in the composition model associated to the activated strategy. The integration process is performed and monitored by the *composition manager*.

Moreover, strategies are recommended continuously throughout the search process. Hence, while using an activated strategy a recommended one can be *merged* into the existing application context. A sample strategy is presented in Figure 4. Activities and conditions of the search strategy "SUPER" are shown as UML state diagram. Context conditions are visualized as transition guards. For instance, the main strategy "SUPER" is suggested when there are less or equal than five search results and when the current query is overspecified. The strategy's purpose is to support users in finding appropriate hypernyms. For this, the main strategy contains two activities. First activity "Select" results in a hypernym that is automatically set as the current query (second "Modify" activity). As discussed in Section III-B our meta-model supports to define template activities, which could be replaced by more specific variants at runtime. In Figure 4 the first activity is replaced with a domain-specific one (green colored), which is activated after the user has selected the chemistry domain. The new strategy allows retrieving chemical formulas from several sources, e. g., a query from search history or a webservice, and to get an appropriate hypernym from a chosen chemical formula. The selected hypernym is used to solve the problem of overspecified queries and to broaden the search results. Modifications on the strategy's activity layer are synchronized with the mashup's composition layer. Considering the sample strategy in Figure 4, generic composition fragments are replaced with domain-specific components.

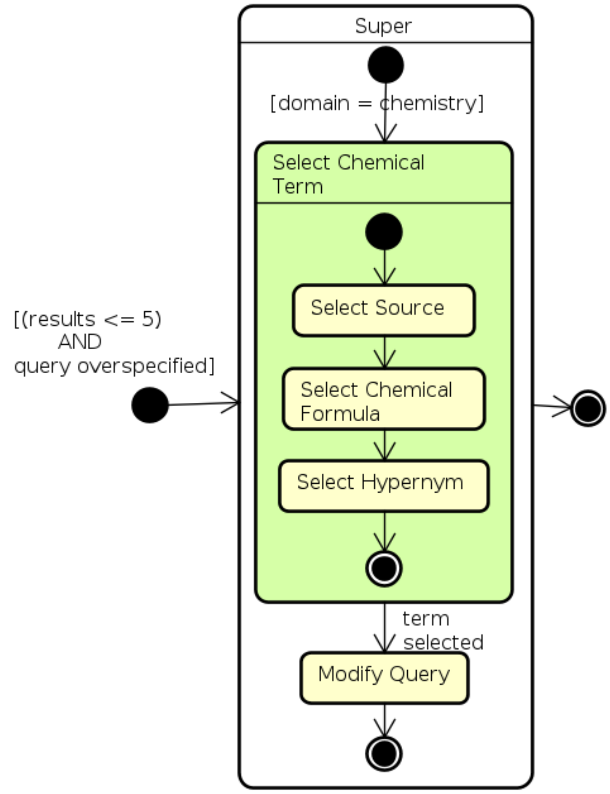

Figure 4. Sample strategy SUPER

Strategy feedback: At this point, we distinguish between *search expert* and *user* feedback on a strategy and its CWA. It is collected and managed by the *strategy recommender*. Feedback is created as a tuple of strategy, CWA, user and search context and represented as *case* stored by the *strategy knowledge base*. When more users give positive feedback according to the same case, the higher the corresponding strategy is weighted, i. e., its suitability rating increases. This in turn causes a higher ranking of the strategy during the recommendation process. Negatively rated strategies are degraded.

IV. CONCLUSIONS

In addition to the insufficient domain expertise, most users lack experience in efficient information search. Thus, there is a need for an intelligent search platform guiding the searcher by recommending appropriate search strategies. We support the user in his/her information seeking activities by continuously recommending collaboratively filtered search strategies depending on the current search context, so the user is able to design his/her CWA only by selecting a preferred strategy description. Further, search experts can teach valid search strategies to the platform. Users of the same community may profit from their expertise, because the platform is able to derive best matching compositions that the experts themselves can not anticipate during their search activities.

However, limitations of the presented approach are (a) the *cold start problem* and (b) the user's *cognitive overload* while using complex search strategies with deep activity hierarchies.

The first limitation is characterized by the necessity to provide predefined strategy descriptions and composition models. Hence, we introduced the *search expert* as actor with sufficient information seeking and programming experience who can explicitly specify strategy models. In order to reduce the inherent cognitive load and time effort, sophisticated strategy development tools that allow designing strategies comfortably are required. For this, a visual editor based on UML activity diagram notations could be used to generate reusable strategy models. Another approach would be to extract strategies from existing CWA automatically. For this, tracking features to capture the expert's input and interaction events as entities of the application context are required. In addition, sufficient analyses and aggregation mechanisms to recognize strategic search decisions from user behavioral patterns are required. Considering black-box components that have app-like granularity, this is not feasible due to missing interaction details. Thus, we decided to rely on (semi-)automatic assignment of adequate CWA to strategies, but this implies that there are always corresponding mashups available. We assume there are several predefined compositions models, which are developed by domain experts without programming experience using EUD-tools [5]. Our platform supports the mashup EUD, but currently lacks the strategy assignment features described above. In the near future, we plan to develop the *strategy miner* and its strategy assignment features as part of the existing *composition and component repository* web-service of the platform's back-end.

The second limitation could be solved using automatically generated tutorials, which give an overview of integrated mashup components and are guiding users while interacting with them. For instance, in the chemistry domain they describe how a chemical formula editor is used in combination with the facet browser and a graph-like molecule viewer of the same application. Based on the assumption that such tutorials are generated from strategy knowledge and additional component interface annotations (e. g., capability descriptions) this solution complicates component development.

Finally, we plan to evaluate our approach with the help of a user study.

Acknowledgements

This work is funded as result of the InfoApp-project (https://mmt.inf.tu-dresden.de/infoapp) by the European Union and the Free State of Saxony within the EFRE program.

References

[1] G. Marchionini and R. White, "Find what you need, understand what you find," International Journal of Human–Computer Interaction, vol. 23, no. 3, 2007, p. 205–237.

[2] A. Tacke and S. Kriewel, "Strategic search support on macro and micro level," Datenbank Spectrum, vol. 14, no. 1, 2014, p. 19–28.

[3] R. White and R. Roth, Exploratory Search:Beyond the Query-Response Paradigm. Morgan & Claypool, 2009.

[4] A. Hassan Awadallah, R. W. White, P. Pantel, S. T. Dumais, and Y.-M. Wang, "Supporting complex search tasks," in Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ser. CIKM '14. New York, NY, USA: ACM, 2014, p. 829–838.

[5] C. Radeck, G. Blichmann, and K. Meißner, "Estimating the functionality of mashup applications for assisted, capability-centered end user development," in Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016), 2016, pp. 109–120.

[6] D. Jiang, K. W.-T. Leung, L. Yang, and W. Ng, "Query suggestion with diversification and personalization," Knowledge-Based Systems, vol. 89, 2015, p. 553–568.

[7] M. Tvarožek and M. Bieliková, "Generating exploratory search interfaces for the semantic web," in Human-Computer Interaction. Springer, 2010, p. 175–186.

[8] T. Le, B. Vo, and T. H. Duong, "Personalized facets for semantic search using linked open data with social networks," in Innovations in Bio-Inspired Computing and Applications (IBICA), 2012 Third International Conference on. IEEE, 2012, pp. 312–317.

[9] S. Kriewel and N. Fuhr, "Evaluation of an adaptive search suggestion system," in European Conference on Information Retrieval. Springer, 2010, p. 544–555.

[10] A. Kangasrääsiö, D. Glowacka, and S. Kaski, "Improving controllability and predictability of interactive recommendation interfaces for exploratory search," in Proceedings of the 20th international conference on intelligent user interfaces. ACM, 2015, pp. 247–251.

[11] A. Musetti, A. G. Nuzzolese, F. Draicchio, V. Presutti, E. Blomqvist, A. Gangemi, and P. Ciancarini, "Aemoo: Exploratory search based on knowledge patterns over the semantic web," Semantic Web Challenge, vol. 136, 2012.

[12] M. J. Bates, "Where should the person stop and the information search interface start?" Information Processing & Management, vol. 26, no. 5, 1990, p. 575–591.

[13] N. Belkin, P. Marchetti, and C. Cool, "Braque: Design of an interface to support user interaction in information retrieval," Information Processing & Management, vol. 29, no. 3, 1993, p. 325–344.

[14] N. J. Belkin, C. Cool, A. Stein, and U. Thiel, "Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems," Expert Systems with Applications, vol. 9, no. 3, 1995, p. 379–395.

[15] A. Sutcliffe and M. Ennis, "Towards a cognitive theory of information retrieval," Interacting with Computers, vol. 10, no. 3, 1998, p. 321 – 351.

[16] G. Brajnik, S. Mizzaro, C. Tasso, and F. Venuti, "Strategic help in user interfaces for information retrieval," Journal of the Association for Information Science and Technology, vol. 53, no. 5, 2002, pp. 343–358.

[17] C. Radeck, A. Lorz, G. Blichmann, and K. Meißner, "Hybrid Recommendation of Composition Knowledge for End User Development of Mashups," in ICIW 2012, The Seventh International Conference on Internet and Web Applications and Services, 2012, pp. 30–33.