Breaking the Recursivity: Towards a Model to Analyse Expert Finders

Matthieu Vergne[1,2] and Angelo Susi[1]

[1] Center for Information and Communication Technology, FBK-ICT, Via Sommarive, 18 I-38123 Povo, Trento, Italy , {vergne,susi}@fbk.eu
[2] Doctoral School in Information and Communication Technology , Via Sommarive, 5 I-38123 Povo, Trento, Italy , matthieu.vergne@unitn.it,

2015-10-22

Abstract

Expert Finding (EF) techniques help in discovering people having relevant knowledge and skills. But for their validation, EF techniques usually rely on experts, meaning using another EF technique, generally not properly validated, and exploit them mainly for output validations, meaning only at late stages. We propose a model, which builds on literature in Psychology and practice, to identify generic concepts and relations in order to support the analysis and design of EF techniques, thus inferring potential improvements during early stages in an expert-free manner. Our contribution lies in the identification and review of relevant literature, building the conceptual model, and illustrating its use through an analysis of existing EF techniques. Although the model can be improved, we can already identify strengths and limitations in recent EF techniques, thus supporting the usefulness of a model-based analysis and design for EF techniques.

Keywords— Expert Finding, Concept Formalization, Model-driven Analysis, Design Support

1 Introduction

Expert finding (EF), also called expertise location or expert recommendation [9], aims at recommending *experts*, or at least the most knowledgeable or skilled people we find within a community of people, on a given domain. EF is broadly useful, because it allows to acquire knowledge and skills through hiring [8], to support decision making and solve problems [8, 9], help in requirements elicitation [11] or even to validate models and approaches in research (e.g. soundness, practicality). One performs EF by evaluating the expertise of performers within the available community, before to rank them or to select the ones to recommend. More precisely, you are an expert when "*having* or *showing* special *skill* or *knowledge* because of what you have been *taught* or what you have *experienced*", as defined by the Merriam-Webster[1] dictionaries. One can notice that it implies to look at the intrinsic properties of the performer (i.e. having skill or knowledge) as well as the perception of some evaluators (i.e. showing skill or knowledge). Research literature in Psychology also identifies properties for *expertise*, which builds on long experience and high performance [13], as well as *expert*, who is identified through such expertise as well as social recognition [3].

While such a literature exists, designing EF techniques in Computer Science remains rather intuition-based [13], and recent works still validate their approaches by evaluating the output of their technique through domain experts [7, 14, 17]. Usually, they design their own technique based on indicators they think are of relevance, and validate it through experts identified based on social recognition [14, 17], self-evaluations [7], or other resources they did not use in their own technique [7]. This kind of validation brings significant threats: (i) output validations occur only at late stages, delaying the identification of

[1] http://www.merriam-webster.com/

This PDF is a personal version of the paper *Breaking the Recursivity: Towards a Model to Analyse Expert Finders*, from Matthieu Vergne and Angelo Susi, published in 2015 in the *Lecture Notes in Computer Science* vol. 9381, pp 539–547. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-25264-3_40.

P. Johannesson et al. (Eds.): ER 2015, LNCS 9381, pp. 539–547, 2015.
DOI: 10.1007/978-3-319-25264-3_40

inadequate techniques, and (ii) we need an already valid EF technique to find these domain experts. We faced this situation for our own EF approach [15] and it hurts the reliability of the validation process, thus we need to find a way to validate EF techniques without relying on domain experts, or at least not only on them.

After clarifying the problem and issues we want to tackle in Section 2, this paper contributes to the research community by (i) identifying some *relevant literature* in Psychology and EF techniques in Section 3, (ii) starting the building of a generic, grounded *conceptual model* for expertise evaluation in Section 4.1, and (iii) performing a *model-driven analysis* of the described EF techniques to illustrate its use in Section 4.2. While our research exploits both the perspectives of the *performer* and the *evaluator* provided by the literature, we restrict here to the latter and add a perspective on the *evaluation* (perception of the evaluator) to focus on EF techniques. Our model, which relies on scientific evidences in expertise evaluation in general, focuses on the foundational basis for the early stages (design and implementation) of an EF technique, thus offering a good complement to output validations through domain-specific experts. We are convinced that other interpretations from the modelling community and other references could help in building a more complete and reliable expertise evaluation model. In the long term, having such a complete model with proper guidelines could help evaluating existing EF techniques through a model-driven analysis by identifying strengths and limitations, and to fasten the design of new EF techniques by suggesting expertise indicators.

2 Expert Finding for Expert Finding: a Recursive Problem

EF is an important task, especially in research where we exploit the knowledge of domain experts to validate conceptual models and the outputs built based on them. A significant problem is that it also applies to EF techniques themselves, which aim at recommending domain experts from a given community, and thus to validate their recommendations through domain experts [7, 14, 17]. They find their "validation experts" through social recognition [14, 17], self-evaluations [7], or other resources not used in their own approach [7].

This "recursive" problem makes EF techniques hard to validate, because the domain experts could be biased [1] and have limited knowledge on the actual expertise of other people in the community [9], leading to a poor validation. We could think for instance about Open Source forums or international companies, where hundreds of people can be involved, thus making it hard to know everyone and in particular who are the most experts. One could consider different cases of application of the EF technique to mitigate this issue, but trying to find and involve the relevant experts could require a significant amount of time and effort. Moreover, this kind of validation focuses on the output of the EF techniques, meaning that we could assess the effectiveness of the technique only in late implementation stages. One could rely on EF techniques already employed in the community, assuming they are empirically validated, but it could lead to techniques which are hard to generalize to other contexts [9].

In this paper, we build an initial, generic conceptual model of expertise evaluation, to support early analysis without relying on domain experts. In particular, we would like to know (i) which concepts and relations are generic enough to appear in this model, implying the review of some relevant literature, (ii) which strengths and limitations can already be found in existing EF techniques, thus analysing them in the light of our model, (iii) and which parts of the model should be completed or refined, thus discussing the current state of the model in the light of the previous analysis. Consequently, this paper provides a model which can already support such analysis, but which could be further improved and validated.

3 Expert Finding Literature Review

3.1 Recent Expert Finding Techniques in Computer Science

Some EF techniques rely on *direct* contributions of performers to evaluate their expertise. For example, Mockus and Herbsleb [10] analyse the amount of code written in a piece of a software to identify knowledgeable programmers. They rank them relatively to the number of changes they made on the source code, possibly restricting the counting to a given period of time. Similarly, Serdyukov and Hiemstra [12] analyse the content of many documents to identify the contributions of their different authors, which helps in identifying their potential knowledge (i.e. terms used). They compute the probability that a given document or a given term relates to a given author and, when looking for experts related to a specific term, sum up the corresponding probabilities to rank the authors.

Other EF techniques rely on *indirect* indicators, especially how much people are recognized as experts into a given community. Zhang et al. [17] look at question/answers forums in an online community to identify people seeking and providing knowledge. In their work, they compare several algorithms to rank people, starting from the simple counting of answers, assuming it is positively correlated with the level of expertise. Another algorithm combines it with the number of questions written, which should be negatively correlated to the level of expertise. A third algorithm propagates these values over the community (PageRank-like), so that people answering questions from experts are themselves considered as more experts.

Finally, some works combine both indicators, direct as well as indirect, for evaluating expertise. Karimzadehgan et al. [7] exploit the content of the e-mails of employees to retrieve their potential knowledge (i.e. terms and topics), but also exploit hierarchical similarities among employees. They compute probabilities similarly to Serdyukov and Hiemstra [12], but smooth the results between hierarchically-related employees to mitigate the potential lack of data for some of them. We also proposed our own approach [15] which explicitly intended to exploit direct evidences of knowledge (i.e. terms and topics) added to social aspects (i.e. roles). We counted co-occurrences, such as how many times someone used a term, a term is used in a topic, or a role is assigned to someone, to build a weighted graph and compute and propagate probabilities all over it, allowing us to rank people.

3.2 Expertise in Psychology

Behind the fact that some EF techniques use specific indicators, we are also interested in how, generally, people build their own expertise, in order to find what are the relevant indicators to consider. Ericsson [2] summarizes a broad literature on this purpose. In particular, an *acceptable level of proficiency* requires some months of experience during which the performer will focus on the actions to perform while avoiding gross mistakes, like in school or any other training course. An *average, independent professional proficiency*, which means performing in an autonomous way, requires often several years, what we call a *lengthy domain-related experience*, to become fluent in the domain-relevant activities. However, what differentiates the average professional, who maintains his level by executing routine work, from the domain *expert* (or *master*) is the continuation of *deliberate practice* to fix weaknesses [5].

Focusing more on the perspective of someone looking for experts, the main perspective for EF, we can consider the review of Chi [1] who presents the two main approaches used to study expertise. The *absolute approach*, on one hand, studies exceptional people to understand what distinguishes them from the masses, in order to identify the properties which allow to reach the top (potentially some innate capacities). The *relative approach*, on the other hand, focuses on distinguishing people within a common, domain-related group, in order to identify what can be provided to the less experts to reach the level of the more experts. Chi [1] also summarizes the *properties* which seem to characterize experts, who excel for example by generating better solutions faster, perceiving deep features, identifying lacks and errors, and managing better their resources (e.g. skill, knowledge, sources of information). However, she also highlights that experts fail in showing similar excellence in different domains and in judging non-expert abilities, as well as they can be over-confident in their abilities, overlook details, and show more biases when their expertise does not apply.

While the literature provide us useful indicators to consider, Ericsson [2] notices that people evaluating the expertise of a performer often rely on simple experience-based indicators, which do not help in finding the highest experts. In these "good but not best" indicators, we can find the length of experience in the domain, the accumulated accessible knowledge, the completed education and the social reputation. In order to identify the highest experts, one need to look at *reproducibly superior performance* on representative, authentic tasks which require domain-specific experience, like a chess master should find the best move on a chess board already set up. However, when such direct evidences are lacking, we think that evidences of deliberate practice could help identify expert-like behaviours, complementing the simple experience-based indicators criticized by Ericsson [2].

4 Conceptual Model of Expertise Evaluation

4.1 Conceptual Model

In order to differentiate common terms from the concepts introduced by the model, shown in Figure 1, we use `This Font` for the concepts of the model.

We start by modelling the the *domain* relating the performer and evaluator, which starts from the `Domain` root concept (e.g. databases, or DB) and assume that it relates to a `Domain Community`. Within this `Domain Community`, we will find the `Performer`s (e.g. DB programmers) who produces the `Outcome`s (i.e. products, services or ideas) relevant to the `Domain`. We will also find the `Evaluator`s (e.g. recruiters) evaluating these `Performer`s based on their `Outcome`s, and who will influence/be influenced by some `Social Recognition`. In order to perform well and to produce creative ideas, a `Performer` hopefully consider already existing `Outcome`s which have been recognized as `Domain Prior Achievement`s, meaning `Outcome`s which have received a significant amount of `Social Recognition` from previous `Evaluator`s (Ericsson [4]).

From the perspective of the `Evaluator`, one considers the `Outcome`s of a `Performer` in order to build the `Perceived Expertise`. More precisely, the `Evaluator` should identify evidences of `Lengthy Domain-Related Experience` (e.g. 10 years experience in DB) to assess a reasonable level of expertise, and evidences of `Reproducibly Superior Performance` (e.g. several projects with complex data) for the highest levels of expertise (Sonnentag et al. [13], Ericsson [2]). Additionally to these direct evidences inferred from the `Outcome`s, some other `Evaluator`s could have provided their own `Perceived Expertise` (e.g. LinkedIn endorsements), leading to the building of some `Social Recognition`, which can be reused by the current `Evaluator` to refine or complete his or her judgement (Ericsson [3]). Going further in detail, we decompose the `Perceived Expertise` into both the `Perceived Domain Knowledge` and the `Perceived Domain Skill`s. They correspond to the specific items supporting the identification of `Lengthy Domain-Related Experience` and `Reproducibly Superior Performance`. These items, however, should be known by the `Evaluator` in order for him to identify them, thus he should have some `Owned Domain Knowledge`.

Because we focus on EF techniques, we go even further in the granularity of *evaluator* by modelling the *evaluation* he produces. A `Performance Evaluation` represents the `Perceived Expertise` of an `Evaluator`, and can be an `Absolute Performance Value` or a `Relative Performance Value` (Chi [1]). Typically, we use `Performance Level`s to express `Absolute Performance Value`s, while we compare `Performer`s through `Performers Ordering`s to express `Relative Performance Value`s. Going more in detail for the `Performance Level`, a concrete scale can be used, such as the `Novice-Master` scale described by Chi [1] (table 2.1, p. 22). If the `Evaluator` cannot use a concrete scale, he can rely on evidences of `Lengthy Domain-Related Experience` to assess an `Average Performance Level`, while additional evidences of `Reproducibly Superior Performance` would help identifying the `Highest Performance Level`s (Sonnentag et al. [13], Ericsson [2]).

4.2 Preliminary Analysis of Expert Finding Techniques

By analysing the works presented in Section 3.1, we can see that Serdyukov and Hiemstra [12] focus mainly on `Perceived Domain Knowledge` items by identifying the terms used. In particular, by evaluating how much a person contributes compared to all the others (via normalization), these approaches infer `Absolute Performance Value`s (i.e. probabilities) and recommend the people having the highest ones. While we could imagine that the values computed could help to infer `Performance Level`s, this approach would need to be completed with correlations between their values and proper levels. Moreover, while such approach is probably efficient to build the `Perceived Domain Knowledge`, it lacks the `Perceived Domain Skill` dimension. Going further, these approaches probably identify evidences of domain-related experience but not necessarily of `Lengthy Domain-Related Experience`, making it difficult to assess even an average level, unless the assumption of a lower bound expertise can be supported by the specific type of documents considered (e.g. peer-reviewed papers accepted for publication). Such assumptions, however, would probably not help in discriminating good from exceptional `Performer`s, meaning finding evidences for `Reproducibly Superior Performance`.

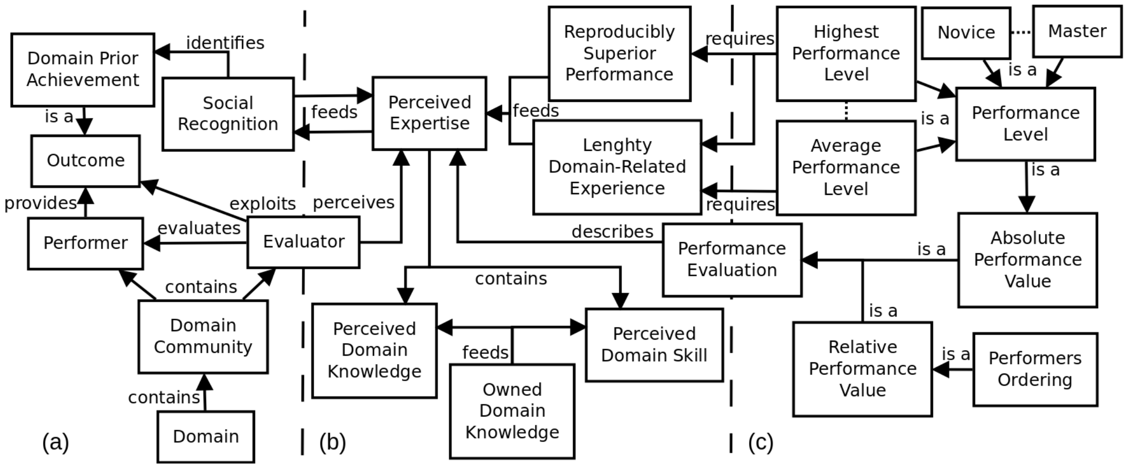

Figure 1: Conceptual model of the `Domain` (a), `Evaluator` (b), and `Performance Evaluation` (c).

Summarizing on the other works presented in Section 3.1, social approaches like Zhang et al. [17] consider `Social Recognition` indicators to provide `Performers Ordering`s (i.e. one is less expert than another) or `Performance Level`s (e.g. *Newbie* or *Top Java expert*). Once again, this approach lacks the identification of `Perceived Domain Skill`s, and they also suffer the same difficulties than Serdyukov and Hiemstra [12] to identify clear evidences for `Lengthy Domain-Related Experience` as well as `Reproducibly Superior Performance`. We retrieve these difficulties in approaches combining documents and social analysis, like Karimzadehgan et al. [7] and our own approach [15]. Although they combine `Social Recognition` indicators (hierarchy for the former, roles for the latter) with `Perceived Domain Knowledge` indicators (terms and topics), they ignore the `Perceived Domain Skill`s.

Only Mockus and Herbsleb [10] provide a rather complete approach by considering the commits (changes on a software) made by programmers. Commits are, at the same time, good indicators of `Perceived Domain Skill`s (coding skills are major skills in software) as well as `Perceived Domain Knowledge` (module modified, names of the variables added/removed/changed, etc.). The number of commits made over time can also show a `Lengthy Domain-Related Experience`, while frequencies of commits per month could show reproducible performances, although it does not necessarily support the high quality required by `Reproducibly Superior Performance`s. Thus, while they already provide supports and results, our model highlights why they are able to do so and identifies potential improvements (i.e. identifying the highest levels of expertise). Though, these good results should be contrasted with the fact that this approach targets a specific `Domain` (software implementation) while the other approaches try to be more generic, making the task more difficult.

5 Discussion and Conclusion

Through this paper, we have seen that EF techniques for finding experts in a given domain involves generally a "recursive" problem, by relying on domain experts to validate it, while other validation methods could be used. In particular, we showed that we can build a generic, grounded model to analyse EF techniques during early stages (design and implementation) by relying on literature in other domains, like Psychology.

However, we should notice a significant incompleteness in our model, like the inability to relate the specific formulae of the existing techniques to specific concepts in the model (i.e. modelling the *evaluation processes*). We also miss notions like time, which is critical to identify `Lengthy Domain-Related Experience`, and relations between `Perceived Domain Knowledge/Skill` and `Lengthy Domain-Related Experience`. Going back to the literature already cited, we did not consider the expert properties provided by Chi [1] (i.e. generate better solutions faster, fail in judging non-expert abilities, etc.), while it could provide relevant indicators to exploit. Similarly, we rely exclusively on literature in Psychology to identify the main concepts (top-down), while it could be complemented with systematic literature reviews of existing EF techniques to identify relevant lower level concepts (bottom-up, like [16]). Other perspectives could also be considered, like creativity [4] (i.e. producing something new and useful), which seems to be a way to identify some of the highest experts.

Based on this conceptual model and its limitations, we think that discussions within the research community about EF design and validation could be of relevant interest, and we encourage people to exchange interpretations and further formalization. From these exchanges, relevant future works could be to have a better formalization of this model, not only more complete but also more rigorous, for instance by using ontologies like in [6]. We also think that a systematic literature review of the existing EF techniques could be useful, not only to identify concrete indicators, but also to see how the existing techniques could be classified with such a model. For example, categories of EF techniques focusing on knowledge indicators could be particularly suited for contexts lacking skills indicators, leading to recommend the right EF techniques depending on the context at hand.

References

[1] Michelene T. H. Chi. Two Approaches to the Study of Experts' Characteristics. In K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman, editors, *The Cambridge handbook of expertise and expert performance*, pages 21–30. Cambridge University Press, New York, NY, US, 2006.

[2] K. A. Ericsson. The Influence of Experience and Deliberate Practice on the Development of Superior Expert Performance. In K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman, editors, *The Cambridge handbook of expertise and expert performance*, pages 683–703. Cambridge University Press, New York, NY, US, 2006.

[3] K. A. Ericsson. An Introduction to Cambridge Handbook of Expertise and Expert Performance: Its Development, Organization, and Content. In K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman, editors, *The Cambridge handbook of expertise and expert performance*, pages 3–19. Cambridge University Press, New York, NY, US, 2006.

[4] K. Anders Ericsson. Creative expertise as superior reproducible performance: Innovative and flexible aspects of expert performance. *Psychological Inquiry*, 10(4):329–333, 1999.

[5] K. Anders Ericsson, Ralf Th Krampe, and Clemens Tesch-romer. The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3):363–406, 1993.

[6] Maryam Fazel-Zarandi, Mark S. Fox, and Eric Yu. Ontologies in Expertise Finding Systems: Modeling, Analysis, and Design. In Mohammad Nazir Ahmad, Robert M. Colomb, and Mohd Syazwan Abdullah, editors, *Ontology-Based Applications for Enterprise Systems and Knowledge Management*, pages 158–177. IGI Global, 2013.

[7] Maryam Karimzadehgan, Ryen W. White, and Matthew Richardson. Enhancing Expert Finding Using Organizational Hierarchies. In Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy, editors, *Advances in IR*, number 5478 in LNCS, pages 177–188. Springer, January 2009.

[8] Mark T. Maybury. Expert finding systems. *MITRE Center for Integrated Intelligence Systems Bedford, Massachusetts, USA*, 2006.

[9] David W. McDonald and Mark S. Ackerman. Just Talk to Me: A Field Study of Expertise Location. In *Proc. of the Conference on CSCW*, pages 315–324, New York, NY, USA, 1998. ACM.

[10] Audris Mockus and James D. Herbsleb. Expertise browser: a quantitative approach to identifying expertise. In *Proc. of the 24th ICSE*, pages 503–512, New York, NY, USA, 2002. ACM.

[11] J.G. Mohebzada, G. Ruhe, and A. Eberlein. Systematic mapping of recommendation systems for requirements engineering. In *2012 ICSSP*, pages 200 –209, June 2012.

[12] Pavel Serdyukov and Djoerd Hiemstra. Modeling Documents As Mixtures of Persons for Expert Finding. In *Proc. of the IR Research, 30th ECIR*, pages 309–320. Springer, 2008.

[13] Sabine Sonnentag, Cornelia Niessen, and Judith Volmer. Expertise in Software Design. In *The Cambridge handbook of expertise and expert performance*. Cambridge University Press, New York, NY, US, 2006.

[14] Jie Tang, Duo Zhang, and Limin Yao. Social Network Extraction of Academic Researchers. In *7th IEEE ICDM*, pages 292–301, October 2007.

[15] Matthieu Vergne and Angelo Susi. Expert Finding Using Markov Networks in Open Source Communities. In Matthias Jarke, John Mylopoulos, Christoph Quix, Colette Rolland, Yannis Manolopoulos, Haralambos Mouratidis, and Jennifer Horkoff, editors, *Advanced Information Systems Engineering*, number 8484 in LNCS, pages 196–210. Springer, June 2014.

[16] Dawit Yimam-Seid and Alfred Kobsa. Expert-Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. *Journal of Organizational Computing and Electronic Commerce*, 13(1):1–24, March 2003.

[17] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *Proc. of the 16th international conference on WWW*, pages 221–230, New York, NY, USA, 2007. ACM.