On the Need for a Body of Knowledge on Recommender Systems

Juri Di Rocco[1], Davide Di Ruscio[1], Claudio Di Sipio[1], Phuong T. Nguyen[1] and Claudio Pomo[2]

[1]*DISIM, Università degli studi dell'Aquila, 67100 L'Aquila, Italy*
[2]*SisInf Lab, Politecnico di Bari, 70125 Bari, Italy*

Abstract
Recommender systems (RSs) are becoming widespread in different application domains to provide personalized items to given service users. Because of such an increasing adoption of RSs, it is becoming urgent to define a precisely curated and organized core set of concepts and practices, i.e., a Body of Knowledge (BOK), as already done in other disciplines, including software engineering and model-driven engineering. The opportunities related to the availability of an RSBOK are manifold, and different stakeholders would benefit from it including, developers, teachers, and newcomers to the RS community. Further than motivating a BOK for recommender systems and discussing corresponding envisioned opportunities and challenges, we also propose a methodology that can be employed to support the definition of an RSBOK.

1. Introduction

Recommender systems (RSs) are complex software systems that can provide users with relevant items of interest for the particular application domains and contexts [1, 2]. Over the last decade, different types of recommendation technologies have been conceived by both industry and academia to improve the relevance of the items being recommended. RSs have become pervasive, and almost in any application domain, there is the availability of software systems in charge of supporting users in undertaking the particular tasks at hand (whether it be software developers who are working on some software components, or users who want to select the next movie to watch).

While RSs are becoming ubiquitous, we believe that it is necessary to ensure that the next generation of engineers will have a clear understanding of the fundamental techniques and tools underpinning the development and usage of RSs. To this end, it is necessary to agree on the core concepts, mechanisms, and practices related to the development and use of RSs. Therefore, as done in other software disciplines, we foster an RS Body of Knowledge (RSBOK) definition to formalize and characterize the Recommender System discipline.

To the best of our knowledge, SWEBOK is the first body of knowledge that was conceived for characterizing the software engineering discipline [3]. SWEBOK was defined with different goals, including "*1) promoting a consistent view of software engineering worldwide; 2) specifying the scope of, and clarify the place of software engineering with respect to other disciplines [. . .]; 3) characterizing the contents of the software engineering discipline; 4) providing a foundation for curriculum development and for individual certification and licensing material*" [3]. Similarly, a Body of Knowledge for Software Language Engineering has been recently promoted with the aim of assembling and organizing *"artifacts, definitions, methods, techniques, best practices, open challenges, case studies, teaching material, and other components that will afterwards help students, researchers, teachers, and practitioners to learn from, to better leverage, to better contribute to, and to better disseminate the intellectual contributions and practical tools and techniques coming from the SLE field*" [4]. A Body of Knowledge for Model-Based Software Engineering (MBSE) has been recently promoted [5] as an extension of SWEBOK to characterize the MBSE discipline in the context of Software Engineering. Many other BoKs have been defined over the last decade in different application domains including data management, enterprise architecture, business analysis, project management, and data management. All of them share the goals of characterizing the contents of a particular discipline and to support various activities including training and the development of new technologies and tools.

In this paper, we introduce some preliminary thoughts of an RS Body of Knowledge (RSBOK) definition for the Software Engineering domain. We promote a comprehensive perception of the core concepts, mechanisms, and practices related to the development and deployment of RSs. The ultimate aim is to understand the fundamental techniques and tools pertinent to the development and usage of recommender systems in software engineering. Altogether, this is expected to come in handy for those who work as recommender systems designers.

3rd Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) & 5th Edition of Recommendation in Complex Environments (ComplexRec) Joint Workshop @ RecSys 2021, September 27–October 1, 2021, Amsterdam, Netherlands
juri.dirocco@univaq.it (J. Di Rocco); davide.diruscio@univaq.it (D. Di Ruscio); claudio.disipio@graduate.univaq.it (C. Di Sipio); phuong.nguyen@univaq.it (P. T. Nguyen); claudio.pomo@poliba.it (C. Pomo)

2. Knowledge-aware Recommender Systems

Despite their enormous popularity and the remarkable performance that recommender systems have achieved in recent years, one of the long-standing problems affecting the performance of these systems concerns the sparsity of interactions between users and items. Over the past years, recommender system designers have relied on additional sources of information to overcome this issue. Modern RSs combine collaborative information with metadata (e.g., tags, reviews), social connections, image and audio signal-derived features, and contextual data [6] to build domain-dependent, cross-domain, or context-aware recommendation models. Among the various sources, one of the most relevant is Knowledge Graphs ($\mathcal{KG}s$). Thanks to the heterogeneous fields covered by $\mathcal{KG}s$ and the myriad of specific techniques that have been developed, knowledge-based recommendation systems emerged as a novel research field in the RecSys community. The field is generally known as knowledge-aware recommendation systems (KaRS [7]) and blends the most advanced machine learning algorithms with cutting-edge knowledge representation paradigms. This collective effort has resulted in several improvements in recommendation [8], knowledge completion [9], preference elicitation, and user modeling research, thus producing a vast literature.

3. Opportunities and Challenges

Conceiving a Body of Knowledge for Recommender Systems (RSBOK) would disclose several opportunities based on the availability of a common and formally defined vocabulary that would prescribe the usage and development of recommender systems on clearly defined foundations, instead of relying on some uncommon understanding.

The RSBOK definition is indeed a community effort to formalize and share knowledge from different stakeholders on conceptual and practical RS aspects. However, the investment would pay off because we foresee at least the following kinds of users that can take advantage of the RSBOK paradigm [4]:

- *Newcomers*: They are perspective users and researchers who do not have any knowledge about RSs, and are willing to understand them. They can benefit from the results of the conceptualization efforts, e.g., to get an overview of the typical algorithms employed to develop RSs, or to explore available linked textual explanations or examples about some typically used evaluation methodologies.
- *Developers*: These include advanced RSs developers, who are interested in conceiving techniques and tools to widen and simplify the adoption of RSs in any complex software systems. The knowledge encoded in the envisioned RSBOK framework can help developers, among others, to distill the algorithms that should be made available to final users.
- *Teachers*: They are researchers, practitioners, and educators in general who are in charge of training new generation of professionals in the design, development and operation of new RSs. To this end, RSBOK can be used as a reference point, e.g., to link code examples, textual explanations, experiences, RS usage and development best practices, and code examples that demonstrate the usage of RS technologies.
- *Contributors*: These users correspond to RSs developers, and adopters who are willing to extend the knowledge formalized in RSBOK.

As previously mentioned, the wanted RSBOK paradigm should consist of a precise formalization of concepts and tools underpinning the development, usage and enhancement activities of any RSs. It is expected to provide an effective means to characterize and to support different related activities including training and the development of new technologies and tools. Recently, we had already the need to investigate and formalize the RS field in Software Engineering, and we came up with a model representing all the features typically supported by RSs [10]. Figure 1 represents only the top-level features or recommendation systems, i.e., *Data Preprocessing*, *Capturing Context*, *Producing Recommendations*, and *Presenting Recommendations* which are the main functionalities typically implemented by recommendation systems [11, 12].

We performed such a conceptualization to underpin the design and development of the different RSs developed in the context of the EU CROSSMINER project [10]. We extracted all the shown components mainly from existing studies [11] as well as from our development experience under the needs of the CROSSMINER project.

A similar conceptualization work has been done to design and develop the Elliot framework [13], which aims at supporting reproducible recommender systems evaluation. The authors had to conceptualize different recommendation algorithms, splitting strategies, evaluation protocols, metrics, and tasks to simplify the specification and execution of experimental pipelines by processing simple configuration files as the one shown in Fig. 2

Even though the opportunities related to the availability of an RSBOK are immense in our opinion, its realization can be hampered by a number of challenges including the following ones:

- *Languages to be adopted:* To make the definition

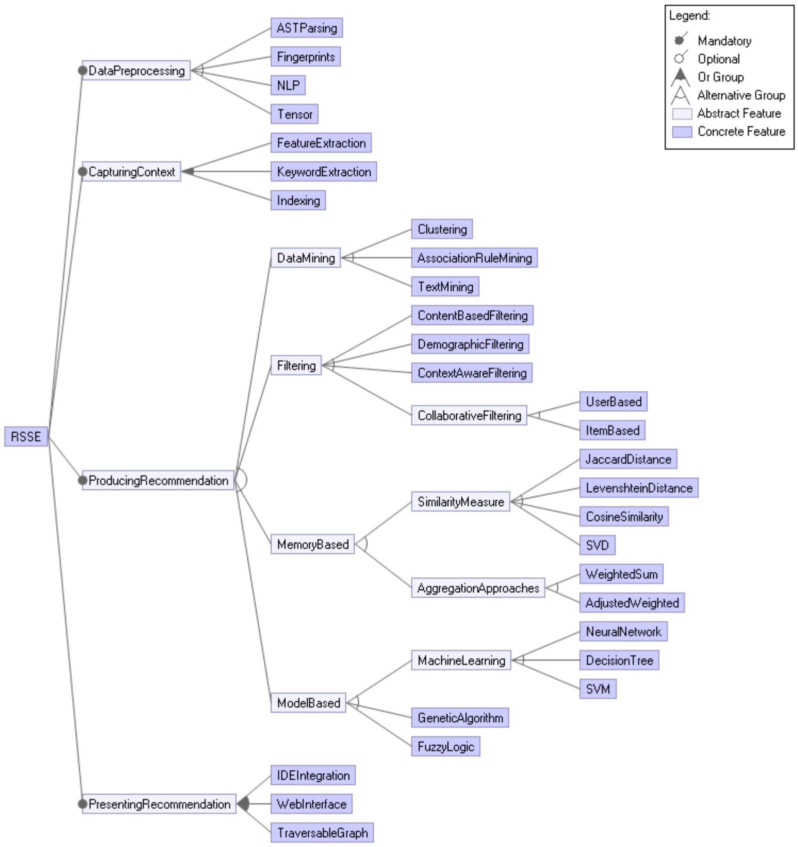

Figure 1: Main design features of recommendation systems in software engineering (refinement of Di Rocco *et al.* [10]).

Configuration 1: `hello_world.yml`

```
experiment:
  dataset: movielens_1m
  data_config:
    strategy: dataset
    dataset_path: ../data/movielens_1m/dataset.tsv
  splitting:
    test_splitting:
        strategy: random_subsampling
        test_ratio: 0.2
  models:
    ItemKNN:
      meta:
        hyper_opt_alg: grid
        save_recs: True
      neighbors: [50, 100]
      similarity: cosine
  evaluation:
    simple_metrics: [nDCG]
  top_k: 10
```

Figure 2: Simple Elliot configuration file.

and the usage of RSBOK homogenous, it is necessary to decide the languages and tools that need to be adopted. Feature diagrams, Ecore models, and OWL ontologies are only examples of possible notations that might be employed to formalize the results of the conceptualization efforts.

- *Realization process:* By looking at the ways other BOKs have been developed, we believe that we need a community effort, which has to be performed by following precise protocols, moderation mechanisms, quality check procedures, to name a few. In this respect, it is of great importance to support contributions that may come from different stakeholders. Consequently, to keep the quality of the resources under control, it is necessary to define processes and setup tools for moderating the different contributions and to make sure that they are all homogenized.
- *Engagement:* Even though the RSs opportunities can be convincing, they might not be enough to engage people in concretely contributing with the RSBOK definition and managing its whole lifecycle.

4. Conclusion and Future Work

To facilitate a clear understanding of the fundamental techniques and tools underpinning the development and usage of RSs, in this paper we envisaged the core concepts, mechanisms, and practices related to the development and use of RSs. We aim to foster RSBOK, an RS Body of Knowledge definition to formalize and characterize the Recommender System domain.

For future work, we plan to tackle the challenges mentioned in Section 3. In particular, it is necessary to put into effect the conceived paradigm by realizing its constituent components. Among others, we will investigate and select suitable tools and languages, attempting to make the definition and usage of RSBOK homogeneous.

References

[1] M. P. Robillard, W. Maalej, R. J. Walker, T. Zimmermann (Eds.), Recommendation Systems in Software Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. doi:10.1007/978-3-642-45135-5.

[2] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), Recommender systems handbook, Springer US, Boston, MA, 2011, pp. 1–35.

[3] P. Bourque, R. E. Fairley, IEEE Computer Society, Guide to the Software Engineering Body of Knowledge, 2014.

[4] B. Combemale, R. Lämmel, E. Van Wyk, SLEBOK: The Software Language Engineering Body of Knowledge (Dagstuhl Seminar 17342) (2018) 10 pages. doi:10.4230/DAGREP.7.8.45.

[5] F. Ciccozzi, M. Famelis, G. Kappel, L. Lambers, S. Mosser, R. F. Paige, A. Pierantonio, A. Rensink, R. Salay, G. Taentzer, A. Vallecillo, M. Wimmer, Towards a body of knowledge for model-based software engineering, in: Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, ACM, Copenhagen Denmark, 2018, pp. 82–89. doi:10.1145/3270112.3270121.

[6] V. W. Anelli, V. Bellini, T. D. Noia, W. L. Bruna, P. Tomeo, E. D. Sciascio, On the role of time and sessions in diversifying recommendation results, in: F. Crestani, T. D. Noia, R. Perego (Eds.), Proceedings of the 8th Italian Information Retrieval Workshop, Lugano, Switzerland, June 05-07, 2017, volume 1911 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 92–96. URL: http://ceur-ws.org/Vol-1911/16.pdf.

[7] V. W. Anelli, P. Basile, D. G. Bridge, T. D. Noia, P. Lops, C. Musto, F. Narducci, M. Zanker, Knowledge-aware and conversational recommender systems, in: S. Pera, M. D. Ekstrand, X. Amatriain, J. O'Donovan (Eds.), Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018, ACM, 2018, pp. 521–522. URL: https://doi.org/10.1145/3240323.3240338. doi:10.1145/3240323.3240338.

[8] V. W. Anelli, T. D. Noia, E. D. Sciascio, A. Ragone, J. Trotta, How to make latent factors interpretable by feeding factorization machines with knowledge graphs, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. F. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web - ISWC 2019, Proceedings, Part I, volume 11778 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 38–56. URL: https://doi.org/10.1007/978-3-030-30793-6_3. doi:10.1007/978-3-030-30793-6_3.

[9] G. He, J. Li, W. X. Zhao, P. Liu, J. Wen, Mining implicit entity preference from user-item interaction data for knowledge graph completion via adversarial learning, in: Y. Huang, I. King, T. Liu, M. van Steen (Eds.), WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, ACM / IW3C2, 2020, pp. 740–751. URL: https://doi.org/10.1145/3366423.3380155. doi:10.1145/3366423.3380155.

[10] J. Di Rocco, D. Di Ruscio, C. Di Sipio, P. T. Nguyen, R. Rubei, Development of recommendation systems for software engineering: the CROSSMINER experience 26 (2021) 69. URL: https://doi.org/10.1007/s10664-021-09963-7. doi:10.1007/s10664-021-09963-7.

[11] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl. Based Syst. 46 (2013) 109–132. URL: https://doi.org/10.1016/j.knosys.2013.03.012. doi:10.1016/j.knosys.2013.03.012.

[12] A. Bellogín, A. Said, Improving accountability in recommender systems research through reproducibility, CoRR abs/2102.00482 (2021). URL: https://arxiv.org/abs/2102.00482. arXiv:2102.00482.

[13] V. W. Anelli, A. Bellogín, A. Ferrara, D. Malitesta, F. A. Merra, C. Pomo, F. M. Donini, T. D. Noia, Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation, in: F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, T. Sakai (Eds.), SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 11-15, 2021, ACM, 2021, pp. 2405–2414. URL: https://doi.org/10.1145/3404835.3463245. doi:10.1145/3404835.3463245.