(Re)Configuration using Web Data: A Case Study on the Reviewer Assignment Problem*

Anna Ryabokon[1], Axel Polleres[2], Gerhard Friedrich[1], Andreas A. Falkner[2], Alois Haselböck[2], and Herwig Schreiner[2]

[1] Alpen-Adria Universität, Klagenfurt, Austria
firstname.lastname@ifit.uni-klu.ac.at
[2] Siemens AG Österreich, Vienna, Austria
firstname.{middleinitial.}lastname@siemens.com

Abstract. Constraint-based configuration is – on the one hand – one of the classical problem domains in AI and also in industrial practice. Additional problems arise, when configuration objects come from an open environment such as the Web, or in case of a reconfiguration. On the other hand, (re)configuration is a reasoning task very much ignored in the current (Semantic) Web reasoning literature, despite (i) the increased availability of structured data on the Web, particularly due to movements such as the Semantic Web and Linked Data, (ii) numerous practically relevant tasks in terms of using Web data involve (re)configuration. To bridge these gaps, we discuss the challenges and possible approaches for reconfiguration in an open Web environment, based on a practical use case leveraging Linked Data as a "component catalog" for configuration. In this paper, we present techniques to enhance existing review management systems with (re)configuration facilities and provide a practical evaluation.

1 Introduction

Constraint-based configuration, i.e. picking and linking a suitable set of components from a component catalog s.t. some predefined constraints are satisfied is a classical problem in AI and also in industrial practice. As users of the Web, we often solve such configuration tasks where in theory the "component catalog" is the Web, e.g. as private persons configuring an itinerary (flight, accommodation, hotel, etc.), or as academics, in the task of assigning expert reviewers to papers. Emerging availability of Linked Data on the Web [6] allows us to apply known configuration techniques to such problems that have been solved manually by Web search. However, Linked Data and adjacent Semantic Web communities focus mainly on taxonomic reasoning and ontologies (RDF Schema, OWL), to better structure Web data or infer implicit Web data, thus largely ignoring, to the best of our knowledge, reasoning tasks required for the configuration.

In this paper we show how configuration can be implemented in the framework of Linked Data. Moreover, we discuss an extension of the original problem for the

* This work was funded by FFG FIT-IT within the scope of the project RECONCILE (grant number 825071). An extended version of the paper can be found at http://proserver3-iwas.uni-klu.ac.at/reconcile/index.php/benchmarks

cases when configuration does not start from scratch. In this case a previously consistent configuration has to be adapted, i.e. a reconfiguration is required. Reconfiguration is an important task in the after-sale life-cycle of configurable products and services, because requirements are changing and there is a need to keep a product or a service up-to-date [2]. As it has been shown in the previous work [3], (re)configuration tasks can be efficiently handled by Answer Set Programming (ASP) [1] which extends logic programming and includes an expressive modeling language and solving tools [4].

The feasibility of (re)configuration based on Open Web Data in a practical scenario is demonstrated on the reviewer assignment problem (RAP): The decision if a paper is accepted on a conference depends on reviews made by the program committee. Therefore, it is required to assign every paper to a number of reviewers such that on the one hand these reviewers are interested in reading the paper and on the other hand have enough expertise. Our experiments show that the reviewer (re-)assignment task, leveraging Open Data and deploying methods of reconfiguration using SPARQL[9] and ASP, can efficiently be applied in practice.

2 (Re)Configuration using Web Data

The reviewer assignment problem can be viewed as a configuration task where papers must be linked to reviewers such that a set of problem specific constraints are fulfilled. A preferred solution can be determined based on an optimization function which ranks the set of valid reviewer/paper assignments (i.e. configurations). In addition, reviewers typically specify their preferences in a process of bidding on the one hand, and on the other hand papers should be reviewed by the most competent reviewers among the program committee (PC). Whereas bidding preference are usually collected by a conference management system, the "expertise match" between reviewers and papers is normally not given explicitly and has to be estimated by program or area chairs while assigning the papers in existing systems, if it is taken into account at all.

In our approach we consider four categories of expertise: *conflict* if a reviewer is an author of the paper or biased by some other circumstances; *low*, *moderate* and *high* expertise. Moreover, the preferences of reviewers provided by the bidding process are encoded as: *conflict* of interest declared by a reviewer; *indifference*, i.e. no bid is provided; *weak* and *strong* willingness to review the paper. The latter two categories correspond to "I can review" and "I want to review" in EasyChair. The goal is to find a match between reviewers and papers s.t. different preferences are reconciled.

Goldsmith and Sloan [5] propose to view RAP as a variant of the stable marriage problem. A paper/reviewer assignment (marriage) is stable if there does not exist an alternative assignment in which paper *and* reviewer are individually better off than in their current assignment. Consequently, a reviewer cannot spot a paper which she prefers more and for which she has more competence compared to the current assignments. There are several variants of the stable matching which differ from the classic stable marriage problem: *polygamy* – reviewers can get more than one paper and vice versa; *incomplete lists* – some reviewers or papers cannot be assigned to each other; and *indifference* – the preferences express a preset number of preference classes. Each variation of the Stable Marriage Problem mentioned above can be solved in polynomial time [5]. The problem becomes more complicated and is known to be NP-hard if both incomplete lists and indifference occur [8] as in the paper assignment variant of the stable marriage problem. Therefore, a problem solving method which is able to deal with NP-hard problems is required and justifies the usage of ASP as a problem representation and solving framework.

In order to reduce the load on the solver we consider stability as a soft constraint and minimize the number of assignments which do not fulfill the stability property. In addition, we minimize the number of assignments of papers to reviewers with low and moderate expertise as well as of reviewers to papers with indifference and weak willingness. The encoding includes also the following hard constraints: (1) each paper must be assigned to a fixed number of reviewers and (2) fairness of the workload should be achieved. In order to distribute the papers among the reviewers as uniformly as possible, we add a *balancing criterion* as a hard constraint, which limits the minimum and maximum number of papers assigned to each reviewer.

We use Linked Data [6] to extract valuable information about connections between authors, such as recent co-authorship, joint affiliation or create expertise profiles. The first two types of connections allow automatic recognition of conflicts of interests. The profiles can be used to compare abstracts or keywords of published papers to submissions, thus determining the level of expertise. For a proof-of-concept implementation we have selected a fictitious set of reviewers composed of persons mentioned at `data.semanticweb.org`, as well as a subset of papers mentioned there as fictitious set of submissions. We also retrieve information about recent co-authorship from `http://dblp.l3s.de/d2r/` where we only link authors with unambiguous unique names present in both DBLP and `data.semanticweb.org`. In this paper, we make the reasonable assumption that the more similar the paper abstract and the abstracts of a reviewer are, the more competent the reviewer is to evaluate the paper. In order to compute these similarities we extracted abstracts of submitted papers and papers written by reviewers using SPARQL queries to `data.semanticweb.org`. The set of abstracts was analyzed by established methods from information retrieval and recommender systems [7] as follows: First, we derived a list of keywords relevant to the abstracts of papers and reviewers by considering only those terms which are provided by the PC chair of a conference in form of keywords. Next, we clean the keywords by employing a lemmatizer such as `http://morphadorner.northwestern.edu/`. The result of this process is a term vector for each reviewer and each paper, which we use in a standard *term frequency – inverse document frequency* (TF/IDF) weighting of the paper's abstract as well as of the union of abstracts for each reviewer. The similarities of vectors describing the papers and vectors describing the reviewers are computed by the *cosine similarity measure* [7].

The similarities were used to generate RAP instances of different size including a set of reviewers and papers as well as their bids and expertises. For each instance we applied ASP solver to find solutions of both configuration and reconfiguration problems. The instances to the latter problem are obtained by modifying corresponding solutions of the configuration problem. In the case of RAP modifications include situations when reviewers may drop out, papers could be withdrawn, or additional conflicts of interests may be discovered. The transformation of the legacy configuration possibly requires that some of its parts are deleted. Therefore, each reconfiguration problem instance includes requirements and transformation knowledge regarding reuse or deletion of parts of a legacy configuration.

We employ the modeling patterns described in [3] to formulate a reconfiguration problem instance. The principle idea is that for every element of the legacy configuration a decision has to be made whether or not to delete or reuse this element. The reused elements are complemented on demand by addition of new elements in order to fulfill all requirements. Note that in the reconfiguration case the optimization criteria of a configuration problem are extended with a criteria minimizing the costs associated with the transformation actions such as delete, reuse or create.

The evaluation results were performed using Potassco ASP collection [4] show that the reasoner was able to find a solution for all test instances and we obtained the best configuration solutions that can be computed within a timeout period of 900 seconds; proving optimality for such (re)configuration instances seems to be infeasible in practice. Note that, the performed experiments have realistic number for PC members and submissions comparable with e.g. the last ISWC conferences from which we took the data. For the reconfiguration problem instances the solver was able to find solutions with optimal reconfiguration costs in all but the two biggest cases from 16 which were tested. In these two cases the solver found solutions which reconfiguration costs are 20% and 8% higher than the optimum. A solution with the optimal reconfiguration costs was usually identified by the solver in the first 10 seconds of the solving process excluding the grounding time. The obtained results show that the proposed method is feasible for realistic reviewer assignment problems.

References

1. Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. Communications of the ACM 54(12), 92–103 (2011)
2. Falkner, A., Haselböck, A.: Challenges of Knowledge Evolution in Practice. In: Workshop on Intelligent Engineering Techniques for Knowledge Bases (IKBET 2010). pp. 1–5 (2010)
3. Friedrich, G., Ryabokon, A., Falkner, A., Haselböck, A., Schenner, G., Schreiner, H.: (Re)configuration using Answer Set Programming. In: Proceedings of the IJCAI 2011 Workshop on Configuration. pp. 17–25 (2011)
4. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Artif. Intell. 187, 52–89 (2012)
5. Goldsmith, J., Sloan, R.: The AI conference paper assignment problem. In: Proceedings of AAAI Workshop on Preference Handling for Artificial Intelligence. pp. 53–57. Vancouver, BC, Canada (2007)
6. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, Synthesis Lectures on the Semantic Web: Theory and Technology, vol. 1. Morgan & Claypool (2011)
7. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press (2010)
8. Manlove, D., Irving, R., Iwama, K., Miyazaki, S., Morita, Y.: Hard variants of stable marriage. Theoretical Computer Science 276(1-2), 261–279 (2002)
9. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation, World Wide Web Consortium (2008), http://www.w3.org/TR/rdf-sparql-query/