NeLMeS: Finding the Best Based on the People Available Leveraging the Crowd

Fabian Wiedemann(✉), Ralph Sontag, and Martin Gaedke

Technische Universität Chemnitz, Chemnitz, Germany
{fabian.wiedemann,ralph.sontag,martin.gaedke}@informatik.tu-chemnitz.de

Abstract. An in-house crowd is composed of persons with different skills, where they can be assigned to different projects on demand. Persons forming an in-house crowd have to meet for different purposes, such as planning the project or distributing objectives. While scheduling meetings is an important task in a knowledge worker's daily business, the problem evolves by adding more specific constraints to the scheduling. It is difficult for a company that uses an in-house crowd to decide which project can be executed on which date. To deal with this problem we propose the NeLMeS approach. NeLMeS extends the principle of meeting scheduling by defining projects and their required skills as well as persons who have some skills. We present a first prototype that uses LinkedIn to extract person skills and schedules which project can be executed at each date.

Keywords: Crowdsourcing · Project management · Meeting scheduling · Web applications

1 Introduction

Crowdsourcing is an increasingly popular paradigm for accomplishing tasks for which persons or companies do not have the required manpower or specific skill [4]. While crowdsourcing in general involves external persons into sensitive internal processes of a company, the concept of an in-house crowd is more applicable. In an in-house crowd persons with different skills are employed who are not assigned to a specific department or division. That is, these persons can be assigned to several projects if their skills are required to accomplish diverse objectives within these projects. To organize the work of one project it is necessary to meet with persons involved in this project. For scheduling meetings efficiently, it would be beneficial to know which project can be executed on which date based on the available persons from the in-house crowd.

In [6] Sen and Durfee propose an approach for distributed meeting scheduling by defining a metric of the efficiency of scheduling the meeting. It takes into account the communication cost and total time taken to schedule the meeting. In [5] Lee and Pan describe an architecture which involves scheduling meetings utilizing a fuzzy agent. Neither approach, however, considers the skills a person

P. Cimiano et al. (Eds.): ICWE 2015, LNCS 9114, pp. 687–690, 2015.
DOI: 10.1007/978-3-319-19890-3_58

has and the skills that are required to execute a project. While expert finding in the crowd is not within the scope of this paper, there are several approaches which deal with expert finding, such as [1,3].

Scheduling a meeting can be formalized as follows: A date d_i is element of the set of all possible $Dates$. The subset of persons $Pers_i \subset Pers$ is defined by the persons which are available at a date d_i ($pers_g \in Pers_i : d_i \in pers_g.Dates$). The meetings where all persons can attend is $d_i : Pers_i = Pers \ \forall d_i \in Dates$.

This paper deals with the problem of planning which project $prj_j \in Prj$ can be executed at a certain date d_i. That is, persons $pers_k \in Pers$ has one or more skills $s_a \in S_k \subset S$ and a project prj_j requires different skills $s_b \in S_j \subset S$ to be executed.

The Next-Level-Meeting-Scheduler NeLMeS focuses on extending the classical scheduling meeting functionality by suggesting which projects can be executed at a scheduled date. Therefore, our approach is applicable in an in-house crowd scenario to suggest which project can be executed on which date.

This paper also presents a prototype illustrating the feasibility of our approach. It uses LinkedIn for extracting and incorporating person skills.

2 The Next-Level-Meeting-Scheduler NeLMeS

To deal with the scenario of an in-house crowd described in Section 1 NeLMeS implements the overall workflow depicted in Figure 1. When launching a new project one person has to be responsible for the organization, execution and success of this project. This person is the project's directly responsible individual (DRI). The DRI defines the objectives and derives a list of required skills based on these objectives. The list of required skills is used to find experts from the in-house crowd. Persons assigned to a project can vote for dates where they are available for a meeting. Based on the required skills of a project and the available persons the NeLMeS approach proposes which project meetings could be held on which date.

To schedule which project can be executed on which date, NeLMeS considers three inputs: on which date a person is available; which person owns which skill; which skill is required by which project to be executed. The output is a plan that proposes which project can be executed on which date.

For each date, NeLMeS checks whether a project can be executed. That is, the tool checks if the skills all available persons have will match the required skills of a project. This is defined as a maximum matching problem in bipartite graphs. Thus, one person can only provide one skill to a project even if the person has two or more required skills. In a bipartite graph, the vertices can be divided into two parts so that no two vertices in the same part are joined by an edge. A maximum bipartite matching is the cardinality maximal set of edges without common vertices in a bipartite graph.

We model a bipartite graph with all available persons at the selected date on the one side and the required skills for the selected project on the other side (cf. Algorithm 1 lines 4 and 5). Then, we insert edges between a person and a

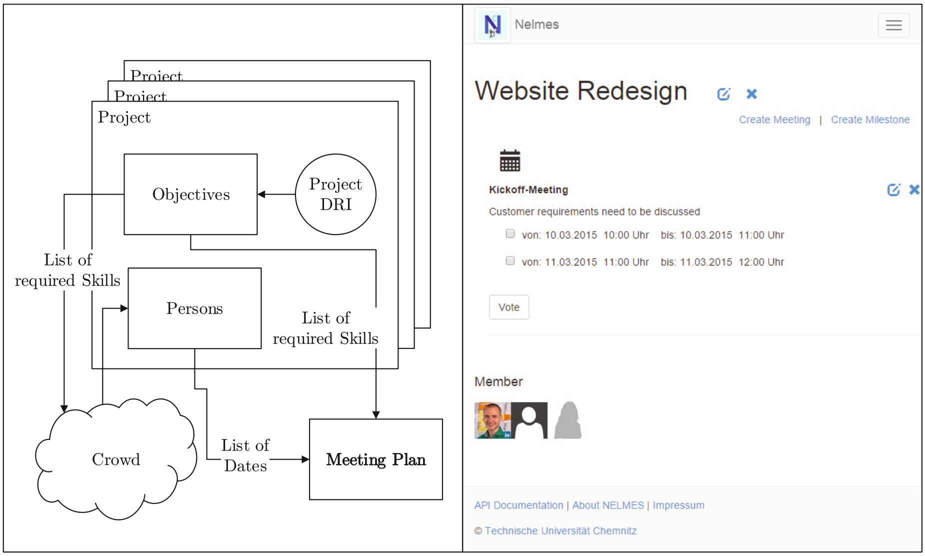

Fig. 1. Architecture of the NeLMeS approach

skill, if the person has that skill (cf. Algorithm 1 line 6). For this graph we have to solve a maximum bipartite matching. If the number of selected edges within a maximum bipartite matching is equal to the number of skills, the project can be executed at the selected date (cf. Algorithm 1 lines 11-13).

This maximum bipartite matching problem can be solved using linear programming. To solve this linear program NeLMeS employs the simplex algorithm [2]. While the Algorithm 1 results in a function (cf. line 12) which returns all possible projects prj_j for a given date d_i, our approach utilizes this function to generate the meeting plan for all given dates (cf. Figure 1).

For demonstrating NeLMeS we implemented a first prototype that uses LinkedIn as identity provider and for extracting person skills. Our prototype offers a RESTful API that is written in JavaScript and uses the runtime environment of node.js. To store data persistently we use the NoSQL database MongoDB. The front-end of our web application uses the frameworks Bootstrap for creating a responsive user interface that is easy to use on desktop computers as well as mobile devices.

Demonstration. The prototype presented in this paper is available for live testing at: http://vsr.informatik.tu-chemnitz.de/demo/nelmes/

3 Conclusion

NeLMeS eases scheduling meetings in an in-house crowd. After finding experts from the crowd it is possible to schedule which project can be executed at which

Algorithm 1. Algorithm of the NeLMeS approach

Input : Dates, Prj, Pers
Output : $f : Dates \rightarrow Prj$
for $d_i \in Dates$ **do**
  for $prj_j \in Prj$ **do**
    Let $G = (V, E)$ with $V = \{\}$ & $E = \{\}$ $S_j = prj_j.Skills$;
    Create vertices $v_g \in V\ \forall pers_g \in Pers : d_i \in pers_g.Dates$; Let V_1 be the set of newly created vertices;
    Create vertices $w_h \in V\ \forall s_h \in S_j$; Let V_2 be the set of newly created vertices;
    Create edges $e_{v_g,w_h} \in E\ \forall v_g \in V_1, w_h \in V_2 : s_h \in pers_g.Skills$;
    `// Linear program as` $A * x = b$ `and` $c^T x \rightarrow min$
    Let $A \in \{0,1\}^{|V| \times |E|} | A_{v,e} = \begin{cases} 1, \text{ if } v \in e\ \forall v \in V, e \in E \\ 0, \text{ otherwise} \end{cases}$;
    Let $c = (-1)^{|E|}$ be the vector of costs;
    Let $b = (1)^{|V|}$ be the vector of coefficients;
    $x = simplexAlgo(A, b, c)$;
    if $1^T x == |S_j|$ **then**
      $f(d_i) = f(d_i) \cup d_i \mapsto prj_j$
    end
  end
end

date. We modeled the problem described in this paper as a maximum bipartite matching. This problem is solved by our approach utilizing the simplex algorithm. We present a first prototype that uses LinkedIn to extract person skill.

Future work will focus on extending our approach towards taking account of not only skills, but the different skill levels persons have. This transforms the problem into a more complex one by finding the best fitting assignment of persons and projects.

References

1. Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., Vesci, G.: Choosing the right crowd: expert finding in social networks categories and subject descriptors. In: Proc. of EDBT (2013)
2. Burkard, R.E., Cela, E.: Linear assignment problems and extensions. Springer (1999)
3. Heil, S., Wild, S., Gaedke, M.: CRAWL·E: distributed skill endorsements in expert finding. In: Casteleyn, S., Rossi, G., Winckler, M. (eds.) ICWE 2014. LNCS, vol. 8541, pp. 57–75. Springer, Heidelberg (2014)
4. Howe, J.: Crowdsourcing: A Definition. Crowdsourcing Weblog (2006)
5. Lee, C.S., Pan, C.Y.: An intelligent fuzzy agent for meeting scheduling decision support system. Fuzzy Sets and Systems **142**(3), 467–488 (2004)
6. Sen, S., Durfee, E.H.: A formal study of distributed meeting scheduling. Group Decision and Negotiation **7**(3), 265–289 (1998)