Planning Teams with Semantic Web Technologies

Susanne Richter (richter@zib.de)
Konrad-Zuse-Zentrum für
Informationstechnik Berlin
Unit Scientific Computing
Scientific Information Systems
Takustr. 7, 14195 Berlin
www.zib.de

Robert Tolksdorf (tolk@ag-nbi.de)
Freie Universität Berlin
Department of Mathematics
and Computer Science
AG Networked Information Systems
Königin-Luise-Str. 24/26, 14195 Berlin
www.ag-nbi.de

Abstract: When planning teams for projects with specific goals, employees of a company have to group together so well, that all necessary knowledge for conquering the project's challenges are met within the member's skills. A tool that facilitates semantic web technologies can support the team recruiter, who is responsible for chosing the members of the team, in terms of finding the most efficient combinations of the company's employees based on their expertises.

1 Introduction

With the rising interest in Semantic Web technologies the comprehensive semantic integration of data inventories and applications is on top of many company's agenda. Surveys like [Joh06] show that the deployment of semantic technologies can improve business process performance by up to 50%. The survey also points out that the introduction of small single applications for supporting major business or information workflows proves to be more efficient and achieves better results than a full semantic integration of all stored data. We will discuss the development of such a focussed semantic application which is to support companies by helping to plan project teams. For this purpose the creation of employee and project descriptions that are based on the concepts of a human resource ontology is mandatory. The proposed team planning algorithms respect criteria of optimal teams and select those members from the pool of employees that can fulfil the project descriptions as a group. Section 3.3 defines these criteria formally as they are used by our approach. The team recruiter is provided with a collection of group proposals that are ranked by their suitability, of which the final team can be chosen.

1.1 Motivation

Teamplanning differs from the more commonly used term teambuilding and its methods (like the "Forming – Storming – Norming – Performing – Adjourning" model of Tuckman [Tuc65]) in the task of selecting team members from a pool of employees by a given project description, which is to support the forthcoming process of teambuilding by a more efficient assembly. Teamplanning is therefore allocated very early in a project life cycle, right after the project's definition.

West [Wes03] discusses the aspects of teamwork as a coordinated working paradigm in companies: "Teams are the best way to enact organizational strategy [..] Team-based organizations with their flat structures can respond quickly and effectively in the fast-changing environments most organizations now encounter." Further arguments for well assembled teams are the boost of creativity and innovation as well as the delivering of results quicker and in a more cost effective way.

Deiters [DL00] states that the increasing importance of project work and the use of flexible teams require improved computational maintenance of these resources. Consequently, skill management would support the teambuilding process as well as knowledge transfer and education planning within a company. The base for finding adequate team members is competence management, which in bigger companies requires a permanently updated knowledge about the employee's capabilities. A team recruiter must know which employee takes care of what task at a certain time, what basic knowledge the employee has and what special knowledge has been acquired. Biesalski writes in [Bie06] that filling vacant project jobs is rarely based on a structured approach but more or less on personal networking and preferences. That increases the risk that professional criteria are elided and appropriate candidates are overseen. Even when a company makes use of a competence management process it stays a difficult task for the team recruiter to identify the essential necessary skills in many different working domains and to recognize the employees that best fulfil these conditions in a team. So teamplanning requires flexible ways to compare skills in a discrete manner. Our aim is to provide team member selection that bases on professional competences also known as "'hard skills"'. Modelling "'soft skills"' seems to be too difficult in this stage of developement.

Semantic Technologies can provide just this: Describe and compare the various dimensions involved in planning a team with knowledge. Ontologies can be used to describe thematic domains and to model skills and practice in a concise way. Algorithms can take advantage of this knowledge base to provide proposals of potential good teams with high quality. Miscellaneous related work also show that the process of teambuilding can benefit from a computer based solution. 1999 Jussi Stader and Ann Macintosh from AIAI (Artificial Intelligence Application Institute) at Universoty Edinburgh developed a knowledge management system [SM99] that provides the following functions: kill gap analysis, project team building analysis, recruitment planning and training analysis. The system's base are models for competence, employee, projects and organisation that are based upon taxonomies. The system generates three result lists where the first list contains employees with relevant skills. The second lists consists of trainees who can further educate in skills that are missing in the first list. Skills that neither appear in the first list nor can be educated by trainees of the second list build together the third list. Teamrecruiter then can select fitting employee.

The group around Norbert Gronau who holds the chair of Business Information Systems and Electronic Government at University Potsdam follows a process oriented approach. They developed KMDL (Knowledge Modeling and Description Language) [FSRG06]. It is a framework for modelling and integration of business processes and knowledge flow in organizations. To their opinion staffing should be executed with the aim of building well-balanced teams that preferably combine all needed skills for fulfilling the project's tasks. For that a semantic net is designed in the first step by hierarchically organized skills. Qualification profiles are stored durably in a repository. The selection of tasks to fulfill in a project is based on KMDL. The teambuilding algorithm calculates team compositions from these inputs. The collection and evaluation of the employee data, the assignment and evaluation of individual skills and data that are always up-to-date are estimated difficult by the authors. They finally conclude that it's up to every single employee's motivation to truly care for its data. Since data should also be controlled by a higher positioned person the employee is forced to insert truthful information in the system. This method is also considered in our work.

Bisalski [Bie06] while being employed by DaimlerChrysler AG, is dealing with establishing standard processes like teamplanning as part of a client-server based personnel application system. For that purpose he defines several competence profiles (e.g. job profiles) which are stored in a central human resource data warehouse (HRDW). The ontology retrieves all required data directly from the corresponding fields of the HRDW and creates a semantic superstructure on top of it. Bisalski identifies a higher risk for overseeing adequate candidates and a temporal risk in using an unstructured approach to recruiting. The goal of a system should therefore be the automatic discovery of appropriate employees and the assignment of project employments based on competence profiles. He uses the comparison of project requirements profiles to employee competence profiles by an aggregated measurement which is built up from four parts, of which only one results of a semantic comparison by using taxonomic similarity. The other measurements consider the under- and overfulfilment of project requirements.

2 Teams and projects

Katzenbach [JRK93] defines teams as the following: "'A team is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable."' He also writes that the focus on the performance of teams is a more important factor of success than communication or a good interconnection. This definition holds for every kind of team. Stock [SH08] distinguishes between different types of teams, depending on the factual and the relational level of the members. The factual level relates to the relevance of task-defined aspects in the team while the relational level provides information about the human interactions. A real team exists if both levels are equally high developed. In such a team cooperation and communication flow is properly adjusted and the participating members also support each other in personal issues. The algorithms for teamplanning should result in proposals for such efficient teams.

The German Institute of Norming (DIN) defines a project in its norm DIN 69901 as an enterprise that is mainly defined by its unique conditions, e.g. goals, timeline, financial or personal limitations. Litke [LK06] gives a more comprehensive definition for a project as a delimitable single enterprise with defined beginning and end which is innovative, risky (time-limited, ecomomically, technically) and complex, in its scope changing organisational requirements are faced and it is of high important to the company it is supported from. Project groups work together for short periods of time, are not domain specific and usually comprise a project leader and a set of members. These have diverse qualifications and originate from different departments of the company. Our work look at such teams for intraorganisatorical projects.

What makes teamplanning difficult is not only finding a single person for a vacant position, but to combine employees for a task in a way that they fulfil Katzenbach's definition and additionally fit in the criteria of optimal teams. West [Wes03] describes the problem the following: "It's about finding the magic mix, that blend of skills and experience that combines maturity, energy, determination and creativity. It's like a ballet when it happens".

3 Teamplanning Algorithms

In the following we devise algorithms that are based on knowledge about potential team members. The following profile types are defined for the calculations:

- Project Requirements Profile (PRP): Contains all relevant information on a project. These basic information like title and timeline, but also the skills which are required for achieving the project goals are attached, rated by the level of required expertise.
- Employee Competence Profile (ECP): Contains all relevant information for an employee like name, gender and age. Additionally, the employee's skills are contained in a rated representation.

The algorithms also make use of the following parameters:

- P-Skills: Set of skills which are needed to complete the tasks of the project.
- E-Skills: Set of skills which the employee is proficient in.

The algorithm's input consists of a PRP and the ECP of all available employees. They differ in what information is actually used from the profiles. The output of both algorithms is a list of rated possible teams. The semantic algorithms rely on a comprehensive consideration of as many aspects as possible on teamplanning, while taking the criteria for optimal teams into account. During the calculations they generate a skill-matrix with the rows containing the P-Skills and the columns the employees. The matrix calculates the highest similarities of all E-Skills of the selected employee to a given P-Skill. The following information is also included in the matrix:

- Normalized row sum: The professional value that an employee can – in average – provide to a project team. The normalized line sum is generated by calculating the average of a matrix line.

- Normalized column sum: The average value of that a certain skill is available in a project team. The value is calculated by creating the average of a matrix row.
- Functional value: The average over the professional values of all team members.

3.1 Basic semantic teamplanning approach

For the basic semantic teamplanning approach the skills from the PRP and the ECP are used. The corresponding levels however are not considered here, each skill is regarded as equally important. In a brief summary the algorithm takes the following steps, which is also pictured in 3.1:

1. Calculate the similarity of all P-Skills and E-Skills.
2. Store each highest similarity in the skill matrix (E,P).
3. Select members.
4. Rank.

The similarity calculation is done with the help of a human resource ontology between all P-Skills and all E-Skills[1]. Relevant is in each case only the highest similarity value of any E-Skill to the given P-Skill. Less similar skills are not used in the selection process because these do not significantly help on the execution of the equivalent project's task. An entry x in the skill matrix implies, that an employee E (row) does at least have one skill, which is with a value of x similar to a project skill P (column). The calculated similarities in the matrix lie in a range of 0 to 1.

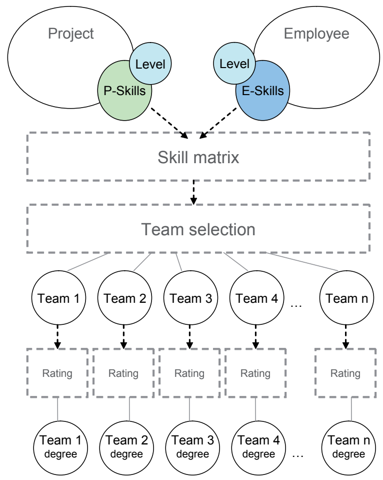

Figure 1: Schema of the basic semantic teamplanning algorithm

The usage of an ontology-based similarity measure is the specific feature of our approach. The ontology used consists of more than 10000 concepts which are related by regular is-a relations but also by knowledge on homonyms, synonyms and related concepts. On that basis, a similarity function calculates a measure amongst two concepts that is based on the mentioned relations. While teamplanning approach outlined below uses arithmetic calculations, the overall team-planning profits from the more precise semantically derived similarity measure which is the input to those calculations..

For the selection of team members for each P-Skill a number of up to two employees are chosen. These are the ones with the highest values in the P-Skill's row. This guarantees

[1]The ontology was provided by the company Ontonym GmbH, Berlin (www.ontonym.de).

that each of the P-Skills can be executed by at least two team members which provides for transactive knowledge. (Transactive knowledge is the knowledge about other people's knowledge and it is the base for discussions and problem solving in a team.) At the same time the chosen employees have to possess the highest normalized column sums compared to each other. This guarantees that such employees are chosen which are on the one hand the most qualified and on the other hand have a high average professional value.

The chosen set of employees for all project skills is permutated in a tree-structure to form a set of teams with 4 to 10 members. This approach takes into account two of the criteria for good teams in the definition of Katzenbach: complementary capabilities and low number of members. The whole set of possible teams is now rated by their suitability to the project by the appropriate mechanisms which are to be described later.

3.2 The rating semantic teamplanning approach

For the rating semantic teamplanning approach skills are used in a rated way from the PRP and the ECP. Additionally the experience and expertise of a skill are used while building the skill matrix. That also expresses the importance of certain P-Skills in the project because higher rated skills are more important than others. Analog to Biesalski [Bie06] we use four rating-levels: **1 - Student** (beginner's skills, that might not have been professionally trained), **2 - Junior** (extended basic knowledge from professional experience), **3 - Middle** (properly developed knowledge), **4 - Senior** (expert knowledge by long-term application). The rating algorithm takes the same steps as the first approach, only with the difference that the calculated similarities are downgraded by the relation of the given to the required level. The equation 1 indicates the calculation of the matrix' entries:

$$sim_{2.Algorithm} = \begin{cases} \frac{sim_{1.Algorithm} * Level_M}{Level_P}, & \text{if } Level_P \geq Level_M \\ sim_{1.Algorithm} * \left(1 - \frac{Level_M - Level_P}{\#Level}\right), & \text{if } Level_P < Level_M \end{cases} \tag{1}$$

$Level_P$ indicates the project level and $Level_M$ the employee level. The actual similarity between the skills, as it was shown in the first approach ($sim_{1.Algorithm}$) is being alleviated by the over- or underfulfilment of the necessary skill level. When a required skill is e.g. level 1 and the employee has level 4 he might not be able to bring his whole value into the project because only part of his knowledge is used. Additionally he is most likely more expensive to the company. He should therefore better be used in another project where his experience can be more profited from. The same goes for a underfulfilment. If a project requires a skill on level 3 and the employee only has this skill on level 1 this similarity value is accordingly reduced. The higher the difference between the necessary and the provided level, the more is the similarity value downgraded. In the case of overfulfilment, $sim_{1.Algorithm}$ has a value of more than 1 that is alleviated by $\left(1 - \frac{Level_M - Level_P}{4}\right)$ according to the difference of the levels. A higher difference results in a smaller similarity value than a lower difference. This way of calculation values between 0 and 1 are created in the matrix. Figure 3.2 shows an example of the skill matrix for the rating algorithm.

project / employees	skill A – L2	skill B – L4	normalized line sum
1	$\frac{0{,}8*1}{2}=0{,}4$	$\frac{0{,}7*4}{4}=0{,}7$	$\frac{1{,}10}{2}=0{,}55$
2	$\frac{1*2}{2}=1$	$\frac{0{,}2*3}{4}=0{,}15$	$\frac{1{,}15}{2}=0{,}58$
3	$1*\left(1-\frac{3-2}{4}\right)=0{,}75$	$\frac{1*2}{4}=0{,}5$	$\frac{1{,}25}{2}=0{,}63$
normalized row sum	$\frac{2{,}15}{3}=0{,}72$	$\frac{1{,}35}{3}=0{,}45$	functional value: 0,57

Figure 2: Example of the rating semantic teamplanning algorithm

For a given project the skills A and B are needed. The matrix entry 0.4 represents the skill of the employee 1 with the highest similarity to A and has a calculated similarity value of 80% from the first approach. It is now downgraded due to underfulfilment of the required level for skill A because the employee is only capable of level 1 even though level 2 is required. That results in a final value of 0.4. The entry 1 for employee 2 represents that the required skill A is available at the employee, even with the required level. Employee 3 also owns skill A but has the higher level 3 even though only level 2 is needed in the project. It is therefore overfulfilled and the value is downgraded to 0.67. The first algorithm calculated the same value for both employee 2 and 3 because it doesn't take the levels into account. The rating algorithm shows that for the given project the employee 2 is more suitable than employee 3.

For the selection process the higher leveled skills are seen as more important for the project and therefore handled with higher weight. Skills are sorted by their level. Starting with the maximum prioritised requirement at least two employees are chosen which have the highest skill matrix entries and at the same time the highest professional values. The same is done for the other P-Skills in decreasing order of their level. The set of employees is now permutated to groups of 4 to 10 employees. The prioritisation of the higher-level P-Skills causes the most valuable employees to appear closer to the root of the permutation tree. That guarantees them to be chosen first and not to be sorted out because of a too high employee count. After team generation, the rating process follows.

3.3 Rating of teams

The rating delivers, independent of a previously performed member selection, an estimate of the team's quality in terms of the criteria of optimal teams. This approach enables a performance comparison of different selection algorithms. Similar to performance appraisals in sports, each criteria is rated in relation to an ideal case. The non-fulfilment of details reduces the value piece by piece.

3.3.1 Team size

An optimal team consists of 4 to 10 members. Team sizes that are outside that range won't be proposed to the team recruiter. An optimal team should need preferably few personnel ressources for fulfilling its tasks. The rating will prefer a team with the same functional value over another if it has fewer members. Some members might have overlapping skills, in which case we call them "congruent". Congruent members result in a loss of competence since the task could be realised by fewer employees. That fact is implemented by equation 2.

$$Rate_{teamSize} = \frac{functionalValue + \frac{4}{numberOfTeamMembers}}{2} \tag{2}$$

Regarding team size the ideal case occurs by achieving the functional value 1 with four team members. Therefore the rating value of the team size is the average between the team's functional value and its size being normalized by four. Teams consisting of four members and having a functional value of 1 will get the highest possible value. The more members a team needs the more the fraction decreases. A team can only be highly rated if its functional value as well as its number of members are optimal.

3.3.2 Heterogenity

According to Katzenbach [JRK93] an optimal team should preferably consist of heterogeneous members because they are more productive than a homogeneous team. Heterogenity can be based on different attributes of the members. Here, we consider knowledge, expertise and experience. Therefore a team is defined as heterogeneous if there are no congruent members in respect to their skills. The calulation is shown by equation 3.

$$Rate_{heterogenity} = 1 - \frac{1}{numberOfTeamMembers} * numberOfCongruentPairs \tag{3}$$

A pair is formed by two congruent team members. Since only one member of such a pair is needed in the team its functional value will be reduced by one employee, metaphorically speaking. In case all members have the same skills the number of congruent pairs equals the number of team members minus 1.

3.3.3 Overlapping

Teamwork won't be successful if there is no transactive knowledge. When there are no overlapping skills there won't be discussions and in the following less valuable solutions might be found. Therefore there should be at least two persons representing one skill by accomplishing a high degree. Equation 4 also starts from an optimum case: for each P-Skill there should be two persons that fulfil the requirement with 100%.

$$Evaluation_{Overlapping} = \sum_{i=1}^{n} \left(\frac{\sum x + y}{2} * \frac{1}{n} \right) \quad (4)$$

The equation shows n as the number of P-Skills, x and y denote the two highest grades of expertise per column in the skill matrix. In the best case, $\frac{\sum x+y}{2}$ equals 1 for each column, while for each of the skills a part of $\frac{1}{Anzahl\_Faehigkeiten}$ is implied, which is in sum 1. Each skill which doesn't fulfil this alleviate the value for overlapping correspondingly.

3.3.4 Skill gaps

Skill gaps should be prevented in a team. According to Katzenbach [JRK93] the presence of a skill gap doesn't have to lead to a failure when the team members are willing to close that gap by learning. For Stader [SM99] there exists a skill gap when a skill is missing or not sufficiently available. For the criteria of skills gaps a 0 would mean the absence of a P-Skill and a value of 1 would be a 100 percent fulfilment. For that reason the value of the minorly fulfilled skill, ie. the smallest normalized column sum is returned here. See equation 5).

$$Evaluation_{Skillgaps} = smallest\_normalized\_rowsum \quad (5)$$

If the smallest normalized row sum is high (close to 1), there is nearly no gap. A very low value influences the whole rating as a skill gap.

3.3.5 Teamleader

Roles in a team, as those from Belbin [Bel04], are based on psychological personal profiles. The most significant role is that of the teamleader. It should be adopted by the person with the highest normalized row sum in the team. See equation 6.

$$Evaluation_{Teamleader} = highest\_normalised\_rowsum \quad (6)$$

The value of the team leader is also taken into account for the rating. The higher its value and therefore its suitability, the better a team is evaluated.

3.3.6 Availability

The availability of all team members over the whole project period is essential for a team. Each member has explicit tasks and the ommission of one of them can even endanger the fulfilment of the whole project. The developed algorithms already observe the availability at the member selection. The set of employees teams are built from is already reduced by such candidates that are not available. To have an independent rating, the availability is checked during that process anyway. Each non-available member weakens the team.

This circumstance is expressed by reducing the team's functional value by the value of the missing employee. See equation 7:

$$Evaluation_{Availability} = Functionalvalue - \sum_{i=1}^{n} normalised\_rowsum_{E_n} \quad (7)$$

In the equation n depicts the number of non-available team members. If none of the members is available, the value of 0 is adopted in the overall rating.

3.3.7 Degree of suitability

The individual values that were calculated in the rating process must be aggregated to an overall rating degree in a last step. At a first look the overall degree is the average value of all single degrees. When considering the criteria in terms of their importance for the team ist is obvious that they don't all have the same influence on the project's success. As an example we compare team size to skill gaps. While skill gaps can be reduced by learning, a too large team would separate into subteams or too small teams can't make use of synergies. For that reason the single criteria are used by priority:

1. Overlapping and transactive knowledge
2. Heterogenity
3. Availability
4. Team size
5. Team leader
6. Skill gaps

Since overlapping and transactive knowledge have a direct influence on teamwork by enabling discussions, this criteria has the highest importance. A heterogenous formation ensures a productive work and is therefore prioritised as second. These two criteria provide for a productive workflow. For keeping that state for the whole duration of the project all members have to remain available. This is followed by the tam size, where a range of seven sizes (4 - 10) is considered as appropriate. A competent team leader is keeping the team's dynamics and represents the team to the outside. Skill gaps meanwhile are the easiest to compensate and are therefore ordered as last. That creates the following equation for calculating the overall rating of a team:

$$Degreeofsuitability := \frac{v_1 * Rate_{Overlapping} + v_2 * Rate_{Heterogenity} + v_3 * Rate_{Availibility} + v_4 * Rate_{Teamsize} + v_5 * Rate_{Teamleader} + v_6 * Rate_{SkillGaps}}{\sum_{i=1}^{6} v_i} \quad (8)$$

The overall rating degree is therefore the aggregate of all prioritised single ratings. The real order of the criteria as well as the values of the vector v_i have to be optimized by a large number of real-world experiments with team recruiters. For our initial implementation, we used a value of V=(6,5,4,3,2,1).

4 Implementation

We have implemented our algorithms in a tool which is shown in figure 3. For the calculation of similarities `SimPack` was used, a "generic Java library for similarity measures in ontologies" [BKKB05]. For using these similarity measures a special *PersistentOntologyAccessor* was implemented, which provides the same functionality as other available accessors, but is applicable to a persistent ontology like the *HR-Ontologie* that we used. The developement of an specific similarity measure for planning teams wasn't in the focus of this work. For our first implementation we then used one measure that `SimPack` provides. It's the edge-counting method of Resnik [Res99]. The implementation uses the *COLT-API* for skill matrices, which was developed at CERN [Hos04]. It is based on a *SparseDoubleMatrix2D*, a two-dimensional matrix of decimals which supply effective methods for saving and retrieving information to and from the matrix. The completed matrices are also persisted for performance reasons and for later access. *JENA* is a Java framework for the development of semantic web applications [DS08] and is used as a programming interface for processing and storing the semantic data.

4.1 Complexity

Particularly when evaluating a possible deployment of an application in a big company, it must be questioned, how the implementation can handle higher requirements. The performance of the semantic algorithms depends mostly on the calculation of the skill matrices. The computation of such a matrix is done in the worst case by comparing each P-Skill and each E-Skill. The number of comparisons is reduced when exactly the wanted skill is present at the employee. To prevent a high number of accesses to the ontology the shortest paths between two nodes in the ontology are precalculated.The number of cells in a skill matrix in both algorithms is:

$$Cells_{SkillMatrix} = Count_{availableEmployees} * Count_{P-Skills} \quad (9)$$

In each row of this matrix the similarity is entered as a decimal double value in the rang of 0 to 1, which requires 8 Byte in Java.

Example: for a company with 30 of 50 total employees available, a skill matrix uses 1200 bytes of space for a project with five P-Skills for the storage of the comparison values. If an average employee has ten skills and the searched skill is not among these, exactly 10 comparisons are necessary. For the given 5 P-Skills in the worst case 50 comparisons are

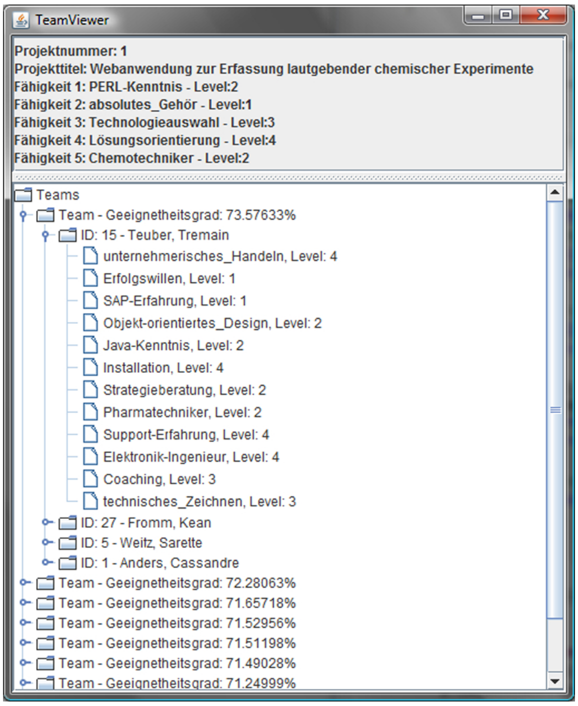

Figure 3: Our experimental GUI

calculated.

The set of relevant employees for the selection process is reduced by a constantly low number of m employees for each project skill. The constant m was initialized by 3 for testing purposes to provide some possible skill overlapping, but can be modified if necessary. These m employees can be selected by performing a linear operation on the matrix. The comparably small lists of relevant employees are permutated to the set of possible teams.

For a large inputs (e.g. many employees) this computation takes the longest part of the calculations, as was expected. However, a large number of already calculated matrices can be re-used efficiently. The rating semantic teamplanning algorithm differs in terms of complexity by considering an additional parameter when creating the skill matrix. Anyway, the effort is increased only by a negligible constant. During the selection, the consideration of the skill's priority also doesn't lead to a significant extra effort.

A further performance improvement could be reached by caching similarity calculations. That should make the reuse of already calculated lines of the skill matrix more simply available in other projects. The already calculated matrices could be used without doing another calculation. They just have to be adjusted first after employee skills have changed or second in case of ontology modifications.

5 Conclusion

Biesalski criticizes in [Bie06] that the project staffing of vacant positions is mostly based on personal networks and preferences. Disadvantages of that predominant method are faced by objective team staffing. That concerns the selection of available employees by automated processing their data, the objective comparison between project and employee profiles as well as an objective rating by using the team criteria.

Moreover Deiters writes in [DL00] how competence management systems with embedded team planning function should be designed. The tool presented here does just that by using an ontology to calculate semantic similarity. The skill matrix also allows other corporate surveys concerning the distribution of competences, the discovery of core competences or skill gap analysis. Therefor the developed concepts still have more potential and give room for further development.

For using the prototype in a corporate area it should be completed by functions concerning further use cases and a system specification. Generally the best available employees are choosen for the given project requirements. That approach has the advantage of building teams with higher degrees of suitability. But it also has the disadvantage that less qualified employees aren't considered. This constellation can happen in many projects one after another so that many employees are frequently choosen and others never. By a solution of [SM99] with trainees employees can educate themselves by working in projects.

The processing of skills allows a competence oriented arrangement of teams. Soft skills were not considered during the conceptual design because their objective recording seemed too difficult. But a suitable team arrangement is mostly based on character skills and their fitting combination in a team. For this reason further topic development by additional examination of soft skills for team planning seems to be meaningful.

The extension by a parallel generation of teams for different projects is also thinkable. Even in huge companies there is a increased fluctuation of projects which means that project work starts and ends at many different times. In that case the algorithms have to equitably distribute the available competences to several projects because otherwise some projects are discriminated in their performance.

Our approach to team planning ist based on the modelling of skills of potential team members. We use an ontology based measure of similarity between required skills and education and that provided by a personen. Since we used a detailed ontology as the basis, we claim that these similarity measures are very precise. We also claim that the results generated by our algorithms are of higher quality for this reason. Our first manual evaluation of the results have indicated that higher quality, a broader statistical evaluation is underway.

Acknowledgements

We thank the Ontonym GmbH, Berlin for providing their HR-ontology, and the Siemens AG, München for continuous support and the provision of real-world requirements and

experiences on manual teamplanning.

References

[Bel04] R. Meredith Belbin. *Management Teams. Why They Succeed Or Fail*. Butterworth Heinemann, Oxford, 2. edition, 2004.

[Bie06] Ernst Biesalski. *Unterstützung der Personalentwicklung mit ontologiebasiertem Kompetenzmanagement*. PhD thesis, Universität Fridericiana zu Karlsruhe, 2006.

[BKKB05] Abraham Bernstein, Esther Kaufmann, Christoph Kiefer, and Christoph Bürki. SimPack: A Generic Java Library for Similiarity Measures in Ontologies. Technical report, Department of Informatics, University of Zurich, 2005.

[DL00] Wolfgang Deiters and Reinhard Lucas. Skill-Management: ein Baustein für das Management flexibler Teams. *Information Management and Consulting*, 15(3):54–60, 2000.

[DS08] Ian J. Dickinson and Andy Seaborne. Jena - A Semantic Web Framework for Java, 2008. `http://jena.sourceforge.net/index.html`, see on 24.02.2009.

[FSRG06] Jane Fröming, Simone Schmidt, Uwe Rüßbüldt, and Norbert Gronau. Anforderungs- und wissensorientierte Zusammenstellung von Projektteams. *Industrie Management*, (22):12–14, 2006.

[Hos04] Wolfgang Hoschek. Colt - Open Source Libraries for High Performance Scientific and Technical Computing in Java, 2004. `http://acs.lbl.gov/~hoschek/colt/index.html`, see on 24.02.2009.

[Joh06] Michael John. Semantische Technologien in der betrieblichen Anwendung. In *Ergebnisse einer Anwenderstudie, Technical Report*. FIRST - Fraunhofer Institut Rechnerarchitektur und Softwaretechnik, September 2006. URL: `http://publica.fraunhofer.de/eprints/urn:nbn:de:0011-n-534743.pdf`, see on 20.11.2008.

[JRK93] Douglas K. Smith Jon R. Katzenbach. *The Wisdoms of Teams*. Havard Business School Press, Boston, 1993.

[LK06] Hans-D. Litke and Ilonka Kunow. *Projektmanagement*. Haufe Verlag, 5. edition, 2006.

[Res99] Philip Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Applications to Problems of Ambuguity in Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[SH08] Ruth Stock-Homburg. *Personalmanagement: Theorien - Konzepte - Instrumente*. Gabler Verlag, 2008.

[SM99] Jussi Stader and Ann Macintosh. Capability Modeling and Knowledge Management. Technical report, Artificial Intelligence Applications Institute, Division of Informatics, The University of Edinburgh, 1999.

[Tuc65] Bruce Wayne Tuckman. Developmental sequence in small groups. *Psychological Bulletin*, (63):384–399, 1965.

[Wes03] Michael A. West. *Effective Teamwork; Practical Lessons from Organizational Research*. Blackwell Publishers, 2. edition, 2003.