TEAMBROKER: CONSTRAINT BASED BROKERAGE OF VIRTUAL TEAMS

Achim P. Karduck
Department of Computer Science in Media, Furtwangen university of applied sciences
Robert-Gerwig-Platz 1, 78120 Furtwangen, Germany

Amadou Sienou
Department of Computer Science in Media, Furtwangen university of applied sciences
Robert-Gerwig-Platz 1, 78120 Furtwangen, Germany

Keywords: Computer supported team formation, resource allocation, constraints satisfaction optimization

Abstract: Some consulting projects are carried out in virtual teams, which are networks of people sharing a common purpose and working across organizational and temporal boundaries by using information technologies.
Multiple investigations covering these teams focus on coordination, group communication and computer supported collaborative work. However, additional perspectives like the formation of teams are also important. Here one should deal with the question: how to form the best team?
To approach this question, we have defined team formation as the process of finding the right expert for a given task and allocating the set of experts that best fulfills team requirements. This has been further transformed into a problem of constraint based optimal resource allocation.
Our environment for computer supported team formation has been developed by having adopted the brokerage view that consists of mediating experts between peers requesting a team and the ones willing to participate in a team. Computer supported brokerage of experts has been realized as a distributed problem solving that involves entities representing experts, brokers and team initiators.

1 INTRODUCTION

Let us suppose, *SAM & associates systems, Inc.* is an organization of experts, member of a worldwide heterogeneous network of consultants. This company intends to develop a knowledge management system. For this purpose, it decides to form a virtual team of experts.

Because of the "virtualness" of the team, members may not know each other. Therefore, the decision to engage a candidate will be based on few factors like expertise, knowledge and cost. After the identification of the factors, the problem of forming the team can be solved by evaluating first the outcome of each candidate related to these factors, then selecting the candidates with the highest score. However, the same procedure is launched whenever an expert revokes his application. In this case a new team should be formed. What about a computational support to the formation of the team?

Basically, virtual teams are networks of people bound by a common purpose, who work across organizational and temporal boundaries in a collaborative environment supported by information technologies (Lipnack and Stamps, 1997) like Computer Supported Collaborative Work (CSCW) systems. Some CSCW systems are optimized in order to support multiple facets of virtual teaming with features like communication and document management, timetabling a.s.o. Teaming virtually seems to be interpreted as the operation in CSCW whereby the process anterior to team working is neglected. We exactly aim to support this process.

2 FORMING VIRTUAL TEAMS

2.1 What is team formation

A virtual team, like any team, goes through the phases forming, storming, norming, performing and adjourning (Lipnack and Stamps, 2000; Tuckman, 1965; Tuckman and Jensen, 1977) to get performed while producing results. During the first step, "forming", members are brought together for the first time. Focusing on this, the question "how does the team emerge to enter the forming stage" is justified. The step anterior to the "forming" stage is what we call team formation. It consists in the identification and

P. Karduck A. and Sienou A. (2004).
TEAMBROKER: CONSTRAINT BASED BROKERAGE OF VIRTUAL TEAMS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 146-153
DOI: 10.5220/0002604401460153

the selection of candidates for each activity of the project in question. The process of team formation is subdivided into the steps (Deborah and Nancy, 1999) summarized as follows:

1. **Requirements definition.** Identification of potentials and definition of requirements for experts' profiles.
2. **Candidate identification.** Exploration of the search space to identify experts who seem to meet the requirements.
3. **Candidate analysis.** Multidimensional Analysis of experts's profiles, the generation of alternative teams and the evaluation of their performance values. Here, the question that really matter is to find out who the best candidates for a team are.
4. **Contact establishment.** Subsequent to the evaluation of alternative teams, one will select the best team and contact members.

The dimensions of candidate analysis are measurable criteria that affect the team work. Since the focus is on the formation phase, factors essential to start the "forming" stage of team development are those that really matter. In the literature, some factors have been empirically evaluated or applied to other teams (Anderson, 1996; Deborah and Nancy, 1999; Lipnack and Stamps, 2000; Tuckman and Jensen, 1977; Schutz, 1955). In previous investigations we have considered the following factors necessary in order to start a team.

Table 1: Factors affecting team formation.

Factors	Description
Interest	What the members desire
Competencies	Skills and experiences of members
Risk	The risk of having a member
Availability	When are members available
Commitment	Deliberate attachment to the team
Budget	Amount available for the project
Project constraints	Cost constraints

Certainly these factors are interdependent. However competency is the most complex one, which affects the rest. It is therefore an object of a deeper analysis.

2.2 Conceptualizing competency

Competency or *"a specific, identifiable, definable, and measurable knowledge, skill, ability and/or other deployment-related characteristic (e.g. attitude, behavior, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context"* (Chuck Allen (ed.), 2001) has following basic properties:

- **Measurability and scales.** Although competencies are generally measurable, some are difficult to quantify because of their nature or the complexity of the metrics.
- **Context/Taxonomy.** Taxonomies are semantical descriptions of competencies, recursions, implications and equivalence relations between classes. In the scope of our investigation the concept of taxonomy is a model describing skills, levels, positions, equivalence and implication.
- **Recursion (inclusion).** Competencies are not always expressed as single values; they may include or extend other measurable competencies.
- **Equivalence and implication.** In a given context, there are equivalence and implication relations between competencies. Equivalent competencies are those that give evidence of semantically identic competencies without using lexically identic expressions. Implication is a relation between two competencies expressing that the semantic meaning of the first competency implies the one of the seconde.
- **Attributes.** Competencies are described with multiple attributes like "evidence" and "weights" which express its existence or sufficiency and its relative importance respectively.

In order to compute competency, it has been necessary to conceptualize the even described model. Following (Deborah and Nancy, 1999) we have organized a competency in a three layered structure consisting of *area of competence, knowledge* and *knowledge item*. Here the area of competence is a wide context of knowledge. Knowledge items are single attributes specifying a given knowledge in a real-life domain.

A skill is a tuple (A, B, C) where A is the area of competence, B the knowledge, C the knowledge item.
A competency owned by an expert is the tuple (A, B, C, ℓ, e) where ℓ is the level of skill and e the experience expressed in number of months.
A competency required for a task is a tuple $(A, B, C, \ell^{min}, e_{min}, \omega_\ell, \omega_e)$ where ℓ^{min}, e_{min}, ω_ℓ, ω_e are the minimum level of skill, the minimum experience required, the weight of the level of skill and the weight of the experience respectively.
The competency (A, B, C, ℓ, e) of an expert is valid regarding to a given requirement $(A', B', C', \ell^{min}, e_{min}, \omega_\ell, \omega_e)$ if the skills (A, B, C) and (A', B', C') match and the constraints $\ell \geq \ell^{min}$ and $e \geq e_{min}$ hold.

Since interests are also competencies, we have adopted similar representations for them, i.e an interest is a tuple (A, B, C, ℓ) where ℓ is the level of interest.

Based on these concepts of team formation and the conceptualization of factors affecting the formation process, we have developed the TeamBroker system to support the formation of virtual project teams. In the literature, investigations cover just the preconfiguration of teams and workflow systems. In our research project, we have adopted a brokerage approach which consists in the brokerage of optimal solutions (set of experts) to a constraint based specification of a project.

3 TEAMBROKER

3.1 A model for team formation

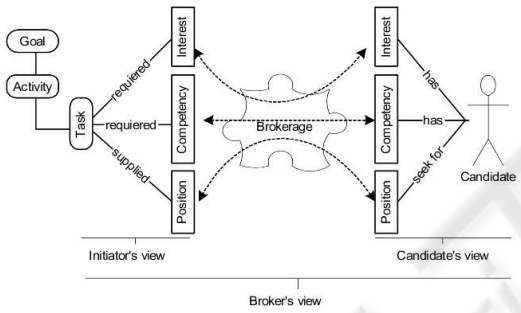

Figure 1: A model for team formation.

According to (Pynadath et al., 1999; Lipnack and Stamps, 2000; Petersen and Gruninger, 2000), a team is defined in order to execute activities which are grouped around a goal. In figure 1 we have refined the model from (Petersen and Gruninger, 2000) by subdividing activities into single tasks able to be carried out by single experts. The problem of assigning tasks is therefore simplified to single resource allocation which is more tractable than the allocation of multiple resources (Bar-Noy et al., 2001).

A task, in order to be performed, requires a position, a set of competencies and interests. Candidates are entities with interests and competencies who are looking for positions. Team brokerage consists of finding candidates for a task so that the positions, the competencies and the interests match (doted arrows). In this model, the team initiator defines tasks and requirements while experts provide information concerning their interests, their competencies and the positions in which they are interested.

Here, allocating a set of resources (experts) to the set of tasks defined in the context of a project is viewed as a Resource Allocation Problem (RAP). There are different approaches to solve RAP. Constraint programming, a framework used to solve combinatorial problems, is one of the successful approaches to these problems (Tsang, 1993). Key concepts of constraint programming are variables, their domains and constraints. Here, constraints are restrictions on the values to which a variable may be instantiated. The goal is to find an instantiation of all variables that satisfy all constraints.

Team formation is the Constraint Satisfaction Problem (CSP) (Z, D, C) specified as follows:

1. $\mathbf{Z} = \{\mathbf{z_1}, .., \mathbf{z_n}\}$ is the set of variables, each standing for the task to which experts should be allocated.
2. $\mathbf{D} = \{\mathbf{d_1}, .., \mathbf{d_n}\}$ is the set of domains of values to which variables z_i can be instantiated. Let us call elements of d_i instances indexed I which are conceptualized as follows:
   - **Budget.** Amount of money planned for the task to be assigned; indexed $B(I)$.
   - **Cost.** A value representing the cost of assigning a task to an expert; indexed $C(I)$. We define the cost as follows:
   $$C(I) = h_w(I) \times d(I) \times h_d(I) \qquad (1)$$
   Where $h_w(I)$ is the hourly wage, $d(I)$ the duration in days and $h_d(I)$ the number of work hours per day.
   - **Level of Commitment.** We refer to the aspect of commitment from (Petersen and Divitini, 2002) as the *level of commitment* expressed in the term of *commitment breaking cost*, which is the amount that a partner will have to pay if he leaves the organization before the achievement of goals. The level of commitment is indexed $L(I) \in [0, 1]$.
   - **Performance.** A value, indexed $P_{instance}$ that expresses the performance value of the instance.
3. $\mathbf{C}$ is a set of constraints in Z and D. We have categorized them into availability constraint, position constraint, instance constraints and team constraints.
   - **Availability constraint.** An expert is eligible for a team if he/she is available during the executing time of the tasks for which he/she is applying.
   - **Position constraint.** An expert applying for a task will be considered only if the position posted in the context of this task is the one that he/she is looking for.
   - **Instance constraints.** These constraints are restrictions to the level of interest, level of skill and experience. An expert is considered having an interest, skill or experience only if his/her levels

of skills or experience are at least equal to the level defined in the constraints.

- **Team constraints.** In contrast to instance constraints where only single experts are taken into consideration, team constraints affect the whole team.
  Let us introduce a binary variable $x_I \in \{0,1\}$ expressing whether an instance is activated; i.e. if the task z_i is assigned to a given expert and the resulting assignment is indexed I, the value $x_I = 1$; otherwise $x_I = 0$. A team is a n-dimensional vector the components of which are the indexes of activated instances.
  Based on the properties of single resource allocation we have defined the following constraints:
  $\mathbf{C_1}$. The total budget planned for a project should not be exceeded:

$$\sum_{i}^{n} \sum_{I \in z_i} C(I) \times x_I \leq \sum_{i}^{n} \sum_{I \in z_i} B(I) \times x_I \quad (2)$$

  For convenience, let us define the quotient

$$\Delta_{budget} = \frac{\sum_{i}^{n} \sum_{I \in z_i} C(I) \times x_I}{\sum_{i}^{n} \sum_{I \in z_i} B(I) \times x_I} \quad (3)$$

  and specify this constraint as follows:

$$\Delta_{budget} \leq 1 \quad (4)$$

  $\mathbf{C_2}$. All tasks must be assigned and each only once:

$$\forall z_i \in Z, \quad \sum_{I \in z_i} x_I = 1 \quad (5)$$

  Like any constraint satisfaction problems, this one will also have one, none or multiple solutions $X = \langle I_1, ..., I_n \rangle$. A solution is valid if both constraints C_1 and C_2 hold. The solution space has a size of $\prod_{i=1}^{n} |d_i|$. Therefore, we need to support the selection of a team by introducing the concept of team performance value P, which maps each solution X to a value $P(X)$. The value $P(X)$ depends on single performance values of the experts involved in the team X.

3.2 Evaluating performance values

Selecting an expert for a given task is a decision problem depending on multiple criteria. The decision is based on the performance of an instance which is an aggregate value of the factors cost, synergy, skills, experiences, commitment and risk. By using the technique of objectives hierarchies (Keeney, 1992) in order to transform these factors, we have eliminated interdependencies and have obtained the following criteria:

- $\mathbf{\Delta_{cost}}$**.** There are many candidates for a given task. The cost of assigning this task to an expert indexed i is $C(i)$. Let $C(max)$ be the cost of assigning the task to the most expensive expert. Δ_{cost} is the standardized value of the deviation of $C(i)$ from $C(max)$; i.e. how expensive is an expert compared to the worst case.

$$\Delta_{cost} = 1 - \frac{C(I)}{C(max)} \quad (6)$$

- **Level of commitment.** Value defined in the previous section as $L(I)$.
- **Synergy.** We have defined the *value of synergy* as the similarity V_s of candidate interests to the interests required for the task in question.
  Let $S^{task} = \{(A_i, B_i, C_i, \ell_i^{min}), i = 1...n\}$ be the set of interests required for a given task;
  $\Omega = \{\omega_i,\ \omega_i \in [0,1],\ \sum_{i=1}^{n} \omega_i = 1,\ i = 1...n\}$ be a set of weighting factors representing the relative importance of interests;
  ℓ_i^{min} be the minimum level required for each interest ;
  $S = \{(A_i, B_i, C_i, \ell_i), i = 1...n\}$ be the set of experts' interests.
  The vector X^{expert} representing the computed values of an expert's interest is defined as follows:

$$X^{expert} = \langle ..., \ell_i \times \omega_i \times \tau_i, ... \rangle, \quad (7)$$

$$\tau_i = \begin{cases} 1 & \ell_i \geq \ell_i^{min} \\ 0 & \ell_i < \ell_i^{min} \end{cases}, \quad i = 1...n \quad (8)$$

  Here, ℓ_i is the expert's level of the i^{th} interest.
  The value of synergy is finally defined as follows:

$$V_s = 1 - \frac{\sqrt{\sum_{i=1}^{n} \left(X_i^{ideal} - X_i^{expert}\right)^2}}{\sqrt{\sum_{i=1}^{n} \left(X_i^{ideal}\right)^2}} \quad (9)$$

  Here X^{ideal} represents the virtual ideal expert according to this factor:

$$X^{ideal} = \langle ..., \ell^{max} \times \omega_i, ... \rangle, \quad i = 1...n \quad (10)$$

- **Competency (skills and experience).** The execution of a task requires a set of skills and experiences. Since experience depends on skills, we have introduced the concept of Competency to explain the aggregate value of both.
  Let $S^{task} = \{(A_i, B_i, C_i, \ell_i^{min}, expe_i^{min}),$
  $i = 1...n\}$ be the set of skills and experiences required for a task.
  $\Omega^s = \{\omega_i^s,\ \omega_i \in [0,1],\ i = 1...n\}$ be a set of weights representing the relative importance of skills.
  $\Omega^e = \{\omega_i^e,\ \omega_i \in [0,1],\ i = 1...n\}$ be a set of weights representing the relative importance of experiences.
  The vector X^{expert} represents the computed values

of the skills of an expert:

$$X^{expert} = \langle ..., \ell_i \times \omega_i^s \times \tau_i, ... \rangle, \quad (11)$$

$$\tau_i = \begin{cases} 1 & \ell_i \geq \ell_i^{min} \\ 0 & \ell_i < \ell_i^{min} \end{cases}, \quad i = 1...n \quad (12)$$

The vector Y^{expert} represents the computed values of the corresponding experiences.

$$Y^{expert} = \langle ..., expe_i \times \omega_i^e \times \tau_i, ... \rangle, \quad (13)$$

$$\tau_i = \begin{cases} 1 & expe_i \geq expe_i^{min} \\ 0 & expe_i < expe_i^{min} \end{cases}, \quad i = 1...n \quad (14)$$

The fused view of skills and experience is the matrix $C = \langle X_i, Y_i \rangle$. The aggregate value of expert's competencies is processed with the vector $Z^{expert} = \langle \|C_1\|, ..., \|C_i\|, ..., \|C_n\| \rangle$. Let Z^{ideal} be the virtual expert with the highest possible values of competencies. The aggregate value of an expert's competencies is the similarity of his/her competencies to the ones of the ideal virtual expert:

$$V_c = 1 - \frac{\sqrt{\sum_{i=1}^{n} \left(Z_i^{ideal} - Z_i^{expert}\right)^2}}{\sqrt{\sum_{i=1}^{n} \left(Z_i^{ideal}\right)^2}} \quad (15)$$

Let us consider the following table illustrating the levels of skills and experiences of 3 experts candidates for a task requiring 2 skills. Since $V_{c2} = 0.996 > V_{c1} = 0.819 > V_{c3} = 0.667$, we conclude that considering this factor, $expert_2$ is the best one for this task.

Table 2: Sample competencies.

Requirement	skilled experts					
	$expert_1$		$expert_2$		$expert_3$	
	ℓ	$expe$	ℓ	$expe$	ℓ	$expe$
$competency_1$	3	30	3	36	4	36
$competency_2$	4	20	3	25	3	12
$competency_1$ =programming,java,2,24,60,45						
$competency_2$ = programming,ejb,2,6,40,55						
level: none=0, 1=low, 2=medium, 3=high, 4=expert						

Let $\omega_i, i = 1...4$ be weighting factors representing the initiator's relative preferences for the criteria Δ_{cost}, V_s, V_c and $L(I)$ respectively. Since the factors are preferential independents and the values are standardized, the weighted sum aggregation procedure is applicable to evaluate the performance $P_{instance}$ of an instance I as follows:

$$P_{instance}(I) = \sum_{i=1}^{4} \omega_i \times I_i^{value} \quad (16)$$

where

$$I_{i=1..4}^{value} = \langle \Delta_{cost}, V_s, V_c, L(I) \rangle$$

This value expresses "how well" the profile of an expert fulfills the requirement specification of a given task. For each task, the expert having the highest score is the best one.

Given that it was possible to order experts interested to tasks, it is now necessary to find the best constellation of experts. For this purpose, one will refer to the team performance value $P(X)$ which is the weighted sum of the utility of assigning single tasks to experts:

$$P(X) = \sum_{i=1}^{n} \omega_i \times \sum_{I \in z_i} P_{instance}(I) \times x_I \quad (17)$$

Here $\omega_i \in [0, 1]$ is a weight representing the relative importance of the instance I and $\sum_{i=1}^{n} \omega_i = 1$.

3.3 Searching the best team

At this stage of the formation process, the main problem to deal with is to find the team with the highest utility so that all tasks are assigned to exactly one expert and the total cost does not exceed the total budget. This is a single resource allocation and Constraint Satisfaction Optimization Problem (CSOP) defined as follows (see eq. 17 for $P(X)$):

$$\begin{array}{ll} find & X = \langle I_1, ..., I_n \rangle \\ s.t. & maximize \ P(X) \\ subject\ to & \\ & \Delta_{budget}(X) \leq 1 \\ & \forall z_i \in Z, \ \sum_{I \in z_i} x_I = 1 \\ & \forall I \in z_i, \ x_I \in \{0, 1\} \end{array} \quad (18)$$

Since it is algorithmically possible to satisfy the last constraint ($\forall z_i \in Z, \ \sum_{I \in z_i} x_I = 1$) by instantiating each variable exactly once, this one is no longer relevant to the solution if the control of the algorithm forces single allocations. The objective consists henceforth in maximizing the team utility without exceeding the total budget.

Note that $\forall I \in z_i, i = 1...n, \omega_i \geq 0$ and ω_i is constant; i.e. the weighting factor of each task is positive and constant during the formation process. Since $\sum_{I \in z_i} P_{instance}(I) \times x_I \geq 0, \ \forall \ I \in z_i$, the value $P(X)$ is maximal if the values $P_{instance}(I)$ are maximal; i.e. the team performance is maximal if tasks are assigned always to experts having the highest performance value. The set of the best experts is however not necessarily a valid solution because the constraints must be satisfied.

Let us suppose that the experts are ranked according to decreasing order of the values of $P_{instance}$. In case that the best team is not a valid solution, one will examine the next best team. This one is formed by replacing exactly one candidate for a task by the seconde best one for the same task. This process is

a 1-flip operation used to expand the search space. In order to find the best team without executing an exhaustive search of the space, we have developed a search strategy extending the iterated hill-climbing search. The hill-climbing search is a local search strategy simulating a hill climber, who aims to reach the peak of a landscape by iteratively jumping from an initial peak to the peak of the neighborhood until he reaches the maximum (Michalewicz and Fogel, 1999). Hill-climbing strategies are confronted with the problem of local maxima. To deal with this problem, we have adopted the tabu approach that consists of recording all visited non-solution nodes in order to avoid them in the future. In order to select always the best team, we have adopted an approach similar to the best search strategy by defining an open-list to store the non-visited neighborhood of a node. The algorithm always selects the best state of the open-list and expands the space by executing the 1-flip operator until a solution is found or the whole space has been processed. The flip-process guarantees to select the next best expert applying for the task in question.

After the definition of the model, the metrics and the search strategy, it has been necessary to develop a technical environment that supports the concept.

3.4 Technical realization

Figure 2 is the architecture of a technical environment supporting our model of team formation. TeamBroker, TeamCandidate, TeamInitiator are Java RMI (Remote Method Invocation) (Sun Microsystems, 2002) servers.

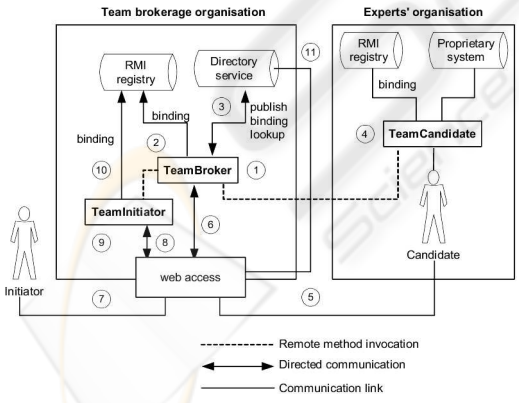

Figure 2: Architecture of the team brokerage system.

The business of the team brokerage organization consists in forming teams of experts that fulfills requirements defined by initiators. It runs TeamBroker (1) RMI servers. Services provided by TeamBroker are published in a directory server (3). Using this information, experts identify and select (5) the suitable TeamBroker to which they send requests for registration. Upon reception of these requests, TeamBroker stores the reference of the server into its directory (3) by using the RMI registry service provider for JNDI (Java Naming and Directory Interface) (Sun Microsystems, 1999). When needed the suitable candidate is searched in the directory. Team initiators and experts use a web interface (7) to search, select and interact with a TeamBroker.

Experts' organizations are networks of experts looking for positions in virtual teams. Each of them runs a TeamCandidate (4) server, which is extensible to wrappers able to convert profiles described in proprietary formats into the one used in TeamBroker.

The initiator is an organization asking for virtual teams. It is responsible for the specification of requirements that the team should fulfill. Initiators access the brokerage system by using a web-interface (7,8,9). TeamInitiator is a RMI server that represents a human initiator. For each initiator requesting a registration, the system starts a TeamInitiator (9), which is bound (10) to the naming service of the broker.

At this level, team formation is an interaction of TeamBroker, TeamCandidate and TeamInitiator as shown in the sequence diagram of figure 3.

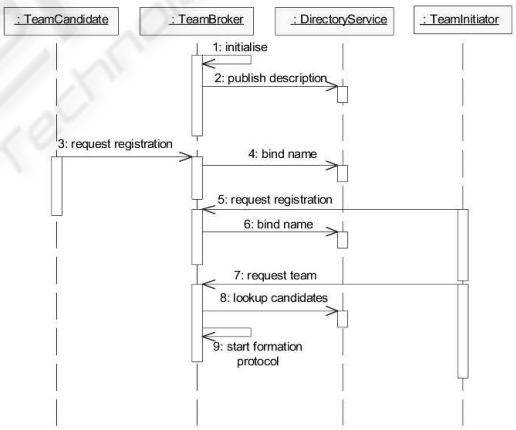

Figure 3: Sequences of team brokerage.

1. A TeamBroker starts running
2. Publication of services into a directory server
3. A TeamCandidate requests registration and supplies its JNDI name.
4. TeamBroker binds the JNDI name of the TeamCandidate to its own directory service.
5. Initiators use the web-interface to request for registration. An instance of TeamInitator is started to forward the request to TeamBroker.
6. TeamBroker binds the TeamInitiator to its directory service.

7. TeamInitiator requests a team by sending a message to TeamBroker.
8. TeamBroker queries the directory for RMI names of TeamCandidate servers which are bound in the naming context of the position defined in initiator's request for the team.
9. TeamBroker connects finally to the RMI servers listed in the result of the previous query and starts the team formation protocol of figure 4.

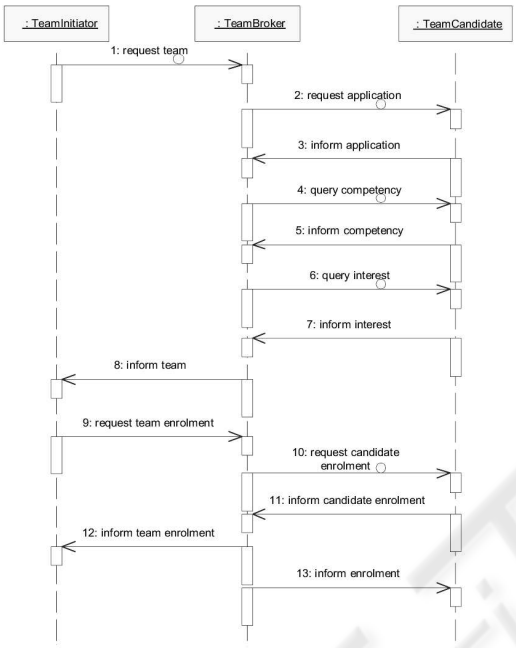

Figure 4: Team formation protocol

The team formation protocol is a set of communication messages shared by TeamBroker, TeamInitiator and TeamCandidate. As outlined in figure 4, it consists of the following six steps:

- **Request team (1).** The TeamInitiator requests the TeamBroker to form teams which fulfills requirements defined in the content of the message. The message contains the list of required positions, schedule, constraints, skills, experiences, interests and budgets.
- **Request application (2)/Inform application (3).** The TeamBroker identifies all potentially interested experts by searching in the directory. In a next step, it requests TeamCandidate to explicitly apply to the position within a deadline. The message addressed to the experts contains information concerning the position and the schedule. Instances of TeamCandidate answer by sending an application message. This message contains the state of the application, the hourly wage and the level of commitment of the expert. The state of the application is either "accepted" or "rejected". If the state is "accepted" TeamBroker continues the formation process with the TeamCandidate. Otherwise this one is no longer considered as a candidate.
- **Query competency (4)/Query interest (6).** During this process, TeamBroker communicates with instances of TeamCandidate having accepted the application by sending an "inform application" message qualified "accepted". TeamBroker queries instances of interested TeamCandidate for level of competencies and experiences. The responses are "inform competency (5)" and "inform interest (7)" respectively.
- **Inform team (8).** At this stage of the formation protocol, TeamBroker has collected all information necessary to form teams. The result (teams and performances) is sent to the TeamInitator.
- **Request team enrollment (9)/Inform team enrollment (12).** TeamInitiator selects the best team and requests TeamBroker to enroll it.
- **Request candidate enrollment (10)/inform candidate enrollment(11).** TeamBroker requests single experts to confirm the enrollment. When an instance of TeamCandidate receives this message, the expert represented by this instance should decide whether to join the team or not. If all members agree in the enrollment, the team is definitively formed and all entities (initiator and candidates) are informed about the success. Otherwise, a fail message is broadcasted (13).

4 RELATED WORK

Works related to our project are the ones from (Rub and Vierke, 1998) and (Petersen and Divitini, 2002; Petersen and Gruninger, 2000). In contrast to the first concept which supports the configuration of virtual enterprizes, we have emphasized the formation of virtual teams of humans. Both concepts share the aspects of optimal resource allocation.
Unlike the second approach, where the virtual team initiator is the entity processing partners' outcome and evaluating their utilities, we have adopted a brokerage approach. In our project, the formation process is a behavior of a configurable mediator called broker, which is able to use multiple taxonomies of knowledge description and different metrics to appraise candidates and form teams for the initiators. Furthermore, we suppose that there is no negotiation between the peers. In contrast to (Petersen and Divitini, 2002; Petersen and Gruninger, 2000), our performance metrics are based on the definition of non-interdependent criteria.

5 CONCLUSION

In our research projet, we have conceptualized factors affecting the formation of teams by defining models and metrics able to evaluate experts and teams. Based on this conceptualization of performance values and the formalization of constraints imposed by the project that a team has to carry out, we have transformed the problem of forming teams into a resource allocation problem. We have solved the resulting RAP by extending the iterated hill-climbing search strategy.

The TeamBroker system has been realized as a distributed system consisting of Java RMI servers. The successful application to our scenario has led to the conclusion that it supports the formation of virtual teams. However, the specification of a concrete project has to suit basic assumptions concerning (1) the structure of the criteria and competencies, (2) performance metrics, (3) aggregation procedures and (4) constraints.

As stated in (Deborah and Nancy, 1999) there are different types of virtual teams. Since for project or product development teams activities are clearly defined in form of technical requirements with fixed duration and measurable results, the TeamBroker system aims to support the formation of this kinds of teams. It is necessary to note that the system can also support the formation or pre-formation of non-virtual product development teams when rational measurable competencies of members are more important than emotional aspects.

In the future we intend to use advanced techniques of knowledge representation and processing in order to handle high inter-related skills.
In contrast to the current system, where the type of constraints are static, in our future work, we intend to assist team initiators by enabling them to add interactively new constraints to the system and to parameterize the resolution strategy by supporting for example partial constraints satisfaction.
We plan to integrate the TeamBroker system into a CSCW environment by adopting a service oriented architecture. This integration should support an automatic tracking of skills used by members while working in the CSCW environment.

REFERENCES

Anderson, W. (1996). Human resource development challenges in a virtual organization. In *IEMC Proceedings: Managing the Virtural Enterprise*, Vancouver, B.C.: IEMC.

Bar-Noy, A., Bar-Yehuda, R., Freund, A., Naor, J., and Schieber, B. (2001). A unified approach to approximating resource allocation and scheduling. *Journal of the ACM (JACM)*, 48(5):1069–1090.

Chuck Allen (ed.), H.-X. C. (2001). Competencies 1.0 (measurable characteristics). http://www.hr-xml.org [15.10.2002 20:00].

Deborah, L. D. and Nancy, T. S. (1999). *Mastering virtual teams : strategies, tools, and techniques that succeed*. Jossey-Bass Pub, San Francisco.

Keeney, L. R. (1992). *Value-focused thinking: a path to creative decision making*. Harvard University Press.

Lipnack, J. and Stamps, J. (1997). *Virtual Teams - Reaching across space, time and organizations with technology*. John Wiley & Sons.

Lipnack, J. and Stamps, J. (2000). *Virtual Teams*. John Wiley & Sons, 2 edition.

Michalewicz, Z. and Fogel, D. B. (1999). *How to solve it: modern heuristics*. Springer Verlag.

Petersen, S. A. and Divitini, M. (2002). Using agents to support the selection of virtual enterprise teams. In *Proceedings of Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002)*, Bologne, Italy. AAMAS 2002.

Petersen, S. A. and Gruninger, M. (2000). An agent-based model to support the formation of virtual enterprises. In *International ICSC Symposium on Mobile Agents and Multi-agents in Virtual Organisations and E-Commerce (MAMA'2000)*, Woolongong, Australia.

Pynadath, D. V., Tambe, M., Chauvat, N., and Cavedon, L. (1999). Toward team-oriented programming. In *Agent Theories, Architectures, and Languages*, pages 233–247.

Rub, C. and Vierke, G. (1998). Agent-based configuration of virtual enterprises. In Holsten, A. e. a., editor, *Proc. of the Workshop on Intelligent Agents in Information and Process Management KI'98*, volume 9.

Schutz, W. (1955). What makes groups productive? *Human Relations*, 8:429–465.

Sun Microsystems, I. (1999). Java naming and directory interface, application programming interface (jndi). http://www.java.sun.com/jndi [10.10.2002 09:00].

Sun Microsystems, I. (2002). Java remote method invocation specification. http://www.java.sun.com/rmi [05.08.2002 18:00].

Tsang, R. (1993). *Foundations of constraint satisfaction*. Academic Press.

Tuckman, B. and Jensen, N. (1977). Stages of small-group development revised. *Group and Organizational Studies*, 2(4):419–427.

Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological Bulletin*, 63:384–389.