# Resource Allocation in Software Projects Using a Bio-inspired Model

Anthony Karageorgos[1*], Elli Rapti[2], Vassilis C. Gerogiannis[3]
[1] Department of Wood and Furniture Design and Technology, Technological Educational Institute of Thessaly, Karditsa, V. Griva 11, 431 00. Greece.
[2] Institute of Research and Technology of Thessaly, Centre for Research and Technology, Dimitriados 95 & Pavlou Mela, Volos, 383 33, Greece.
[3] Dept. of Business Administration, Technological Educational Institute of Thessaly, Larissa, 411 10, Greece.

* Corresponding author. Tel.: +30-24410-64712; email: karageorgos@teithessaly.gr
Manuscript submitted June 4, 2015; accepted September 25, 2015.
doi: 10.17706/jsw.10.12.1351-1358

**Abstract:** This paper introduces a model for supporting human resource allocation decisions in software development projects. The model's underlying allocation method is based on an extension of the bio-inspired *response threshold model* and takes into account various aspects of the human resource allocation problem, such as the skills of available human resources, activity-related skill requirements as well as social ties/relationships among involved human resources. The model is demonstrated using an exemplar case problem concerning human resource allocation to a set of software development tasks.

**Key words:** Human resource allocation, software project management, social networks, bio-inspired methods, response-threshold model.

## 1. Introduction

In addition to technical aspects, software engineering processes involve highly social activities [1]. Communication, cooperation and co-ordination between human resources are considered quite important, particularly in the context of open source and distributed software development projects [2], [3]. Project members, for example in a large distributed software project, work better when they engage in effective collaboration and interactions with their colleagues [3]. Consequently, the success of a large software development project does not depend only on the expertise of the people involved in the various project tasks, but also quite importantly on how effectively they collaborate, communicate and work together in teams [4], a fact that is indicated by their social ties/relationships. It has been argued that failure to consider collaboration issues in human resource allocation may compromise the project success [3].

In this paper, a model for allocating human resources to software development tasks while taking social ties/relationships into account is introduced. The proposed model considers human resource skills, activity-related skill requirements, and social ties represented in a social network structure. The allocation method employed is based on an extension of the bio-inspired *Response-Threshold Model* (RTM) [5] with the aim to dynamically allocate human resources to tasks in an optimized manner. The RTM model concerns association of entities to objects, such as bees with flowers, generated by entity responses to stimuli produced by objects. Each entity has an inherent threshold influencing its response to object stimuli. More specifically, an entity is considerably more likely to respond to a particular stimulus when the stimulus intensity exceeds the respective threshold of that entity. The remainder of this paper is structured as follows: Section 2 introduces the proposed human resource allocation model and Section 3 presents an exemplar case study scenario. Section 4 presents the benefits of the proposed model and a brief comparison with representative relevant work. Finally, Section 5 concludes the paper.

## 2. Socially-Related Human Resource Allocation

### 2.1. Human Resource Allocation Problem

In the proposed model a software development project $P$ is considered as a set composed of $N$ different software development tasks $t_i$ $(i = 1,2, \dots, N)$. Each task is planned to be executed as a set of $M$ individual development activities, that is $t_i = \{a_{i1}, a_{i2}, \dots, a_{iM}\}$. Each activity $a_{ij}$ $(i = 1,2, \dots, N,\ j = 1,2, \dots, M)$ requires a specific level of skills to be completed successfully. In particular, let $K$ be the number of development skills offered by the available human resources. Let $h_{ijk} \in [0, g]$ be the minimum level of skill required on skill $c_k (k = 1, 2, \dots, K)$ by activity $a_{ij}$ $(i = 1,2, \dots, N,\ j = 1,2, \dots, M)$, where $g \in \mathbb{N}$ and $g \geq 0$. A value of $h_{ijk}$ equal to 0 represents that no skill is required and a value equal to $g$ denotes that the highest skill level is required. Furthermore, let $D$ be a set comprising $L$ available human resources $d_l$ $(l = 1, 2, \dots, L)$ For each skill $c_k$ $(k = 1, 2, \dots, K)$ each $d_l \in D$ is characterized by a respective skill level $r_{lk} \in [0, g]$.

The model also considers the software project social network structure $SN = (D, E)$, that is defined as a weighted undirected graph, where each vertex $d_l \in D$ represents a human resource and each edge $e = \left(d_l, d_q, w_{lq}\right) \in E$ indicates a social "tie" denoting a collaboration relationship between human resources $d_l$ and $d_q$ [4]. A weight $w_{lq} \in [0, 1]$ near 0 represents that human resources $d_l$ and $d_q$ have not communicated/collaborated as "closely" as two other human resources $d_{l'}$ and $d_{q'}$ with a weight $w_{l'q'}$ near 1. Given a set of project tasks $P$ with their respective development activities and a group of available developers $D$, the human resource allocation problem refers to an effective activity-to-developer assignment such that that task (skill-related) requirements are satisfied and social relationships between team members allocated to activities of the same task are maximized.

### 2.2. Socially-Aware Human Resource Allocation Model

In the proposed model each available individual human resource $d_l$ $(l = 1, 2, \dots, L)$ is considered a passive object characterized by a set of stimuli $S_l$. $s_{lk} \in S_l$ denotes the stimulus generated from human resource $d_l(\mathrm{l} = 1, 2, \dots, L)$ with respect to skill $c_k$ $(k = 1, 2, \dots, K)$. $s_{lk}$ can be calculated by (1) as follows:

$$s_{lk} = \begin{cases} r_{lk}, & d_l \text{ is available} \\ \left|r_{lk} - h_{ijk}\right|, & d_l \text{ is engaged in activity } a_{ij} \end{cases} \tag{1}$$

where, $r_{lk}$ is the level of skill $c_k$ possessed by human resource $d_l$ and $h_{ijk}$ is the level of skill $c_k$ which is required for proper execution of activity $a_{ij}$ $(\mathrm{i} = 1,2, \dots, N,\ j = 1,2, \dots, M)$ in which $d_l$ is engaged. $s_{lk}$ is calculated for each skill $c_k$ and each developer $d_l \in D$.

To capture social relationships among members of a development team formed in the context of a task, the concept of *social factor* is introduced. The social factor, denoted by $F$, reflects the degree to which a human resource $d_l$ $(d_l \in D,\ \mathrm{l} = 1, 2, \dots, L)$, is socially related with human resources already assigned to activities of a task. Considering that a number of $E_i \leq L$ human resources have been assigned to perform the activities of task $t_i \in P$, $F_{il}$ is given by (2):

$$F_{il} = \frac{\sum_{q=1}^{E_i} w_{lq}}{E_i} \tag{2}$$

where $\sum_{q=1}^{E_i} w_{lq}$ is the sum of weights of the social connections of a human resource $d_l \in D$ with the human resources ($d_q \in D$, q $= 1,2,\dots,E_i$, and $d_q \neq d_l$) engaged in task $t_i \in P$, and $E_i$ is the total number of human resources engaged in activities of task $t_i \in P$. $F_{il}$ is calculated for each human resource $d_l \in D$ which is a candidate for undertaking execution of an activity of a task $t_i$.

Each activity $a_{ij}$ $(i = 1,2,\dots,N,\ j = 1,2,\dots,M)$ in task $t_i$ $(i = 1,2,\dots,N)$ is associated with a set of response thresholds $\Theta_{ij}$ $(i = 1,2,\dots,N,\ j = 1,2,\dots,M)$, where $\theta_{ijk}$ is the response threshold of activity $a_{ij}$ $(i = 1,2,\dots,N,\ j = 1,2,\dots,M)$ with respect to skill $c_k$ $(k = 1,2,\dots,K)$. Threshold $\theta_{ijk}$ reflects the tendency of activity $a_{ij}$ to respond to stimulus $s_{lk}$ of human resource $d_l \in D$, $l = 1,2,\dots,L$ with respect to skill $c_k$ $(k = 1,2,\dots,K)$. The response threshold $\theta_{ijk}$ of activity $a_{ij}$ with respect to skill $c_k$, considering human resource $d_l$ is calculated by (3):

$$\theta_{ijk} = \begin{cases} h_{ijk},\ E_i = 0 \\ 0,\ E_i > 0\ and\ h_{ijk} < F_{il} \\ h_{ijk} - F_{il},\ E_i > 0\ and\ \mathrm{h_{ijk}} > F_{il} \end{cases} \tag{3}$$

where, $h_{ijk}(i = 1,2,\dots,N,\ j = 1,2,\dots,M,\ k = 1,2,\dots,K)$ is the level of skill $c_k$ required for execution of activity $a_{ij}$, $E_i$ is the total number of human resources engaged in activities of task $t_i \in P$, and $F_{il}$ is the social factor of task $t_i$ with respect to human resource $\mathrm{d_l} \in D$, $l = 1,2,\dots,L$, with $\theta_{ijk} \in [0, g]$. $\theta_{ijk}$ is calculated for each skill $c_k$ and each developer $d_l \in D$. Consequently, the probability $T_{ij}^l$ of activity $a_{ij}$ to select a human resource $d_l$ (i.e., the probability that resource $d_l$ will be assigned to activity $a_{ij}$) can be calculated by (4):

$$T_{ij}^l = \frac{\sum_{k=1}^{K} s_{lk} \times \theta_{ijk}}{\sqrt{\sum_{k=1}^{K} (s_{lk})^2} \times \sqrt{\sum_{k=1}^{K} (\theta_{ijk})^2}} \tag{4}$$

where $s_{lk}$ is the stimulus of resource $d_l \in D$ (as calculated by (1)), $\theta_{ijk}$ is the response threshold of activity $a_{ij}$ with respect to skill $c_k$ (calculated by (3)). $T_{ij}^l \in [0, 1]$ with values close to 0 representing low probability, while values close to 1 standing for high probability for allocating resource $d_l$ to activity $a_{ij}$. $T_{ij}^l$ is calculated for each developer $d_l \in D$. Equation (4) is inspired by cosine similarity [6] which is a measure of similarity between two vectors. The proposed formula ensures that the most "suitable" resource will be selected for each task, since the higher the variance between the two sets, stimulus and threshold, the lower the probability. This ensures that the most suitable and not the "best" human resource will be selected for each task, as finding the "best" is not always related with the optimal decision [7].

Finally, for defining entities and objects, as required by the RTM model, an agent-based approach is adopted. Tasks are modelled as autonomous decision-making entities termed *task agents* that are able to select human resources, considered to be passive objects, based on local information. One of the most important benefits of agent-based systems is their ability to capture emergent phenomena resulting from the interactions of individual entities [8]. For this reason, in the proposed model tasks are modelled as agents which act locally based on the bio-inspired response threshold model, and the allocation of human resources to tasks emerges as the result of individual agent actions.

## 2.3. Human Resource Allocation Recommendation Algorithm

Based on the above model, a simple algorithm for generating human resource allocation recommendation is described below. Initially the stimuli thresholds are estimated for all activities and all tasks. Subsequently for each activity of each task the following steps are executed:

The algorithm takes as input the required skill levels $h_{ijk}(i = 1,2,\dots,N,\ j = 1,2,\dots,M,\ k = 1,2,\dots,K)$ for all skills and all activities of all tasks.

1) For each activity $a_{ij}(i = 1,2, \dots, N, j = 1,2, \dots, M,)$ of each task $t_i$ $(i = 1,2, \dots, N)$:

a) For each human resource $d_l$ $(d_l \in D,\ l = 1, 2, \dots, L)$:

i. For each skill $c_k$ a stimulus $s_{lk}$ is calculated using (1).

ii. The social factor $F_{il}$ for task $t_i$ and human resource $d_l$ is calculated using (2).

iii. For each skill $c_k$ a skill threshold $\theta_{ijk}$ is calculated using (3).

iv. The probability $T_{ij}^{l}$ of $a_{ij}$ selecting human resource $d_l$ is calculated using (4).

v. By sampling from the discrete distribution defined by $T_{ij}^{l}$, for example using a method such as the alias method described in [9], a human resource $d_a$ is selected and assigned to activity $a_{ij}$.

2) Finally, a recommendation for human resource allocation is generated when all activities are assigned to human resources, or when no feasible allocation can be found.

We assume that the level of skills required by task activities and the level of skills possessed by available human resources are subjectively evaluated by the involved project managers. To achieve an, as much as possible, objective evaluation of knowledge and skills, we can apply a group-based decision method, such as the one we recently introduced in [10]. This method follows a similarity degree-based aggregation technique to derive an objective and aggregated assessment (on behalf of all involved project managers) for provided/required skills/competencies. In the general case, the above algorithm could be repeated for a number of cycles and the order in which activities will be assigned human resources would be randomized before each cycle.

## 3. Exemplar Application Scenario

In an exemplar scenario, we assume that five $(L = 5)$ human resources in a software project, $d_1$, $d_2$, $d_3$, $d_4$ and $d_5$, create the project social network presented in Fig. 1. We also assume that these human resources have been evaluated by following the group-based assessment introduced in [10]. In particular, we assume that the involved project managers have assessed the candidate human resources according to three $(K = 3)$ required development skills, which are knowledge in SQL, Java and C programming. Table 1 shows the results of applying the assessment method introduced in [10], where final (aggregated) assessments take values in the interval $[0, 5]$, where 0 denotes “very low” and 5 denotes “absolutely high” level of the corresponding skill, respectively. We finally assume that there are two development tasks $(N = 2)$, $t_1 = \{a_{11}, a_{12}\}$ and $t_2 = \{a_{21}, a_{22}, a_{23}\}$ which have been also evaluated by the involved project managers (i.e., by applying again the assessment method introduced in [10]) on the skill level required for their respective activities. The resulted assessment results are shown in Table 2.

Initially, a randomly chosen task agent $A_{t_1}$ calculates the probability to respond to (i.e. select) the available human resources with respect to activity $a_{11}$, considering that no human resource $(E_1 = 0)$ is engaged to any other activity of task $t_1$. The stimuli of the available human resources and the response threshold of activity $a_{11}$ are calculated according to equations (1) and (3) respectively, and they are provided in Table 3. The probability of task agent $A_{t_1}$ responding to human resource $d_1$ is calculated by applying equation (4), as follows:

$$T_{a_{11}}^{1} = \frac{\sum_{k=1}^{3} s_{1k} \times \theta_{11k}}{\sqrt{\sum_{k=1}^{3}(s_{1k})^2} \times \sqrt{\sum_{k=1}^{3}(\theta_{11k})^2}} = \frac{1 \times 5.43 + 3 \times 1 + 4.69 \times 2.43}{\sqrt{1^2 + 3^2 + 4.69^2} \times \sqrt{5.43^2 + 1^2 + 2.43^2}} = 0.58$$

Accordingly, the computation of equation (4) yields the probabilities for human resources $d_2$, $d_3$, $d_4$ and $d_5$, which are $T_{a_{11}}^{2} = 0.94$, $T_{a_{11}}^{3} = 0.58$, $T_{a_{11}}^{4} = 0.58$ and $T_{a_{11}}^{5} = 0.89$, respectively. Since no teams have formed yet, the human resource with skill level closer to the skill levels required by the activity has higher probability to be selected. Therefore, considering the distribution defined by the probabilities, task agent $A_{t_1}$ selects human resource $d_2$ to be allocated in activity $a_{11}$.

Considering that $d_2$ has been allocated in activity $a_{11}$ of task $t_1$, task agent $A_{t_1}$ calculates the corresponding probabilities for activity $a_{12}$. The updated human resource stimuli and the respective response thresholds of $a_{12}$, considering the social network shown in Fig. 1., are presented in Table 4 and Table 5, respectively.

Table 1. Available Human Resources and Their Assessment on Provided Skills

| Development Skills ($c$) | Software Human Resources ($d$) $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ |
|---|---|---|---|---|---|
| SQL | 1 | 3.25 | 0.55 | 0.55 | 5.31 |
| Java | 3 | 2 | 1 | 1 | 4.22 |
| C | 4.69 | 2.24 | 3 | 3 | 4.69 |

Table 2. Development Tasks and Their Assessment on Required Skills

| Development Skills (c) | Tasks ($t$): $t_1$ | | | $t_2$ | |
|---|---|---|---|---|---|
| | Activities ($a$): $a_{11}$ | $a_{12}$ | $a_{21}$ | $a_{22}$ | $a_{23}$ |
| SQL | 5.43 | 2 | 5.69 | 3.75 | 4.75 |
| Java | 1 | 1.5 | 4.75 | 5.31 | 1.69 |
| C | 2.43 | 4.22 | 2 | 1.31 | 1.31 |

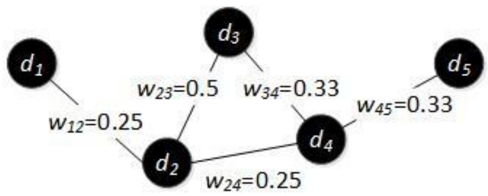


Fig. 1. Graph of social relationships between developers.

Table 3. Stimuli of Human Resources and Response Threshold for Activity $a_1{}^1$

| | | Development Skills ($c$) SQL | Java | C |
|---|---|---|---|---|
| Stimulus ($s$) | $d_1$ | 1 | 3 | 4.69 |
| | $d_2$ | 3.25 | 2 | 2.24 |
| | $d_3$ | 0.55 | 1 | 3 |
| | $d_4$ | 0.55 | 1 | 3 |
| | $d_5$ | 5.31 | 4.22 | 4.69 |
| Threshold ($\theta$) | $a_{11}$ | 5.43 | 1 | 2.43 |

Table 4. Updated Stimuli of Human Resources

| | | Development Skills ($c$) SQL | Java | C |
|---|---|---|---|---|
| Stimulus ($s$) | $d_1$ | 1 | 3 | 4.69 |
| | $d_2$ | 2.18 | 1 | 0.19 |
| | $d_3$ | 0.55 | 1 | 3 |
| | $d_4$ | 0.55 | 1 | 3 |
| | $d_5$ | 5.31 | 4.22 | 4.69 |

The probability of task agent $A_{t_1}$ responding to human resource $d_1$ with respect to activity $a_{12}$ is calculated as follows:

$$T^1_{a_{12}} = \frac{\sum_{k=1}^{3} s_{2k} \times \theta_{12k}}{\sqrt{\sum_{k=1}^{3}(s_{2k})^2} \times \sqrt{\sum_{k=1}^{3}(\theta_{12k})^2}} = \frac{1 \times 1.75 + 3 \times 1.25 + 4.69 \times 3.97}{\sqrt{1^2 + 3^2 + 4.69^2} \times \sqrt{1.75^2 + 1.25^2 + 3.97^2}} = 0.94$$

Table 5. Response Threshold for Activity $a_{12}$ for Each Human Resource

| | | Activity $\boldsymbol{a_{12}}$ | | |
|---|---|---|---|---|
| | | SQL | Java | C |
| Threshold ($\boldsymbol{\theta}$) | $d_1$ | 1.75 | 1.25 | 3.97 |
| | $d_2$ | 2 | 1.5 | 4.22 |
| | $d_3$ | 1.5 | 1 | 3.72 |
| | $d_4$ | 1.67 | 1.17 | 3.89 |
| | $d_5$ | 2 | 1.5 | 4.22 |

Accordingly, the probabilities for human resources $d_2$, $d_3$, $d_4$ and $d_5$ are $T^2_{a_{12}} = 0.56$, $T^3_{a_{12}} = 0.98$, $T^4_{a_{12}} = 0.97$ and $T^5_{a_{12}} = 0.9$, respectively. Based on these probabilities, even though human resources $d_3$ and $d_4$ have been evaluated as equally skilled (Table 1), human resource $d_3$ has a higher probability to be selected and allocated to activity $a_{12}$ and form a team with human resource $d_2$, with which he/she shares a higher social relationship. Thus, $A_{t_1}$ will select human resource $d_2$ considering the distribution defined by the probabilities for activity $a_{12}$. Task agents will sequentially continue calculating thresholds and stimuli as they change the allocation until all activities are assigned to candidate human resources.

## 4. Relevant Work and Discussion

In the context of software projects, considering social relationships, when taking human resource allocation decisions, has been seen from difference perspectives in the relevant literature. Some representative research works are summarized below.

Zhou in [11] suggests a project human resource allocation method based on software architecture and social networks in a project organization. The author introduced an algorithm for matching employees and tasks based on task attributes, software architecture, and employee skills, preferences and social relations. Social relations among employees are defined based on their personal relations. Lappas, Liu and Terzi in [4] were the first to address the problem of forming a team of skilled individuals to perform a given task, while minimizing the communication cost among the members of the team through their social network. They studied and analyzed instances of the problem and they also proposed practical algorithmic solutions considering communication costs. Anagnostopoulos *et al*. in [12] proposed efficient algorithms that address task allocation and team formation such that the required skills are satisfied and approximation guarantees, with respect to team communication overhead, are provided.

The above approaches treat available resources as equally skilled. However, in human centered activities, such as software development projects, personnel assignment decisions considering individual level of skills of each available resource can clearly contribute to project success [13]. Skills assessments are one the main considerations of the model proposed in the current paper. Furthermore, the proposed model supports human resource allocation in software projects based on a bio-inspired paradigm, while taking into account social relationships among available human resources. Up to our level of knowledge, there is no human resource allocation approach in the relevant literature comprising both an effective approximation method and a computation mechanism with the aim to consider social relationships when allocating human resources to software development tasks. A significant advantage of applying bio-inspired approximation methods is that they are able to handle the inherent problem complexity by using a set of relatively simple rules and the can achieve effective management of constrained resources reaching efficient solution equilibria. Therefore, when the number of the task allocation problem parameters increases, the proposed approach is expected to outperform other resource allocation approaches which also consider social relationships when taking allocation decisions. Such a comparison that will evaluate the performance of the proposed model is currently ongoing work.

## 5. Conclusions

This paper suggested a model for human resource allocation in software development tasks with the aim to ensure maximum collaboration between co-workers. The proposed model takes into account human resource skills, task-related skill requirements and social relationships between human resources, and dynamically allocates human resources to tasks in an optimised manner using the response-threshold bio-inspired paradigm. The consideration of social relationships among human resources aims to ensure the best possible collaboration between project team members. Our future research plans include extensive simulation experiments and comparison of the proposed model with other approaches.

## Acknowledgment

This research has been co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework, in the context of the research project “SPRINT SMEs”, Act: “ARCHIMEDES III” in the Operational Program “Education and Lifelong Learning 2007-13”.

## References

[1] Bell, D., (1986). *Software Engineering: A Programming Approach*. Prentice Hall International (UK).

[2] Hahn, J., Moon, J. Y., & Zhang, C. (2008). Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Information Systems Research*, *19(3)*, 369-391.

[3] Layzell, P., Brereton, O. P., & French, A. (2000). Supporting collaboration in distributed software engineering teams. In D. Poo, J. S. Dong, J. He, & M. Purvis (Eds.), *Proceedings of the Seventh IEEE Asia-Pacific Software Engineering Conference (APSEC 2000)* (pp. 38-45). Singapore: IEEE.

[4] Lappas, T., Liu, K., & Terzi, E. (2009). Finding a team of experts in social networks. In J. Elder, F. S. Fogelman , P. Flachl, & M. Zaki (Eds.), *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 467-476). Paris, France: ACM.

[5] Bonabeau, E., Theraulaz, G., & Deneubourg, J.-L. (1998). Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, *60(4)*, 753-807.

[6] Lee, M., Pincombe, B., & Welsh, M. (2005). An empirical evaluation of models of text document similarity. In B. G. Bara, L. Barsalou, & M. Bucciarelli (Eds.), *Proceedings of the 27th Annual Conference of the Cognitive Science Society* (pp. 1254-1259). Mahwah, NJ: Lawrence Erlbaum Associates.

[7] Callegari, D. A., & Bastos, R. M. (2009). A multi-criteria resource selection method for software projects using fuzzy logic, In J. Felipe & J. Cordeiro (Eds.), *Proceedings of the 11th International Conference in Enterprise Information Systems* (pp. 476-388). Milan, Italy: Springer-Verlag.

[8] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences* , *99(3)*, 7280-7287.

[9] Vose, M. D. (1991). A linear algorithm for generating random numbers with a given distribution*. IEEE Transactions on Software Engineering*, *17(9)*, 972-975.

[10] Gerogiannis, V. C., Rapti, E., Karageorgos, A., & Fitsilis, P. (2015). A fuzzy linguistic approach for human resource evaluation and selection in software projects. In A. Ali (Ed.), *Proceedings of 5th International Conference on Industrial Engineering and Operations Management (IEOM 2015)* (pp. 1-9). Dubai, UAE: IEEE.

[11] Zhou, L. (2008). A project human resource allocation method based on software architecture and social network. In L. Cuthbert & Z. T. Wang (Eds.), *Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)* (pp. 1-6). Dalian, China: IEEE.

[12] Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., & Leonardi, S. (2012) Online team formation in

social networks. In M. Rabinovich & S. Staa (Eds.) *Proceedings of the 21st international conference on World Wide Web* (pp. 839-848). Lyon, France: ACM.

[13] Otero, L. D., Centeno, G., Ruiz-Torres A. J., & Otero, C. E. (2009). A systematic approach for resource allocation in software projects. *Computers & Industrial Engineering*, *56(4)*, 1333-1339.

**Anthony Karageorgos** holds a B.Sc. in applied mathematics, an M.Sc in computer science and a PhD in agent-based software engineering. He has participated in numerous research projects in Greece and UK, which resulted in patents and several international journal and conference publications.

Furthermore, he has taught undergraduate and postgraduate courses in the areas of information systems and software engineering, and since January 2014 he is with the Department of Wood and Furniture Design and Technology, TEI of Thessaly, Greece, holding an assistant professor position on the subject of "Applied Informatics".

Dr. Karageorgos is an associate editor of the International Journal (ACM Transactions on Autonomous Adaptive Systems). Furthermore, he serves as a reviewer in various international journals, and he regularly participates in the organisational or program committees of several international conferences. His current research interests include intelligent software, self-organising and service-oriented systems, multi-agent applications, and simulation modelling. Dr Karageorgos is a member of IEEE Computer Society and ACM.

**Elli Rapti** is a PhD candidate at the Department of Electrical and Computer Engineering, University of Thessaly, Greece. She received her diploma degree in electrical and computer engineering from the Democritus University of Thrace, Xanthi, Greece, in 2010, and her master's degree in science and technology of computers and telecommunications from the University of Thessaly, Greece, in 2014.

She is currently a research assistant at the Technological Educational Institute of Thessaly, Greece specializing in the area of distributed intelligent systems.

**Vassilis C. Gerogiannis** holds a diploma and a PhD in software engineering from the University of Patras, Greece.

He is an associate professor at the Business Administration Department of the Technological Education Institute of Thessaly, Greece. He is also an adjunct professor at the Hellenic Open University. He has published a number of papers in international journals and conference proceedings. His research interests include software engineering and project management. He is currently the coordinator of the SPRINT SMEs project, an R&D project focusing on the improvement of software development processes in small & medium-sized SW enterprises. Dr. Gerogiannis has been involved as a project manager in a member of EU and national R&D projects.