

# Using Analytic Hierarchy Process approach in ontological multicriterial decision making – preliminary considerations

Katarzyna Wasielewska and Maria Ganzha

*System Research Institute Polish Academy of Sciences, Warsaw, Poland*

## **Abstract.**

In this paper we consider combining ontologically demarcated information with Saaty's Analytic Hierarchy Process (AHP) [1] for the multicriterial assessment of offers during contract negotiations. The context for the proposal is provided by the *Agents in Grid* project (AiG; [2]), which aims at development of an agent-based infrastructure for efficient resource management in the Grid. In the AiG project, software agents representing users can either (1) join a team and earn money, or (2) find a team to execute a job. Moreover, agents form teams, managers of which negotiate with clients and workers terms of potential collaboration. Here, ontologically described contracts (*Service Level Agreements*) are the results of autonomous multiround negotiations. Therefore, taking into account relatively complex nature of the negotiated contracts, multicriterial assessment of proposals plays a crucial role. The AHP method is based on pairwise comparisons of criteria and relies on the judgement of a panel of experts. It measures how well does an offer serve the objective of a decision maker. In this paper, we propose how the AHP method can be used to assess ontologically described contract proposals.

**Keywords:** semantic data processing, AHP, multicriterial proposal assessment, autonomous negotiations

**PACS:** <Replace this text with PACS numbers; choose from this list: <http://www.aip.org/pacs/index.html>>

## **INTRODUCTION**

Nowadays, Grids are gaining more and more popularity as an interesting approach to create highly available computing infrastructure being a source of “computational power” and “big data” (however, in this contribution we focus our attention on computational Grids only). For such infrastructure, including heterogeneous, geographically distributed computer resources, it was always assumed that it will become a source of income to its owners [3] (e.g. similar to the Sun Grid by Oracle [4]). When considering integration of business into the Grid infrastructure, note that “business” computations are usually conducted in a predefined order, and have to be completed within a fixed time frame. Therefore, in order to assure availability of resources and timeliness of job execution, it is necessary to formalize the contract between the user of a service and the service provider, specifying details of the service, in the form of a *Service Level Agreement (SLA)*. The complex nature of the job execution commission problem (and, similarly, the team joining problem) that includes both tangible and intangible aspects, justifies the need for utilizing a multicriterial decision making approach. The *Analytical Hierarchy Process (AHP)*; [1, 5, 6]) is a multicriteria decision making method known since the 70s that helps the decision maker to solve complex problems with subjective criteria. There have been numerous applications of the method in its original version as well as further developments in areas such as project management [7], strategy selection [8], supplier selection [9, 10], and many more [11]. Moreover, the method has been extended and evaluated in multiple application areas (for more details, see [12, 13]). The idea of use of pairwise comparisons and scaling, to compare individual criteria stems from psychological observations. One of the advantages of the AHP method is the idea to structure criteria as a hierarchy, which gives the user a better view on the problem and precisely determines problem's scope. Note that, when there are too many elements on a single level (usually more than 9), decision maker should arrange them in clusters to optimally allocate weights.

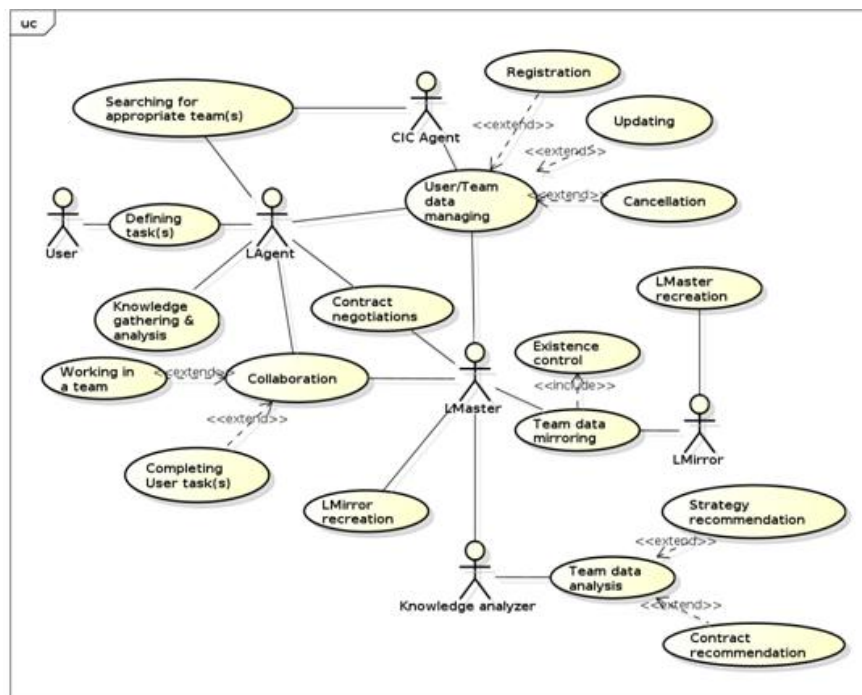
The context of this paper, is provided by the *Agents in Grid* project (AiG; [2]), which aims at the development of an agent-based infrastructure for intelligent resource management in the Grid. However, it is important to note that consideration presented in what follows apply to any situation where (computational) resources are to be provisioned through some form of contract negotiations. Therefore, everywhere that the AiG project is mentioned, readers is advised to substitute her/his favourite scenario. Overall, in the considered scenarios, contract structure is determined by the ontology with the a set of criteria represented as properties. More specifically, contract describing terms of collaboration between agents (i.e. managers representing teams of agents and users/workers) in the Grid should include both Grid resource description as well as economical factors. For such contracts, we propose to take advantage of

the hierarchical nature of ontologically demarcated data and to use it to construct problem hierarchy as proposed by Saaty in the *AHP* method. Note that, in our literature search, we were able to find only one contribution that combined ontologies and the *AHP* method [9]. These authors consider applying the *AHP* process to select services that are described with an ontology. However, their work reports very early stages of research that does not seem to be continued. Within the *AiG* project we want to take advantage of the characteristics of the ontological description to automatically generate comparison matrices and be able to extend them in an easy way. This will allow multicriterial evaluation of proposals in multi-round *SLA* negotiations.

## AGENTS IN GRID

Let us start by briefly introducing the *Agents in Grid* project (*AiG*), which aims at utilizing teams of software agents as resource brokers in the Grid. In the *AiG* system agents representing users can either (1) join a team and earn money, or (2) find a team to execute a job. Furthermore, agents form teams, consisting of managers and workers. Managers of teams negotiate with potential clients and workers terms of collaboration. The initial overview of the approach can be found in [14], while the two main scenarios (team joining and finding team to commission job execution) were discussed in [15, 16]. Since one of the main assumptions of the *AiG* system was to use ontologies and semantic data processing, we have developed an ontology of the Grid combined with ontologies needed for contract negotiations (for details, see [17]). In [14] we have proposed the agent-semantic system based on the following tenets (for more details, see the Use Case diagram in Figure 1, and the discussion presented in [2]):

- agents work in teams (groups of agents),
- each team has a single leader—*LMaster agent*,
- each *LMaster* has a mirror *LMirror agent* that can take over its job,
- incoming workers (*worker agents*) join teams based on user-criteria,
- teams (represented by *LMasters*) accept workers based on team-criteria,
- each *worker agent* can (if needed) play role of the *LMirror* or the *LMaster*,
- matchmaking is facilitated by the *CIC* component.



**FIGURE 1.** Use case diagram of *AiG* system.

Since in both scenarios (team joining and job execution) the goal is to autonomously reach “agreement on rules of collaboration” between agents (representing interests of their users). Such agreement is then formalized in a *Service Level Agreement (SLA)* “document”. In the *AiG* system, ontologically demarcated *SLA*’s result from multiround negotiations, within which offers are analyzed with respect to multiple criteria. Such criteria include, among others: price, completion time, availability of a specific hardware or software, etc. Obviously, different criteria may be of different importance to the user, e.g. completion time may be significantly more important than price. Therefore, multicriterial assessment of proposals has to play a crucial role in the negotiations.

## SAATY’S ANALYTIC HIERARCHY PROCESS

Let us now turn our attention to the *AHP* by Saaty, which can be used for multicriterial analysis of offers materializing during the negotiation process. Saaty’s method for multicriterial analysis is based on pairwise comparisons of criteria and judgement of experts (stakeholders), which enable to derive the priority scales. Briefly, the *AHP* method proceeds in three main phases: (i) choice of evaluation criteria, (ii) selection of experts, and (iii) quantitative evaluation. First two phases are handled at the design stage, when one gathers information about the decision that is to be made and factors that may influence it, as well as possible experts that will be able to evaluate alternative offers. In the *AHP* multiple experts may be included in the evaluation of offers during the decision-making process. Observe that there are evaluation criteria that can be assessed in a straightforward way e.g. the lower the price for the job execution the better for the user. However, to assess which payment method is better for a given user or a given type of problem, an expert knowledge is required. Therefore, in the context of the *AiG* system (or any similar system), it is relatively easy to distinguish two experts: (i) the automated module within the system, which has a set of rules to assess the criteria values and a self-learning knowledge base filled with information, e.g. which payment methods are better, (ii) the user that may manually assess how criteria values from alternative offers match her/his preferences. As to what concerns the knowledge base, it may be a “single-expert” base, or a “multi-expert” base. In the first case, knowledge of a single expert, or knowledge of multiple experts “combined together” will be instantiated. In the second case, knowledge of multiple experts will be stored separately and applied in the *AHP* method (which is a most natural scenario for its application). Furthermore, one of the main advantages of using the *AHP* method is that, in the decision-making process, it involves both tangible and intangible factors. Factors are compared using a predefined scale and an assessment is made on how much does one factor dominate another with respect to a given attribute.

Finally, the quantitative evaluation phase of *AHP* can be decomposed into several steps described below. The goal of this phase is to evaluate each offer numerically and to rank all of them according to how they meet user preferences.

First let us introduce the following notation:

- $C_1, \dots, C_M$ —criteria for alternatives evaluation,
- $A = \{A_i\}_{i=1, \dots, N}$ —the set of alternatives,
- $E = \{e_i\}_{i=1, \dots, K}$ —the set of experts.

Now, let us discuss how the *AHP* method proceeds step by step, when evaluating an offer.

1. Notice that preferences and factors that influence a decision can be depicted as a hierarchy (tree), such that the root is the goal, e.g. in our case, the contract. The level below the root contains the main objectives and group of factors influencing the decision, followed in subsequent levels by the criteria on which they depend, to the lowest level (which can be the set of alternatives). Figure 2 shows a sample decision hierarchy with  $n$  factors influencing the decision (criteria to consider during evaluation) about the goal from the broad perspective, 1-st factor has two component sub-factors, and  $n$ -th factor has  $m$  component sub-factors. Notice, that such structure can be further extended (in both breadth and depth).
2. In the next step a set of pairwise comparison matrices (see, Matrix 1) is constructed for the criteria, and the weights of the individual criteria are estimated. All elements on a given level are compared with respect to the element immediately above it (in the tree structure mentioned above).

Each comparison  $c_{ij}$ ,  $i, j = 1, \dots, M$  is made using a specific “scaling values” proposed by Saaty that indicate to what extent one element is more important than another (see, Table 1). For instance, if we want to contract job execution, and we consider  $i$ -th criteria (price) and  $j$ -th criteria (completion time) then  $c_{ij} = 5$  means that the price is strongly more important than the completion time. Similarly, we assign a value for each compared pair of criteria. In this way, user (decision maker) expresses his/her preferences regarding the goal and may influence the final estimate of an alternative by defining which factors are dominant. Table 1 shows possible values that can be

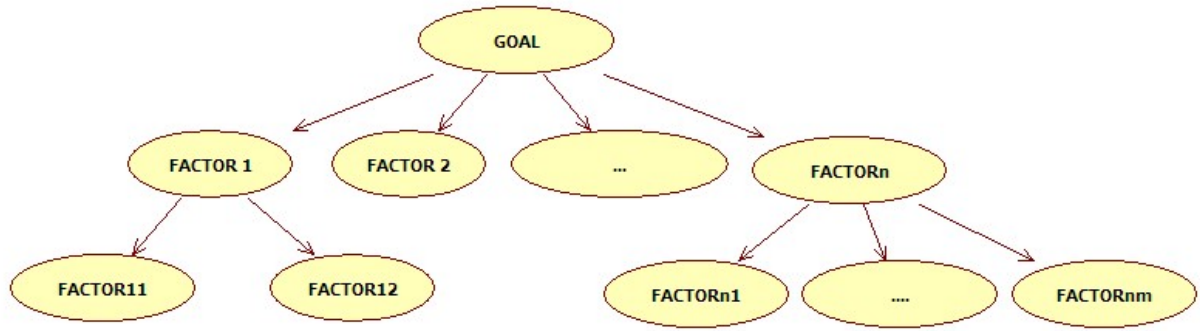


FIGURE 2. Example of decision hierarchy

assigned for comparisons. Let us notice, that the proposed method of criteria comparison allows one to express preferences in an easy way and compare factors that may seem incomparable e.g. delay penalty and job execution timeline.

TABLE 1. Scale used for criteria/experts comparisons ( [5])

Intensity of importance*	Definition
1	criteria/expert $i$ and $j$ are equally important
3	criteria/expert $i$ is moderately more important than $j$
5	criteria/expert $i$ is strongly more important than $j$
7	criteria/expert $i$ is very strongly more important than $j$
9	criteria/expert $i$ is extremely more important than $j$

\* values 2,4,6,8 are used in intermediate situations, moreover  $c_{ij} = \frac{1}{c_{ji}}$  and  $c_{ii} = 1$

$$\begin{bmatrix} 1 & c_{12} & \dots & c_{1m} \\ c_{21} & 1 & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mm} \end{bmatrix} \quad (1)$$

Note that:  $c_{ij} = \frac{1}{c_{ji}}$  and  $c_{ii} = 1$ .

As it was proved in [1], matrix 1 has a maximal eigenvalue  $\lambda_{max} = M + \Delta\lambda$  (where  $\Delta\lambda$  is small) and a corresponding eigenvector  $c$  with positives coordinates  $c_i > 0, i = 1, \dots, m$  corresponding to the  $\lambda_{max}$ . Weights of the criteria are the coordinates of the normalized eigenvector  $\bar{c}$ .

$$\bar{c}_i = \frac{c_i}{\sum_{i=1}^m c_i}$$

$$\sum_{i=1}^m \bar{c}_i = 1$$

3. We check the consistency ratio (CR) for constructed comparison matrices using a consistency index (CI) proposed by Saaty, if it less than 0.1 the matrix can be considered as having acceptable consistency.

$$CI = \frac{\lambda_{max} - m}{m - 1}$$

$$CR = \frac{CI}{RI}$$

where  $m$  - dimension of the matrix,  $\lambda_{max}$  - maximal eigenvalue,  $RI$  - random index

4. In the next step, the priorities evaluated from comparisons of criteria on each level are used to weight the priorities on the level immediately below and obtain global priorities e.g. if FACTOR 1 from Figure 2 has priority 0.2 and FACTOR 12 has priority 0.3, (calculated respectively from the comparison matrices for children of node GOAL and children of node FACTOR 1) then the global priority for FACTOR 12 is  $0.2 * 0.3 = 0.06$ .
5. If there is a group of experts involved in alternative evaluation of offers, comparison matrix for experts should be constructed and weights of individual experts should be estimated in the same way as it was done for the criteria. If there is only one expert that assesses the alternatives then his weight is 1, if there are  $K$  experts, whose estimates have the same importance then their weights are  $\frac{1}{K}$ . Otherwise, the procedure allows to specify how much more important is the estimate of one expert over another e.g. when contracting job execution estimates done by a panel of Grid / HPC experts may be treated as more significant than estimates of a user who wants to commission job execution. Weights of experts  $e_j$  are the coordinates of the normalized eigenvector  $w$  of the comparison matrix corresponding to its maximal eigenvalue.  
 $w = (w_j)_{j=1, \dots, K}$  and  $\sum_{j=1}^K w_j = 1$   
 Note that the system may provide default values in comparison matrices for both criteria and experts e.g. such that all experts are of the same importance and relative importance of criteria is assigned based on the knowledge at the moment of system design. However, user should have a possibility to change the default values and to specify e.g. that estimates done according to his rules of assessment are more important than assessments done with other experts' rules available in the system, e.g. that the time of the execution of the task is not as important as the low price of the service.
6. Finally, the last step in the quantitative evaluation phase is to evaluate alternatives with respect to experts. For every alternative offer we create an evaluation matrix where every element of the matrix is an estimation of the badness of the  $n$ -th alternative for  $j$ -th expert with respect to the  $i$ -th criteria  $x_{ij}^n, i = 1, \dots, M, j = 1, \dots, K$  (see, Table 3). Here, we use the scale from Table 2 to estimate alternatives. Let us notice, that the provided scale of absolute judgment allows to measure both tangible and intangible terms. The challenge is to design mappings of exact criteria values from offer onto the scale, e.g. which payment mechanism is good, and which is excellent for a given problem, or which completion time can be treated as bad, and which as good.

**TABLE 2.** Scale used for alternative estimation

Intensity of badness	Definition
-9	extremely bad
-7	very strongly bad
-5	strongly bad
-3	moderately bad
-1	bad
0	does not affect
1	good
3	moderately good
5	strongly good
7	very strongly good
9	extremely good

**TABLE 3.** Evaluation matrix for  $n$ -th alternative

	$e_1$	$e_2$	...	$e_K$
$C_1$	$x_{11}^n$	$x_{12}^n$	...	$x_{1K}^n$
$C_2$	$x_{21}^n$	$x_{22}^n$	...	$x_{2K}^n$
...	...	...	...	...
$C_M$	$x_{M1}^n$	$x_{M2}^n$	...	$x_{MK}^n$

The estimate  $M_n$  of the  $n$ -th alternative is calculated as:

$$M_n = \sum_{i=1}^M (c_i \sum_{j=1}^K x_{ij}^n w_{ij})$$

The best alternative is the one with the maximal estimate.

## SIMPLE EVALUATION EXAMPLE

To illustrate the above considerations, let us now present a simple example of an ontologically described contract evaluation using the *AHP* method. Let us consider a job execution scenario from the *AiG* project, where the user represented by a software agents is looking for a team to commission his/her job to execute. After finding teams that satisfy user requirements concerning resources, the negotiations start. First, a *Call for Proposal* message is sent to managers of selected teams. Managers respond with job execution contract offers that have to be ranked, and eventually one of the offers may be accepted. Note that all information exchanged in the system (including requirements and contract offer) is ontologically demarcated. For simplicity, we consider only a part of the structure that is available in the *AiG* ontology (see, [17]) to define a contract. In the *AiG* ontology, a set of classes and properties describe a contract. The following code snippet presents a selected part of the *AiG* ontology, describing a contract offer that has specified deadline penalty, job execution timeline and payment conditions with two properties: (i) delay penalty, and (ii) fixed availability price:

```
<ObjectProperty rdf:about="AiGConditionsOntology#paymentConditions" />
<DatatypeProperty rdf:about="AiGConditionsOntology#deadlinePenalty" />
<DatatypeProperty rdf:about="AiGConditionsOntology#fixedAvailabilityPrice" />
<Class AiGConditionsOntology#JobExecutionConditions" />
<Class_rdf:about="AiGConditionsOntology#PaymentConditions" />
<Class_rdf:about="AiGConditionsOntology#JobExecutionTimeline" />

<NamedIndividual_rdf:about="AiGConditionsOntology#Contract">
<rdf:type_rdf:resource="AiGConditionsOntology#JobExecutionConditions" />
<AiGConditionsOntology:deadlinePenalty_rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
  >91.56
</AiGConditionsOntology:deadlinePenalty>
<AiGConditionsOntology:paymentConditions_rdf:resource="AiGConditionsOntology#
  ContractPaymentConditions" />
<AiGConditionsOntology:jobExecutionTimeline_rdf:resource="AiGConditionsOntology#Dec15H18" />
</NamedIndividual>

<NamedIndividual_rdf:about="http://gridagents.sourceforge.net/
  AiGConditionsOntology#ContractPaymentConditions">
<rdf:type_rdf:resource="AiGConditionsOntology#PaymentConditions" />
<AiGConditionsOntology:delayPenalty_rdf:datatype="http://www.w3.org/2001/XMLSchema#float">20
</AiGConditionsOntology:delayPenalty>
<AiGConditionsOntology:fixedAvailabilityPrice_rdf:resource="AiGConditionsOntology#
  FixedAvailabilityPricing" />
</NamedIndividual>
```

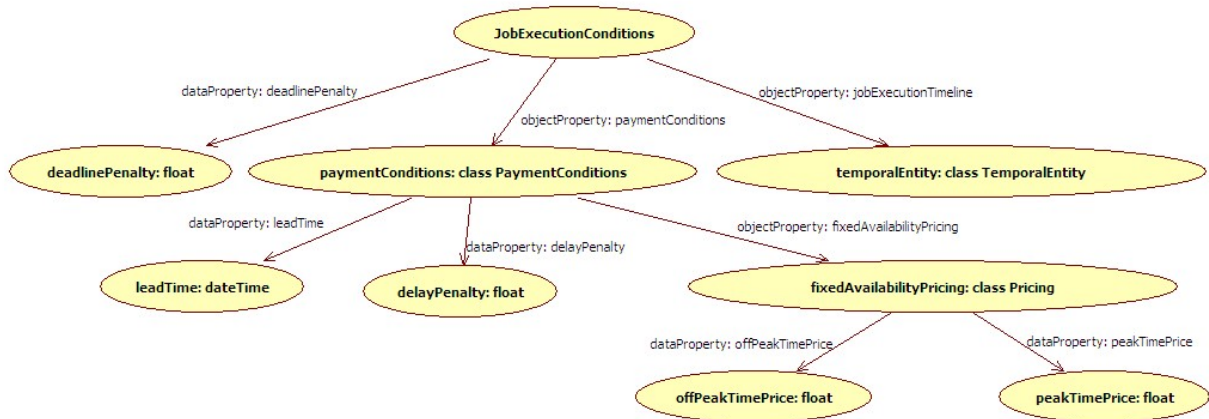
To be able to evaluate offers with Saaty's method first: (i) set of criteria has to be defined and global priority has to be computed for each criterion, (ii) set of experts (stakeholders) has to be established and, in case of a group of experts with different importance, their priorities have to be computed.

Based on the properties defined in the ontology, we can construct matrices with pairwise comparisons of criteria corresponding to these properties. Moreover, since ontology can be represented on an acyclic directed graph, we may quite intuitively determine the structure of the decision hierarchy. Henceforth, the top node is the "main goal." In the case of the *AiG* ontology, it is the contract represented with an instance of the *JobExecutionConditions* class (see, Figure 3). The children of the top node in the hierarchy are factors influencing the goal from a broad perspective i.e. properties with the domain *JobExecutionConditions* from the ontology, e.g. deadline penalty, job execution timeline and payment conditions. All other intermediate levels are criteria with further sub-criteria that they depend on, e.g. payment conditions depend on its lead time, delay penalty and fixed availability price.

Let us also assume (for this example) that a single expert evaluates the incoming contract offers i.e. in the *AiG* system there is a set of rules that allow to assess, which value from the Table 2 can be assigned to value of criteria in a given offer.

Now, we will discuss how the *AHP* method can be applied to evaluate two alternative contract offers. First, we construct pairwise comparison matrices for each level in the hierarchy from the Figure 3, where the elements in the lower level are compared with respect to the element immediately above them (see, Tables 4, 5, 6).

As stated above, values are assigned from the scale presented in Table 1. In the *AiG* system, the user will be provided with predefined values in the comparison matrices. However these values can be modified and adjusted to the user preferences and to the nature of the job to be executed. From the values assigned in the comparison table Table 4



**FIGURE 3.** Part of the contract structure from *AiG* ontology

**TABLE 4.** Comparison matrix for criteria top level criteria

	<b>deadline Penalty</b>	<b>payment Conditions</b>	<b>jobExecution Timeline</b>
deadlinePenalty	1	$\frac{1}{3}$	3
paymentConditions	3	1	5
jobExecutionTimeline	$\frac{1}{3}$	$\frac{1}{5}$	1

it can be deduced, that for the “example user” the deadline penalty is moderately less important than the payment conditions and moderately more important than the job execution timeline. Next, the payment conditions are strongly more important than the job execution timeline. From Table 5, we can see that the lead time is strongly more important than the delay penalty and moderately less important than the fixed availability price. Similarly, in Table 6 the peak time price is moderately more important than the off peak time price. It can be concluded that for this user payment conditions and specifically the peak time price are most important, but also important is the deadline penalty. While the lead time in payment plays a crucial role, job execution timeline is of minor importance.

In the *AHP* algorithm, the weights of criteria are coordinates of the normalized eigenvector corresponding to the maximal eigenvalue. Thus weights calculated from Table 4 constructed for the top level criteria are: deadline penalty 0.2, payment conditions 0.68 and job execution timeline 0.12. Analogically, we calculated weights for the criteria in Table 5 and Table 6. The weights of the sub-criteria with respect to the payment conditions are: lead time 0.28, delay penalty 0.07, fixed availability price 0.65, and the weights of the sub-criteria with respect to the fixed availability pricing are: off peak time price 0.25, peak time price 0.75.

Next, having weights (priorities) calculated as above for each group of criteria, we may calculate global priorities by weighting the priorities on the consecutive sub-levels. For this example we obtain: deadline penalty 0.2, lead time 0.19, delay penalty 0.05, off peak time price 0.11, peak time price 0.33, job execution timeline 0.12.

**TABLE 5.** Comparison matrix for PaymentConditions’ sub-criteria

	<b>lead Time</b>	<b>delay Penalty</b>	<b>fixedAvail. Price</b>
leadTime	1	5	$\frac{1}{3}$
delayPenalty	$\frac{1}{5}$	1	$\frac{1}{7}$
fixedAvail.Price	3	7	1

**TABLE 6.** Comparison matrix for FixedAvailabilityPricing’s sub-criteria

	offPeakTimePrice	peakTimePrice
offPeakTimePrice	1	$\frac{1}{3}$
peakTimePrice	3	1

**TABLE 7.** Evaluation matrix for 1-th alternative

	$e_1$
deadlinePenalty	1
leadTime	3
delayPenalty	5
offPeakTimePrice	0
peakTimePrice	7
jobExecutionTimeline	1

Let us further assume that the software agent representing the user received two alternative contract offers that have to be evaluated (and compared). For every alternative, an evaluation matrix is created, estimating badness of that alternative by a given expert for a given criteria, found in the Table 7 and Table 8. We use the scale proposed in Table 2. In the ontologically demarcated contract offer, exact values are received as in the code snippet presented above. Next, these exact values are mapped onto the values from a proposed absolute scale of judgement. Mapping utilizes rules defined in the system and expert’s knowledge e.g. price should be minimized, deadline penalty should be maximized, payment mechanism and adequacy of resource should evaluated based on expert’s knowledge.

The 1-st offer evaluated in Table 7 has good deadline penalty, very strongly good peak time price, moderately good lead time, good job execution timeline. The estimate  $M_1$  of the 1-st alternative is:

$$M_1 = 0.2 * 1 + 0.19 * 3 + 0.05 * 5 + 0.11 * 0 + 0.33 * 7 + 0.12 * 1 = 3.45$$

The 2-nd offer evaluated in Table 8 has moderately good deadline penalty, very strongly good peak time price, moderately good job execution timeline, however it has bad lead time. The estimate  $M_2$  of the 2-nd alternative is:

$$M_2 = 0.2 * 3 + 0.19 * (-1) + 0.05 * 5 + 0.11 * 1 + 0.33 * 5 + 0.12 * 3 = 2.78$$

As we can see, the 1-st alternative has a better estimate and should be selected. Considering the first three criteria with the highest weights, the 1-st offer has much better peak time price, better lead time and, even though the deadline penalty is not as satisfactory for the user as in 2-nd offer, the overall estimate is better.

## CONCLUDING REMARKS

We have demonstrated, how to apply the *AHP* method to compare ontologically described offers in generic contract negotiations. Obviously, a full graph of the ontological contract, would result in Saaty’s hierarchy with more levels, and more elements at each level. However, the structure of the ontology makes the mapping between the ontological contract description and a decision hierarchy from the Saaty’s *AHP* quite intuitive. Moreover, the method should be

**TABLE 8.** Evaluation matrix for 2-th alternative

	$e_2$
deadlinePenalty	3
leadTime	-1
delayPenalty	5
offPeakTimePrice	1
peakTimePrice	5
jobExecutionTimeline	3



able to deal with arbitrary large structures, The challenge for the future is to capture the expert knowledge about contractual terms and resource characteristics and to provide methods to assist the user in adjusting comparison matrices so that they correctly represent his/her preferences and are adequate for the profile of the job that is to be executed. Fortunately, the same challenge applies to the *AiG* project, as well as to any other scenario were the *AHP* method is applied, and as such has been already addressed in the literature.

## REFERENCES

1. T. L. Saaty, "The Analytic Hierarchy Process," RWS Publications, Pittsburg, 1990.
2. K. Wasielewska, M. Drozdowicz, M. Ganzha, M. Paprzycki, N. Attaui, D. Petcu, C. Badica, R. Olejnik, and I. Lirkov, "Trends in Parallel, Distributed, Grid and Cloud Computing for Engineering," Saxe-Coburg Publications, Stirlingshire, UK, 2011, chap. Negotiations in an Agent-based Grid Resource Brokering Systems.
3. J. Foster, and C. Kesselman, editors, *The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure*, The Elsevier Series in Grid Computing, Elsevier, 2004.
4. Sun utility computing, <http://www.sun.com/service/sungrid/> (2012).
5. T. L. Saaty, *International Journal of Services Sciences* **1**, 83–98 (2008).
6. T. L. Saaty, *European Journal of Operational Research* **48**, 9–26 (1990).
7. K. M. Al-Subhi Al-Harbi, *International Journal of Project Management* **19**, 19–27 (2001).
8. M. Chen, and S. Wang, *Expert Systems with Applications* **37**, 694–704 (2010).
9. G. Kabir, and R. Sumi, *Suranaree Journal of Science and Technology* **17**, 249–257 (2010).
10. I. Chamodrakas, D. Batis, and D. Martakos, *Expert Systems with Applications* **37**, 490–498 (2010).
11. A. Ishizaka, and A. Labib, *ORInsight* **22**, 201–220 (2009).
12. A. Ishizaka, and A. Labib, *Expert Systems and Applications* **38**, 14336–14345 (2011).
13. J. Alonso, and T. Lamata, *International Journal of Uncertainty* **14**, 245–259 (2006).
14. M. Dominiak, W. Kuranowski, M. Gawinecki, M. Ganzha, and M. Paprzycki, "Utilizing agent teams in grid resource management—preliminary considerations," in *Proc. of the IEEE J. V. Atanasoff Conference*, IEEE CS Press, Los Alamitos, CA, 2006, pp. 46–51.
15. W. Kuranowski, M. Ganzha, M. Paprzycki, and I. Lirkov, "Supervising Agent Team an Agent-based Grid Resource Brokering System Initial Solution," in *Proceedings of the Conference on Complex, Intelligent and Software Intensive Systems*, edited by F. Xhafa, and L. Barolli, IEEE CS Press, Los Alamitos, CA, 2008, pp. 321–326.
16. W. Kuranowski, M. Ganzha, M. Gawinecki, M. Paprzycki, I. Lirkov, and S. Margenov, *International Journal of Computational Intelligence Research* **4**, 9–16 (2008).
17. M. Drozdowicz, K. Wasielewska, M. Ganzha, M. Paprzycki, N. Attaui, I. Lirkov, R. Olejnik, D. Petcu, and C. Badica, "Trends in Parallel, Distributed, Grid and Cloud Computing for Engineering," Saxe-Coburg Publications, Stirlingshire, UK, 2011, chap. Ontology for Contract Negotiations in Agent-based Grid Resource Management System.