



MULTIPLE EQUIVALENT SIMULTANEOUS OFFERS STRATEGY IN AN AGENT-BASED GRID RESOURCE BROKERING SYSTEM – INITIAL CONSIDERATIONS

NAOUAL ATTAOUI*, MARIA GANZHA† MARCIN PAPRZYCKI‡ KATARZYNA WASIELEWSKA‡ AND MOHAMMAD ESSAAIDI ‡

Abstract. The *Agent in Grid (AiG)* project attempts to integrate the concept of the Grid and an agent-based system to facilitate efficient resource management in the Grid. In this paper, we present preliminary considerations concerning multiple equivalent simultaneous offers strategy that can be used in the *Service Level Agreement (SLA)* negotiations. These negotiations are the key part of the main use case scenarios within the *AiG* project. In this context, first, we describe the *AiG* system. Second, we introduce the simultaneous offer strategy, as a mechanism known from economy, and suggest an approach for using it in the *AiG* negotiations.

Key words: Grid computing, multiple equivalent simultaneous offers, AiG, agent-based negotiations, SLA, multi-attribute negotiations

1. Introduction. The context for the presented work is provided by the *Agents in Grid* project (*AiG*; [16, 17, 23]) that aims at creating a management system for the Grid computing resources. The system utilizes the concept of software agents that can act flexibly in a dynamic and uncertain environment, where resources can appear and disappear at any time. Furthermore, the *AiG* project integrates business principles, by allowing users to earn money by offering resources, or to pay for the use of available resources. The key part of the realization of this economic model is the negotiation process that should lead to contracts beneficial to the engaged parties. In the *AiG* project, negotiations materialize in the two main scenarios: (i) when the user would like to execute her job and is searching for resources that satisfy her needs, (ii) when a resource owner is seeking a team to join, to earn money. Here, we are confronted with one of the key assumptions behind the *AiG* project. Based on the analysis of working conditions of volunteer systems (e.g. *seti@home*; [32]), and taking into account progress in the development of agent systems (their capability for autonomously reaching an agreement [33]) it was stipulated that agents that offer their resources to earn money should work in teams. In this way, teams should be able to deliver results even if some members disappear without warning (for more details, see [16]).

Observe that negotiations taking place in the *AiG* project have to take into account multiple attributes. In an example given in [23] simple job execution negotiations involved simultaneous evaluation of *price*, *job start time* and *job end time*. Currently, after the *AiG* ontology was fully developed (see, [22] for more details), truly multi-criterial negotiations can be used. Note that in both negotiation scenarios introduced above, the negotiation strategy is constrained by the following conditions: (1) no information about the preferences of the other participant(s) in the negotiation, and (2) the agreement reached should be optimal to both negotiating parties.

Let us observe that recent studies in social psychology and economics [1, 2] have shown that using a strategy based on multiple offers is more profitable than a strategy based on a single offer. Here, the proposer can choose combinations of parameter values in such a way to obtain equivalent offers i.e. they generate equal values of the proposer's utility function. On the other hand, the second party in negotiations is more likely to be satisfied with one of these offers. Taking this into account, in this paper we would like to consider how the multi-offer mechanism can be introduced into the *AiG* system.

The remainder of the paper is organized as follows. Section 2 briefly describes the *AiG* system. Next, in section 3, the related work concerning multi-offer negotiations is reviewed. We follow, in section 4, with a description how to generate multiple equivalent offers in the *AiG*, and suggests how to model this strategy in the context of the *AiG* system. We conclude with a summary and description of future work.

2. System overview.

*Information and Telecommunication Systems Laboratory, Faculty of Science, Tetuan, Morocco

†Systems Research Institute Polish Academy of Sciences, Warsaw, Poland

‡National School of Computer Science and Systems Analysis, ENSIAS, Rabat, Morocco

2.1. AiG infrastructure. Let us start from a high-level overview of the *AiG* system. The *AiG* approach is based on (i) software agents that facilitate resource management in the Grid [15], and (ii) use of ontology for knowledge representation, and semantic data processing for knowledge utilization. As mentioned before, the *AiG* system constitutes a “return” to the roots of Grid computing, where providers were to sell their resources in the Grid, while their customers were to pay for the resource use. In other words, the *AiG* project attempts at (re)integrating business mechanisms into the Grid. This being the case, in the system we distinguish two roles, which are interchangeable: resource (service) provider and client (resource consumer). Finally, the key part of the proposed approach is: combining agent into teams. As stated above, this is a result of an analysis of realistic situations that take place in the *global Grid*. Here, the global Grid is contrasted with the *local Grid*, where resources run under control of one or more administrators. Let us now summarize main assumptions that have been driving the development of the agent aspect of the system:

- (i) Users and workers in the system are represented by *LAgents*.
- (ii) Agents work in teams [16, 17].
- (iii) Each team is managed by its leader, the *LMaster* agent.
- (iv) *Client Information Center* (represented by the *CIC* agent) plays the role of a central repository, where information about all teams and their member workers is stored [25].
- (v) In addition to the *LMaster* one of the worker agents plays a role of *LMirror*. This agent stores a copy of the information necessary for the team to persist in the case when the *LMaster* crashes.

The proposed approach can be thus represented in the form of a Use Case Diagram depicted in Fig. 2.1.

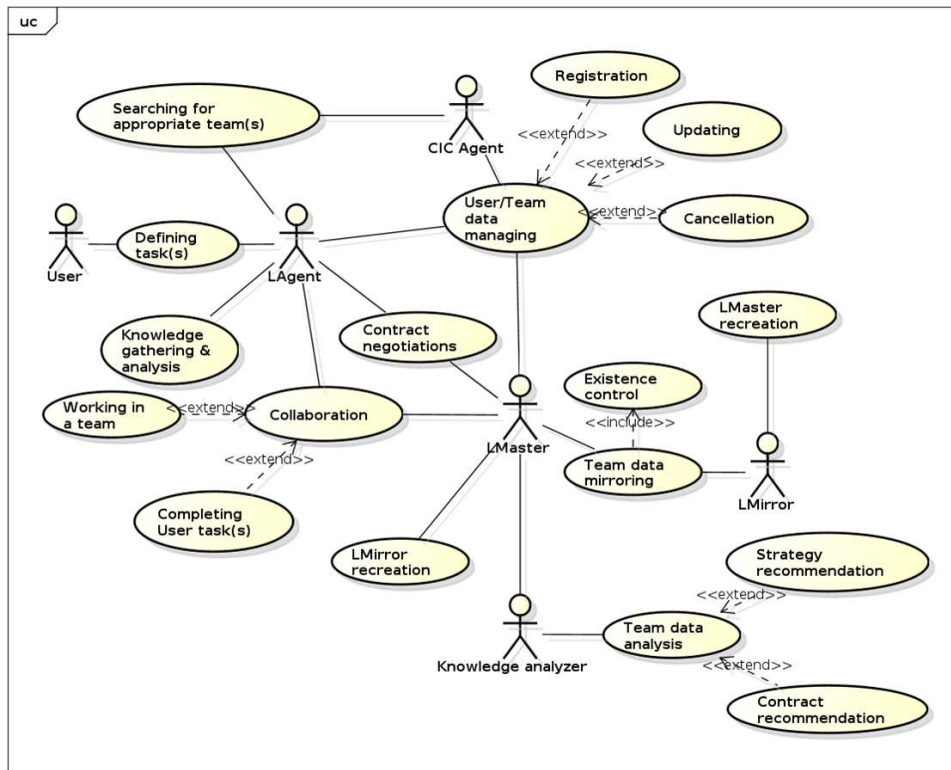


FIG. 2.1. Use Case Diagram of *AiG* system.

The work of the system is structured around two main scenarios: a user who wants to execute a job, and a user who tries to join a team. In the first scenario, when the user is seeking a team to execute her job, she specifies job constraints via web application that is ontology-driven and transforms the constraints into an ontology fragment. Next, using a special *Gateway Agent*, the interface forwards this fragment to the *LAgent*. The *LAgent* contacts the *CIC* to obtain the list of teams that can execute the job (under the defined constraints).

Next, the Iterated FIPA Contract Net Protocol [26] is applied to negotiate the *Service Level Agreement (SLA)*. The second, team joining, scenario is very similar. The only difference is in the constraints passed by the *LAgent* to the *CIC*. Here, the *CIC* is matching the description of the potential worker against requirements of teams that would like to have more members. In the context of this paper, the crucial observation is that the Iterated FIPA Contract Net Protocol can be modified to support the multi-offer strategy. Recall that the negotiation process covers multiple parameters and decisions about joining a team, accepting new workers, choosing a team to execute a job, or accepting to execute a job involve multi-criteria analysis (see section 2.3).

2.2. Ontologies in the system. The representation of the knowledge with ontologies has recently attracted attention, as a promising approach to development of a new generation of information systems (including the Semantic Web). Furthermore, agreeing with J. Hendler, who in [14] argued that software agents and ontologies should work very well together; it was decided that in the *AiG* system all concepts will be ontologically represented. Thus, the envisaged ontology had to encompass the structure of the Grid, the characteristics of the resources, and the negotiations that occur in both scenarios.

Preliminary research in this area involved analyzing existing Grid ontologies (see [18]). On the basis of this analysis it was concluded that the closest and the most adequate ontology, for the *AiG* system, was the *Core Grid Ontology (CGO)*; [19, 20]). However, some modifications and extensions had to be brought to this solution, especially so that it would include concepts related to the *SLA*. The complete description of the resulting ontology can be found in [21, 22]. Here, let us briefly outline its main features. For better modularity, and thus maintainability, we have split the constructed ontology into three (sub)ontologies:

1. *AiG Grid Ontology*, which is an extension of the original *CGO* that adds additional classes and properties that make specifying the hardware and software configuration of a Grid resource easier and more complete.
2. *AiG Conditions Ontology*, which contains concepts necessary to describe the terms of contracts between entities that are involved in business relationships: (a) the *LAgent* representing a user offering its resources and the *LMaster* looking for workers, and (b) the *LAgent* looking for a team to execute a job and the *LMaster* representing teams offering to run this job.
3. *AiG Messages Ontology* is used internally by the proposed solution and contains definitions of messages that can be exchanged between agents.

2.3. Negotiations in the system. Human negotiation is a daily act; we negotiate at work, at home, in the political domain, as well as in trading. Generally, stakeholders negotiate in order to reach an agreement, which satisfies their respective interests. Automated negotiations have the same goal, but they use software tools to assist negotiators to express their preferences, to accelerate reaching an agreement (even in complex situations [35]), and to help making a good decision. In our case, we must facilitate negotiations in the two scenarios mentioned earlier: (i) a user has some requirements concerning resource(s) needed to execute a job, while the resource owner wishes to maximize her gain, (ii) a prospective worker who wants to “sell” a resource is negotiating with a team that would like to assure an appropriate contract with the new worker. To reach agreement automated negotiations of the *SLA* between both parties are utilized. Based on earlier work [21], our ontology allows to consider, among others, the following parameters in the first scenario: (i) job execution start and end time, (ii) penalty for delay of job execution, and (iii) price. Parameters that can be negotiated in second scenario are (this list is, again, not exhaustive) : (i) revenue, (ii) contract duration, (iii) possibility to extend contract, and (iv) guaranteed utilization. Obviously, we deal with multi-attribute negotiations that can generate complications on several levels: generating offers, evaluating proposals, guessing negotiator preferences, etc.

From the above discussion, it can be concluded that autonomous negotiations, and the *SLA* management, are a key part of the *AiG* system (see [23] for more details). To be sure that the negotiation reaches its goal, we should choose protocols and strategies appropriate for the situation. Therefore, after considering the European projects integrating business and Grid infrastructure [24], it was decided to base the negotiations on the FIPA Iterated Contract-Net Protocol [26], to give the possibility for the user to modify his offer if no team presented an acceptable offer. Here, negotiations can involve both negotiable, e.g. deadline penalty, job execution timeline, and static parameters, e.g. resource description provided by the user.

Fig. 2.2 represents the sequence diagram of a situation when a user is attempting to execute a job (with

a help of her *LAgent*). The *LAgent* registers with the *CIC*, if not registered already. Afterwards, it requests from the *CIC* a list of teams that have workers representing resources meeting the criteria. As a result the *CIC* agent responds with a list of *LMaster* agents representing suitable teams. After that, the *LAgent* negotiates directly with the *LMasters*. In the initial step, the *LAgent* sends out the *Call-For-Proposal (CFP)* message to all *LMasters* of teams acquired from the *CIC*. On the basis of the *CFP*, *LMasters* prepare their responses. Responses from the *LMasters* can have the following forms: (1) refusal (an *ACL Refuse* message), or (2) a specific offer (an *ACL Propose* message). The *LAgent* awaits a predefined time for responses and then finds the best of them using a multi-criteria analysis (MCA). The team that has the best offer is selected to execute the job. If no acceptable offer is received, a revised *CFP* is sent to as selected subset of participants, while others are removed from further considerations.

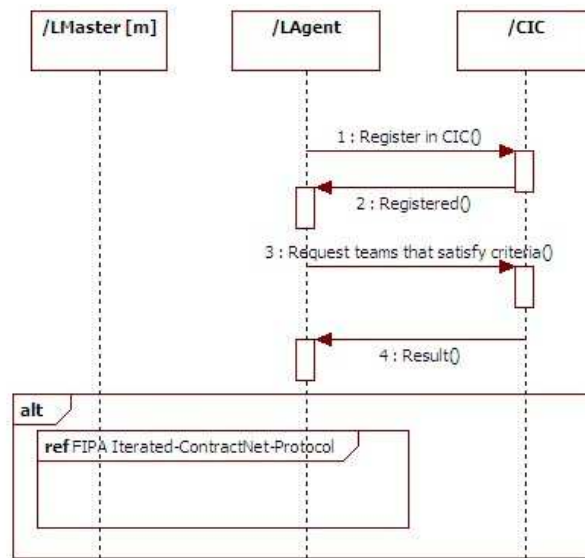


FIG. 2.2. Sequence diagram for job execution scenario.

In this paper we consider possibility of further developing/extending the negotiation mechanism through application of the Multiple Equivalent Simultaneous Offers strategy, which is introduced in the next section.

3. Multiple Equivalent Simultaneous Offers strategy. Multiple Equivalent Simultaneous Offers (MESOs) [1] is a technique used in negotiations where a set of attributes is involved. The principle is to make multiple offers at once. One negotiating party should make offers having the same utility and different values for individual parameters. In this case, the client will have more chance to be satisfied by an offer and thus negotiators are more likely to reach an agreement.

Recent research in social psychology and economy suggests that the choice of negotiation strategy based on the simultaneous offers is more profitable than the unique offer. A number of scholars have conducted studies and published articles related to the MESOs. Authors in [1] have demonstrated numerous benefits (distributive, integrative, and interpersonal) of making MESOs by conducting four online experiments, in which participants were offered a single offer, or two offers. Here, authors believe that the use of MESOs allows negotiators to get more information about each other across subsequent negotiation rounds and to achieve better overall outcomes. Other research ([3]), shows that negotiators who use MESOs achieve better outcomes than those who make a single packaged offer, without sacrificing relationships or losing credibility. While there is not a fixed (best overall) number of offers, authors of [2] suggest to present no more than three offers at a time. Similarly, according to [4], three equivalent offers can be a good strategy. Here, authors describe a software company that, in financial negotiations, presented three equivalent software packages to its clients at once: for example, a \$1

million package with payment in 30 days, the same software for \$1.5 million with payment in 120 days, and an enhanced package for \$1.35 million with a 30-day payment. This strategy received a positive response from the customers, and profits of the company rose. Yinping Y et. al. confirmed in [5] the effectiveness of a novel strategy based on simultaneous equivalent offers to make a significant positive impact on both the economic and social-psychological outcomes of negotiations. Authors of [28] opt for multiple offers to decrease the number of negotiation rounds. They exposed the difficulties of generating multiple equivalent offers. To solve the problem, they started from the idea of finding equivalent offers in the sense that they have the same utility value. The solution was to change the values of various attributes and adjust the price to balance the offer to the desired utility value. Finally, O. Kivanc has shown in [29] that when MESO is applied, the customer's utility increases significantly without decreasing the buyer's utility. To generate multiple offers in the automated negotiations, he has utilized graph-based statistical clustering to partition the space of the offers .

Based on these considerations, it can be stipulated that in an environment of an open business Grid, like that considered in the *AiG* project, simultaneous equivalent offers could increase the chances to reach success in negotiations between team leaders and a new worker (team joining scenario), or between team leaders and a user that tries to contract a job execution (job executing scenario). In fact, a user or a new worker, will more likely be satisfied by one of the offers from the set proposed by the teams, which can reduce the number of rounds of negotiations.

Therefore, in what follows, we suggest how the *AiG* system can be modified to use simultaneous equivalent offers, instead of the basic single offer iterative strategy.

4. Multiple Equivalent Simultaneous Offers strategy in the *AiG* system.

4.1. Generating multiple equivalent offers in the *AiG*. The *AiG* approach uses autonomous *SLA* negotiations, facilitated by agents representing Grid resource providers and consumers. Using the MESO strategy in the project makes us ponder the necessary changes in the team joining and the job execution scenarios. Currently, when the *LAgent* sends a *CFP* to a list of teams, each one of them answers with a single offer. However, when the MESO approach is to be used, each response would contain a set of offers (Fig. 4.1) covering the space of possible deals acceptable to the given team. Note that this would, among others, require modification the ontology so that one message could contain several offers.

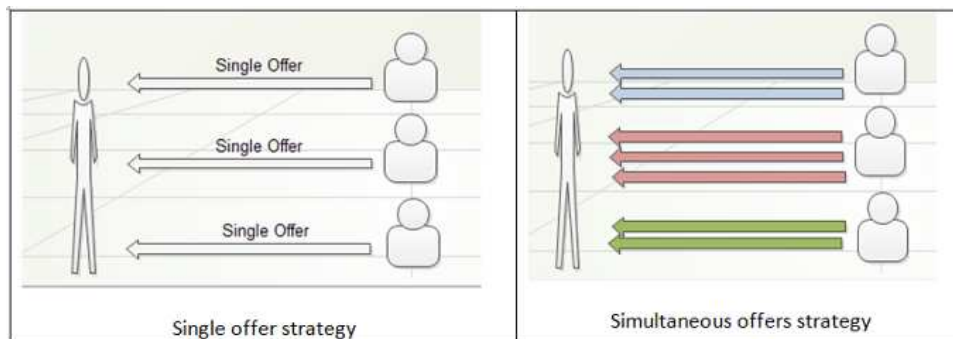


FIG. 4.1. Single offer strategy and simultaneous offers strategy.

Let us now see in details how offers can be prepared in an example of the job execution scenario: user is looking for a team to commission job to be executed on a resource that should run Windows Vista SP2 operating system. He would like to meet the deadline 20 of September 2013, with a penalty of at least 10 monetary units for the delay of contract completion. Furthermore, the maximum price that can be charged for the job is 100 monetary units. First, our user would specify resource requirements using the ontology-driven GUI (see, [34]). These requirements would be forwarded to the *LAgent* that would communicate with the *CIC* to find teams that satisfy them. The user would also provide its *LAgent* with contract conditions that would be used in negotiations. In such negotiations, first, a *CFP* message is sent by the *LAgent* to the *LMasters* of the selected teams. Note that all information exchanged in the system (including requirements and contract

offer) is ontologically demarcated. In this example, we use three negotiation parameters: deadline penalty, job execution time line, and payment conditions with three properties: (i) delay penalty, (ii) fixed utilization price, and (iii) lead time. For each of these parameters the user specifies its importance by assigning weight that are later used in the MCA.

The following snippet shows declaration of selected properties, classes and a simple class expression with restrictions on the contract, where the required resource should run Windows Vista SP2 operating system, the deadline penalty should be at least 10 , the fixed resource utilization price less than 100 and the execution job should meet the deadline of 20 of September 2013:

```
<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#contractedResource -->
<owl:ObjectProperty
rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#contractedResource">
  <rdfs:range rdf:resource="&cgo;GridComponent"/>
  <rdfs:domain>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <rdf:Description
        rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#JobExecutionConditions"/>
      <rdf:Description
        rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#WorkerContractConditions"/>
    </owl:unionOf>
  </owl:Class>
</rdfs:domain>
</owl:ObjectProperty>

<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#fixedUtilizationPrice -->

<owl:ObjectProperty
rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#fixedUtilizationPrice">
  <rdfs:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment rdf:datatype="&xsd:string">
    A price for the availability of a resource paid irrespective of offered/used ServiceChargableItems.
  </rdfs:comment>
  <rdfs:domain
    rdf:resource="http://gridagents.sourceforge.net/AiGConditionsOntology#PaymentConditions"/>
  <rdfs:range rdf:resource="http://gridagents.sourceforge.net/AiGConditionsOntology#Pricing"/>
</owl:ObjectProperty>

<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#jobExecutionTimeline -->

<owl:ObjectProperty
rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#jobExecutionTimeline">
  <rdfs:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain
    rdf:resource="http://gridagents.sourceforge.net/AiGConditionsOntology#JobExecutionConditions"/>
  <rdfs:range rdf:resource="&time;TemporalEntity"/>
</owl:ObjectProperty>

<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#paymentConditions -->

<owl:ObjectProperty
rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#paymentConditions">
  <rdfs:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:range
    rdf:resource="http://gridagents.sourceforge.net/AiGConditionsOntology#PaymentConditions"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
  <rdfs:domain>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <rdf:Description
        rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#JobExecutionConditions"/>
      <rdf:Description
        rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#WorkerContractConditions"/>
    </owl:unionOf>
  </owl:Class>
</rdfs:domain>
</owl:ObjectProperty>

<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#deadlinePenalty -->

<owl:DatatypeProperty
rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#deadlinePenalty">
  <rdfs:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain
    rdf:resource="http://gridagents.sourceforge.net/AiGConditionsOntology#JobExecutionConditions"/>
  <rdfs:range rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>

<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#offPeakTimePrice -->

<owl:DatatypeProperty
rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#offPeakTimePrice">
  <rdfs:domain
    rdf:resource="http://gridagents.sourceforge.net/AiGConditionsOntology#Pricing"/>
  <rdfs:range rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>

<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#JobExecutionConditions -->

<owl:Class
rdf:about="http://gridagents.sourceforge.net/AiGConditionsOntology#JobExecutionConditions"/>

<!-- http://gridagents.sourceforge.net/AiGConditionsOntology#PaymentConditions -->
```

```

<owl:Class
rdf:about="http://gridagents.sourceforge.net/AIGConditionsOntology#PaymentConditions"/>
<!-- http://gridagents.sourceforge.net/AIGConditionsOntology#Pricing -->
<owl:Class rdf:about="http://gridagents.sourceforge.net/AIGConditionsOntology#Pricing"/>
<!-- http://www.owl-ontologies.com/unnamed.owl#GridComponent -->
<owl:Class rdf:about="&cgo;GridComponent"/>
<!-- http://www.w3.org/2006/time#TemporalEntity -->
<owl:Class rdf:about="&time;TemporalEntity"/>

```

The *CFP* is equivalent to:

```

JobExecutionConditions that contractedResources some
(WorkerNode that isRunningOS value vista,sp2) and paymentConditions some
(PaymentConditions that fixedUtilizationPrice some (Pricing that peakTimePrice some float[<= 100]))
and deadlinePenalty some float[< 10] and jobExecutionTimeline some dateTime[=2013-09-20T23:59:59]

```

In response to such *CFP*, the *LMasters* may respond with job execution contract offers. Contract offer presented below specifies the fixed utilization price to be 100, the deadline penalty to be 10 and the job deadline to be 20 of September 2013. In what follows, we represent this offer as: $Offer1 = (100, 10, 20/09/2013)$.

```

<owl:NamedIndividual rdf:ID="Contract">
  <deadlinePenalty rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >10.0</deadlinePenalty>
  <rdf:type rdf:resource="#JobExecutionConditions"/>
  <jobExecutionTimeline>
    <owl:NamedIndividual rdf:ID="Sept20">
      <time:inXSDDateTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
      >2013-09-20T23:59:59</time:inXSDDateTime>
      <rdf:type rdf:resource="http://www.w3.org/2006/time#Instant"/>
    </owl:NamedIndividual>
  </jobExecutionTimeline>
  <paymentConditions>
    <PaymentConditions rdf:ID="PaymentConditions_1">
      <fixedUtilizationPrice>
        <Pricing rdf:ID="Pricing_2">
          <offPeakTimePrice rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
          >100.0</offPeakTimePrice>
        </Pricing>
      </fixedUtilizationPrice>
    </PaymentConditions>
  </paymentConditions>
</owl:NamedIndividual>

```

As mentioned before, in the multiple-offer approach there is more than one offer to be made by each team. However, it is assumed all of them have the same utility. Thus, we have to evaluate *Offer1*, which combines multiple attributes. According to the multi-attribute utility theory (MAUT; [30]), each team has a (private) utility function $(U_p(v), U_d(v), U_j(v))$ for each criterion (*fixedUtilizationPrice*, *deadlinePenalty* and *jobExecutionTimeline* respectively). The utility function expresses the team's evaluation of each value in the criterion's domain.

To define the attribute's utility function we refer to [31]. For each attribute, the team chooses a minimum (*a*), a maximum (*c*) and a best (*b*) values. On their basis we can distinguish three cases:

- (i) In many situations, when the minimum is considered the best, appropriate equation is as follows:

$$U(v) = \begin{cases} 1 & , v \leq a \\ (v - c)/(a - c) & , a < v < c \\ 0 & , v \geq c \end{cases} \quad (4.1)$$

- (ii) When the maximum value is considered the best, the utility function can be defined as:

$$U(v) = \begin{cases} 0 & , v \leq a \\ (c - v)/(c - a) & , a < v < c \\ 1 & , v \geq c \end{cases} \quad (4.2)$$

(iii) Finally, when the best utility is placed in between the minimum and the maximum, the following equation is considered to be the most adequate:

$$U(v) = \begin{cases} (v - a)/(b - a) & , a \leq v \leq b \\ (v - c)/(b - c) & , b \leq v \leq c \\ 0 & , \text{Otherwise} \end{cases} \quad (4.3)$$

Now we must look at the overall offer. In our example, weight is assigned to each criteria, and W_p , W_d and W_j are the weights for *fixedUtilizationPrice*, *deadlinePenalty* and *jobExecutionTimeline*, respectively. Note here that those weights are different from those of the user; they concern teams' assessment of importance of each criterion. Here, the method of assigning weights is out of scope of this paper.

The combination of attribute values in the offer gives a particular utility value according to the utility formula expressed as follows:

$$Utility(offer1) = U_p(100) \times W_p + U_d(10) \times W_d + U_j(20/09/2013) \times W_j \quad (4.4)$$

Recall that offers equivalent to the previous ones (but with a different attribute combination), should have the same utility value for the team:

$$Utility(offer1) = Utility(offer2) \quad (4.5)$$

To solve this problem, we have decided to use the method proposed in [28]. The idea is to change the values of various attributes and to balance to the same utility value of the first offer by adjusting the cost.

In our case we suggest to apply random values dR for the *deadlinePenalty* and jR for *jobExecutionTimeline*. Thus we have:

$$Utility(offer2) = U_p(x) \times W_p + U_d(dR) \times W_d + U_j(jR) \times W_j \quad (4.6)$$

where x is the value of the price that we have to look for, in order to satisfy equation (4.3).

Then we deduce that:

$$U_p(x) = \frac{Utility(offer1) - U_d(dR) \times W_d - U_j(jR) \times W_j}{W_p} \quad (4.7)$$

Therefore, generating another offer means finding a price that brings to the team the utility expressed in equation (4.5).

We can generate additional offers by adopting the same method. Note that this method is also suitable for the team joining scenario, for instance by considering in the offer the following attributes: *contract Period*, *Availability*, *guaranteed utilization* and *payment*.

The next section presents messages exchanged in the *AiG* negotiations process that implement the above approach. These messages will be exchanged in the context of a modified FIPA Iterated contract net protocol.

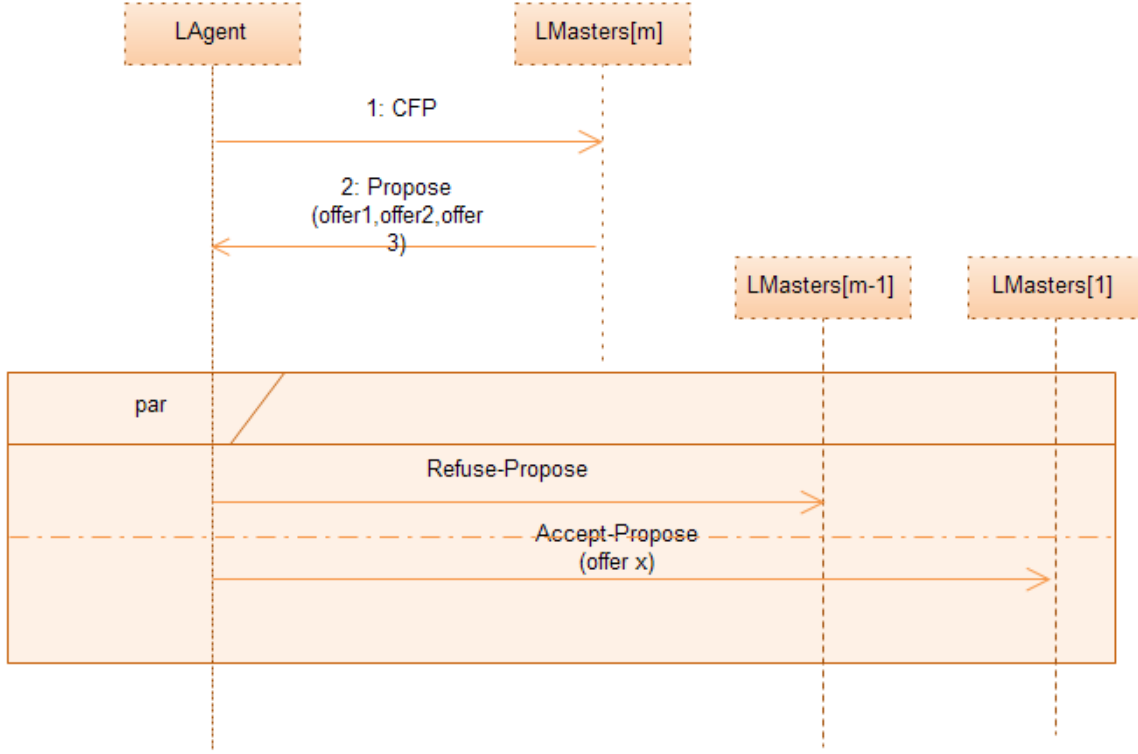
4.2. MESOs modeling. Following the guidelines found in [1], we fix the number of offers generated by teams to no more than three.

There are two possibilities of handling the MESO in the *AiG*, depending on the possibility to assemble offers in a single message. Note that we initially illustrate the different approaches using a simplified version of the protocol – FIPA Contract Net Protocol. Only later, we present the adopted solution in the context of the Iterated FIPA Contract Net Protocol.

4.2.1. Single Propose message including multiple offers. Here, a *Propose* message can contain more than one offer, and each *LMaster* sends one *Propose* message with a list of offers included. The *Accept-Proposal* sent to the *LMaster* of a chosen team contains reference to the accepted offer (Fig. 4.2).

4.2.2. A separate Propose message for each offer. In the second approach, each *Propose* message contains a single offer. The *LMaster* of the selected team receives an *Accept-Proposal* message for the accepted offer and the *Refuse-Proposal* messages for the remaining offers (Fig. 4.3).

4.2.3. Proposed solution. To generate a smaller number of messages, and because handling a set of offers jointly is more clear conceptually and easier practically, we opt for the first solution, where all offers are sent in a single message. Here, the response of the *LAgent*, will be unique and will use the already implemented solution, with the only difference that it will specify (in the *Accept-Proposal* message) the offer that was selected, among the received ones. However, as a result, we need to change our ontology, by adding an identifier to the

FIG. 4.2. One *Propose* with all offers.

offer accepted for the contract (in the *Accept-Proposal* message). The *Propose* message has also to be changed, to allow it to contain a list of offers.

Fig. 4.4 presents a sequence diagram for the modified Iterated FIPA Contract Net Protocol that is to be utilized in the *AiG* project to support multiple offers. The modified messages exchanged in the team joining negotiation scenario, which are very similar to the job execution scenario, are as follows:

1. Teams submit to the *CIC* characteristics of their potential workers.
2. The *LAgent*, representing the user, sends an ACL message of type *CFP*, that contains the description of worker's resource (resource that it is willing to offer e.g. available software, operating system, processor type, etc.) and constraints concerning contract terms, such as the contract length.
3. A message of type *Propose* sent by the *LMaster* containing details of the multiple (though no more than three) contracts offers it is proposing to the *LAgent*.

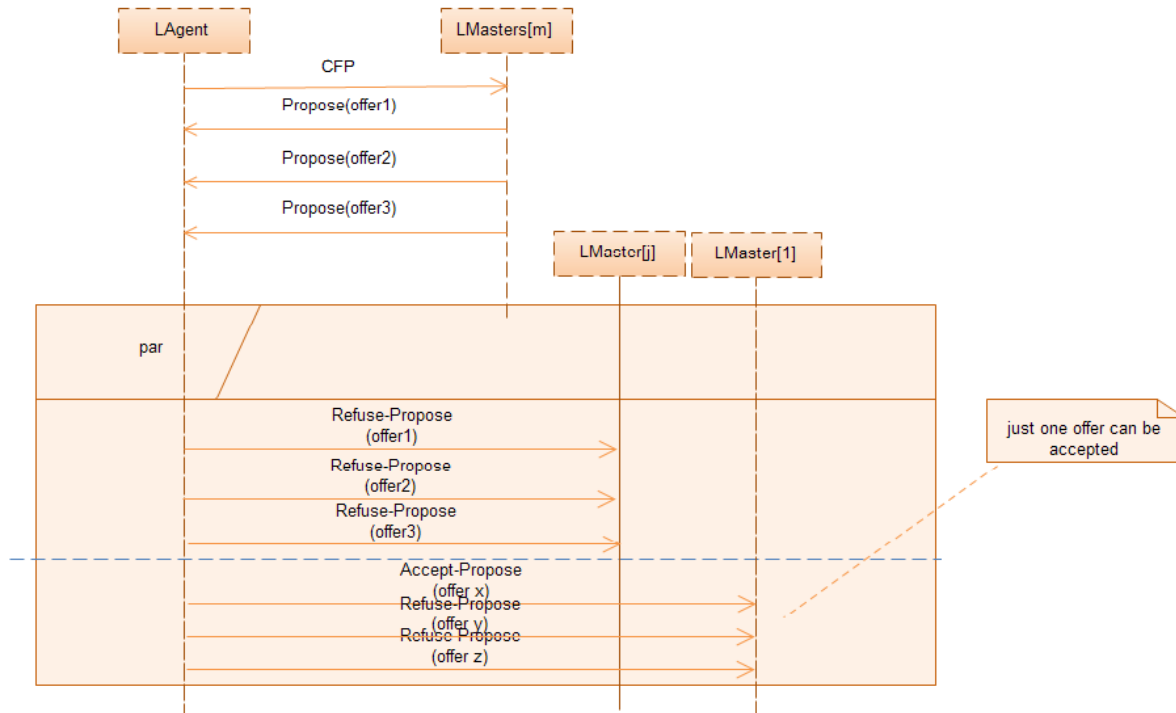
If it's the last round:

- (i) Messages of type *Accept-Proposal* and *Refuse-Proposal* are sent respectively to the *LMaster* of the selected offer (specifying, which offer was accepted), and to the remaining *LMaster* agents,
- (ii) *Inform-Done* message sent by the *LMaster* that received the *Accept-Proposal*.

If this is not the last round, revised *CFP* is send to a selected subset of *LMasters*, and rejections to the others.

5. Concluding remarks. Multiple equivalent simultaneous offers is a novel strategy that begins to play an important role in the field of economics. A project that will include an economic aspect, such as the *AiG*, should consider using this technique, in order to benefit from its advantages.

The goal of this paper was to discuss how to integrate the MESOs in a multi-attribute negotiation within an agent-based Grid resource management system. Thus, we had to consider generating a package of offers in an environment where no information about negotiator preferences is available. Since this is the first attempt

FIG. 4.3. a *Propose* message and a response for each offer.

at integrating the multiple equivalent offers strategy into negotiations in the *AiG* project, there are still some adaptations to be made, especially to the ontology. Also, we should note that this strategy requires a complex preparations prior to the negotiation, and a question that remains open is, will it be always possible to generate multiple offers. Filling these gaps and implementing the approach are our next research goals. The implementation will help us to compare MESO with single strategy and to evaluate experimentally what happens the number of simultaneous offers is changing.

REFERENCES

- [1] V. H. MEDVEC, L. GEOFFREY, A. D. GALINSKY, AND C. S. ALETHA, *Choice and Achievement at the Bargaining Table: The Distributive, Integrative, and Interpersonal Advantages of Making Multiple Equivalent Simultaneous Offers*, IACM 18th Annual Conference, June 2005. Available at SSRN: <http://ssrn.com/abstract=732665> or <http://dx.doi.org/10.2139/ssrn.732665>.
- [2] MANDELBAUM AND ROBB, *How to Negotiate Effectively*, INC Magazine. <http://www.inc.com/magazine/20101101/how-to-negotiate-effectively.html>. Retrieved 27 September 2012.
- [3] V. H. MEDVEC AND A. D. GALINSKY, *Putting More on the Table: How Making Multiple Offers Can Increase the Final Value of the Deal*, Harvard Business Review. <http://hbr.org/product/putting-more-on-the-table-how-making-multiple-offe/an/N0504B-PDF-ENG>.
- [4] V. H. MEDVEC AND A. D. GALINSKY, *The Value of Making Several Offers in Business Negotiations*, <http://www.pon.harvard.edu/daily/business-negotiations/the-value-of-making-several-offers-in-financial-negotiations/>. Retrieved 27 September 2012.
- [5] Y. YINGPING, S. SHARAD AND X. YUNJIE (CALVIN), *Offer with Choices and Accept with Delay: A Win-Win Strategy Model for Agent Based Automated Negotiation*, (2009). ICIS 2009 Proceedings. Paper 180. <http://aisel.aisnet.org/icis2009/180>.
- [6] D. OUELHADJ, J. GARIBALDI, J. MACLAREN, R. SAKELLARIOU, AND K. KRISHNAKUMAR, *A multi-agent infrastructure and a service level agreement negotiation protocol for robust scheduling in grid computing*, In Proceedings of the 2005 European Grid Computing Conference (EGC 2005), February 2005, p. 651–660.
- [7] M. PARKIN, P. HASSELMEYER, B. KOLLER AND P. WIEDER, *An SLA Re-negotiation Protocol*, Proceedings of the 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC08), CEUR Workshop Proceedings, ISSN 1613-0073, Volume 411. Dublin, Ireland. November 2008.
- [8] A. ANDRIEUX, *Web Services Agreement Specification (WS-Agreement)*, Proposed Recommendation, Open Grid Forum,

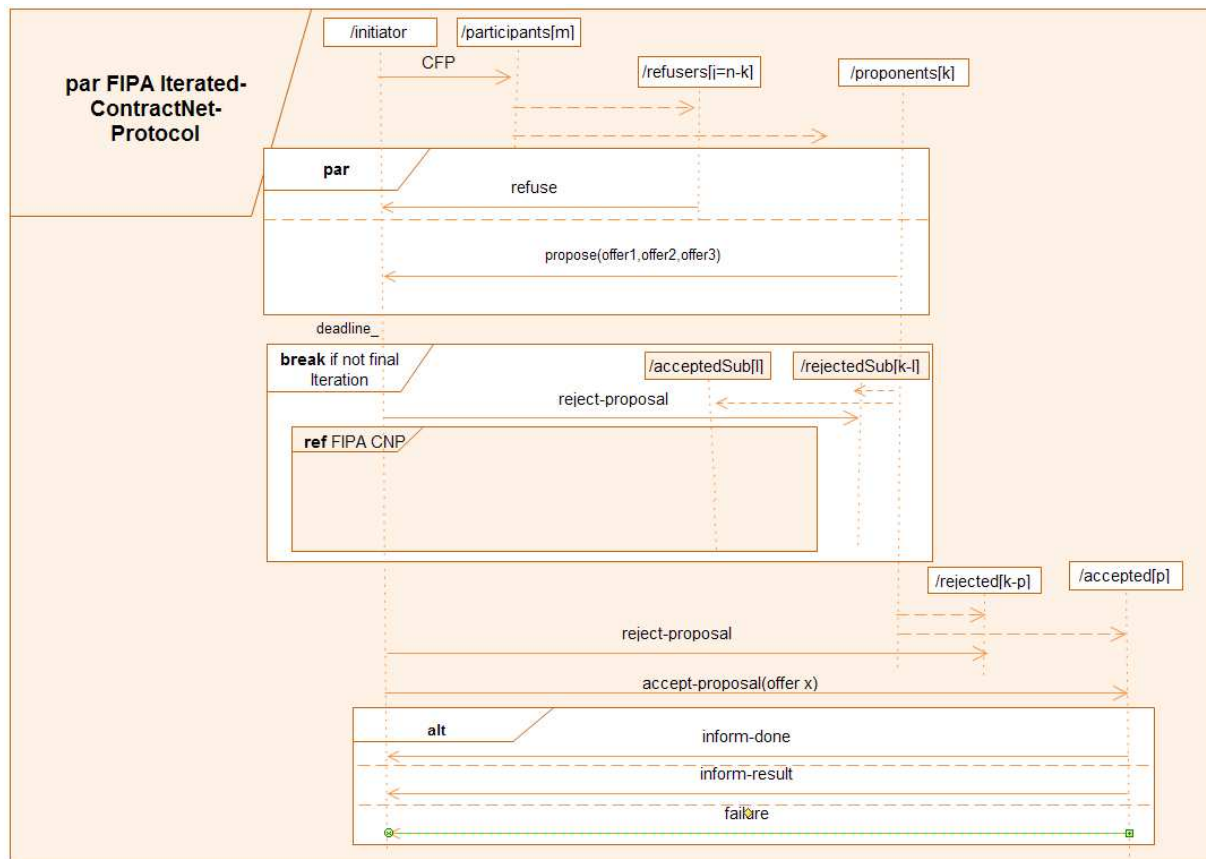


FIG. 4.4. Modified FIPA Iterated Contract Net Protocol.

- September 2006. Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG).
- [9] G. D. MODICA, V. REGALBUTO, O. TOMARCHIO AND L. VITA, *Enabling re-negotiations of SLA by extending the WS-Agreement specification*, IEEE International Conference on Services Computing (SCC 2007), pp. 248–251.
 - [10] CoreGRID European Research Network on Foundations, software infrastructures and applications for large scale distributed, GRID and peer to peer technologies, <http://www.coregrid.net/>
 - [11] NextGRID: Architecture for next generation Grids, <http://www.nextgrid.org/>
 - [12] BREIN Business objective driven Reliable and Intelligent Grids for Real BussiNess, <http://www.eu-brein.com/>
 - [13] Grid Resource Allocation Agreement WG (GRAAP WG), <http://www.ogf.org/gf/group-info/view.php?group=graap-wg-arbitrary>
 - [14] J. HENDLER, *Agents and the semantic web*, IEEE Intelligent Systems, 16(2), pp. 30-37.
 - [15] I. FOSTER, N. R. JENNINGS AND C. KESSELMAN, *Brain meets brawn: Why grid and agents need each other*, Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Vol. 1, 2004, pp. 8–15.
 - [16] W. KURANOWSKI, M. GANZHA, M. GAWINECKI, M. PAPRZYCKI, I. LIRKOV, AND S. MARGENOV, *Forming and managing agent teams acting as resource brokers in the grid-preliminary considerations*, International Journal of Computational Intelligence Research, Vol. 4, No. 1, 2008, pp. 9–16.
 - [17] W. KURANOWSKI, M. GANZHA, M. PAPRZYCKI AND I. LIRKOV, *Supervising agent team an agent-based grid resource brokering system-initial solution*, Proceedings of the Conference on Complex, Intelligent and Software Intensive Systems, 4-7 March 2008, pp. 321–326.
 - [18] M. DROZDOWICZ, M. GANZHA, M. PAPRZYCKI, R. OLEJNIK, I. LIRKOV, P. TELEGIN, M. SENOBARI, *Ontologies, Agents and the Grid: An Overview*, in B.H.V. Topping, P. Ivnyi, (Editors), "Parallel, Distributed and Grid Computing for Engineering", Saxe-Coburg Publications, Stirlingshire, UK, Chapter 7, 2009, pp. 117–140.
 - [19] Core grid ontology. <http://grid.ucy.ac.cy/grisen/cgo.owl>
 - [20] W. XING, M. D. DIKAIKOS, R. SAKELLARIOU, S. ORLANDO, AND D. LAFORENZA, *Design and Development of a Core Grid Ontology*, In Proc. of the CoreGRID Workshop: Integrated research in Grid Computing, 2005, pp. 21–31.
 - [21] M. DROZDOWICZ, M. GANZHA, M. PAPRZYCKI, K. WASIELEWSKA, I. LIRKOV, R. OLEJNIK, N. ATTAOUI, *Utilization of Modified Core GRID Ontology in an Agent-based Grid Resource Management System*, PROCEEDINGS OF THE CATA 2010 CONFERENCE.

- [22] M. DROZDOWICZ, K. WASIELEWSKA, M. GANZHA, M. PAPRZYCKI, N. ATTAOUI, I. LIRKOV, R. OLEJNIK, D. PETCU AND C. BADICA, *Chapter Ontology for Contract Negotiations in Agent-based Grid Resource Management System*, IN P. IVNYI, B.H.V TOPPING,(EDITORS), TRENDS IN PARALLEL, DISTRIBUTED, GRID AND CLOUD COMPUTING FOR ENGINEERING, SAXE-COBURG PUBLICATIONS, STIRLINGSHIRE, UK, 2011.
- [23] K. WASIELEWSKA, M. DROZDOWICZ, M. GANZHA, M. PAPRZYCKI, N. ATTAOUI, D. PETCU, C. BADICA, R. OLEJNIK, I.LIRKOV, *Negotiations in an Agent-based Grid Resource Brokering Systems*, IN P. IVNYI, B.H.V TOPPING,(EDITORS), "TRENDS IN PARALLEL, DISTRIBUTED, GRID AND CLOUD COMPUTING FOR ENGINEERING", SAXE-COBURG PUBLICATIONS, STIRLINGSHIRE, UK, 2011.
- [24] M. PARKIN, R. BADIA, J. MARTRAT, *A Comparison of SLA Use in Six of the European Commissions FP6 Projects*, INSTITUTE ON RESOURCE MANAGEMENT AND SCHEDULING, COREGRID-NETWORK OF EXCELLENCE, APRIL 2008, PP. 1–38.
- [25] M. DOMINIAK, W. KURANOWSKI, M. GAWINECKI, M. GANZHA, M. PAPRZYCKI, *Efficient Matchmaking in an Agent-based Grid Resource Brokering System*, PROCEEDINGS OF THE INTERNATIONAL MULTICONFERENCE ON COMPUTER SCIENCE AND INFORMATION TECHNOLOGY, 2006, PP. 327-335.
- [26] FIPA ITERATED CONTRACT NET INTERACTION PROTOCOL SPECIFICATION, [HTTP://WWW.FIPA.ORG/SPECS/FIPA00030/INDEX.HTML](http://www.fipa.org/specs/fipa00030/index.html)
- [27] FIPA CONTRACT NET PROTOCOL SPECIFICATION, [HTTP://WWW.FIPA.ORG/SPECS/FIPA00029/SC00029H.HTML](http://www.fipa.org/specs/fipa00029/SC00029H.HTML).
- [28] NEUBERT, R. GORLITZ, O. TEICH AND TOBIAS, *Automated negotiations of supply contracts for flexible production networks*, INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS, ELSEVIER, VOL. 89(2), PP. 175–187, MAY 2004.
- [29] K. OZONAT, *Multiple Offer Strategy for Automated Negotiation Agents*, [HTTP://WORLD-COMP.ORG/P2011/ICA3527.PDF](http://world-comp.org/p2011/ICA3527.pdf)
- [30] M. BARBUCEANU AND W. LO *A Multi-Attribute Utility Theoretic Negotiation Architecture for Electronic Commerce*, PROC. 4TH INT. CONF. ON AUTONOMOUS AGENTS, BARCELONA, CATALONIA, SPAIN (2000).
- [31] A. D. YOUNGBLOOD AND T. R. COLLINS, *Addressing balanced scorecard trade-off issues between performance metrics using multi attribute utility theory*, ENGINEERING MANAGEMENT JOURNAL, MAR 2003; VOL. 15, N1; PP. 11–17.
- [32] SETI@HOME WEBSITE. [HTTP://SETIATHOME.SSL.BERKELEY.EDU](http://setiathome.ssl.berkeley.edu)
- [33] C. BARTOLINI, C. PREIST, AND N. R. JENNINGS, *A software framework for automated negotiation*, IN SOFTWARE ENGINEERING FOR MULTI-AGENT SYSTEMS III: RESEARCH ISSUES AND PRACTICAL APPLICATIONS, SPRINGER VERLAG 2005, LNCS 3390: PP. 213–235.
- [34] M. DROZDOWICZ, M. GANZHA, M. PAPRZYCKI, P. SZMEJA, AND K. WASIELEWSKA, *OntoPlay - A Flexible User-Interface for Ontology-based Systems*. AT, VOLUME 918 OF CEUR WORKSHOP PROCEEDINGS CEUR-WS.ORG, (2012), PP. 86–100.
- [35] C. BADICA, S. ILIE, M. KAMERMANS, G. PAVLIN, A. PENDERS, M. SCAFES, *Multi-Agent Systems, Ontologies and Negotiation for Dynamic Service Composition in Multi-Organizational Environmental Management*. NATO SCIENCE FOR PEACE AND SECURITY SERIES - D: INFORMATION AND COMMUNICATION SECURITY, VOLUME 32: SOFTWARE AGENTS, AGENT SYSTEMS AND THEIR APPLICATIONS, (2012), PP. 286–306. [HTTP://DX.DOI.ORG/10.3233/978-1-60750-818-2-286](http://dx.doi.org/10.3233/978-1-60750-818-2-286)

Edited by: Viorel Negru and Daniela Zaharie

Received: June 1, 2013

Accepted: Jul 9, 2013