A Generic Ontology of Rational Negotiation

Vadim Ermolayev, Natalya Keberle

Department of IT,
Zaporozhye National University
Zhukovskogo 66, 69063, Zaporozhye, Ukraine
vadim@ermolayev.com, kenga@zsu.zp.ua

Abstract: The paper presents the generic conceptual framework for modeling rational negotiation processes. The central piece of the presented work is the upper-level negotiation ontology. This ontology is designed to be capable to model the arbitrary types of negotiation dialogues by providing the upper-level namespace for that. The models of the specific negotiation types refine this namespace by providing the instances of its concepts, applying additional restrictions on the properties, possibly adding new subclasses and properties. The approach to model negotiations with the Generic Negotiation Ontology (GNO) is initially evaluated by applying to contracting negotiation type. The choice of the negotiation type for the evaluation is suggested by the fact that contracting negotiations are the typical encounters among the agents taking part in the simulation of a Dynamic Engineering Design Process (DEDP). This modeling activity is performed in the frame of our PSI project[1].

1 Introduction

People rarely do things on their own. On the contrary, when people act they normally seek for at least the understanding or the support and, more often, for the back-up or the assistance by other people. For example, having a dinner alone is of course a useful action. However, having a dinner with a mate is definitely more inspiring – the action is supported by the mate, you may combine eating and discussing the paintings by Monet[2]… In a word, it is more preferred or, from the game-theoretical point of view, brings more utility. That is why people like to transform actions to encounters, to involve other people, to make the actions social. Social activities evidently require the means to coordinate the actions of the participants. These means are known as dialogues. It is important that people engaged in social actions understand the dialogues they use coherently – commit themselves to the certain configurations or the settings of the dialogues. People develop software systems because they want to delegate some actions to them. Therefore, software systems are designed in a way to model human activities. That is why it is critically important to design the dialogues among the components of a distributed software system in a way that each autonomous software component commits itself to the standardized rules and settings of this encounter. These rules and settings may of course be hard-wired in the software. However, if a software system is open and

[1] Performance Simulation Initiative (PSI) is the project of Cadence Design Systems, GmbH.
[2] http://www.ibiblio.org/wm/paint/auth/monet/

comprises heterogeneous components, the hard-wiring approach is hardly feasible. It may be more rational to design the pieces of the software committing to the standardized explicitly defined ontology which explicitly describes the possible dialogues, as for example proposed in [EKT02], [Ta05]. Walton and Krabbe's typology [WK95] distinguishes the following six dialogue types which differ by their pragmatic content: persuasion, negotiation, inquiry, information seeking, deliberation, and eristic. In this paper we focus on negotiation because of the following reasons:

– "… perhaps the most fundamental and powerful mechanism for managing inter-agent dependencies at run-time is *negotiation* — the process by which a group of (software) [3] agents comes to a mutually acceptable agreement on some matter." (c.f. [Je01])
– In PSI we focus on the simulation of the optimal performance of a DEDP which require negotiations – i.e. reaching agreements among the members of a design team on several matters, for example, who will perform this or that task.

The contribution of this paper is the proposal of the Generic Ontology of Negotiation (GNO). We plan that GNO will be used as the upper-level descriptive framework to standardize various types of negotiation dialogues used by software agents in PSI. We also anticipate that GNO may also be used in a much wider scale. For example, its expressive power allows describing negotiation dialogues for meaning negotiation [Er05a] – a very important mechanism for solving interoperability problems in distributed information retrieval. The reminder of the paper is structured as follows. Section 2 specifies the requirements and defines the key concepts of GNO. Section 3 reports on the initial evaluation of GNO by modeling contracting negotiations on out-sourcing tasks. Section 4 places the reported work in the context of the contributions of the colleges. Section 5 summarizes the results and reports on our plans for the future work.

2 The Generic Negotiation Ontology (GNO)

Let's suppose that the upper-level negotiation ontology is developed. The next step is to gain the commitment to this ontology by the designers of the software agents which will actually negotiate according to the specified rules of encounter. To make this commitment more attractive we need to (a) specify the ontology in a standard language; (b) design the ontology in a way to reserve as much flexibility as possible to the users. Our approach to solve (a) is to use OWL DL[4] + SWRL[5] for ontology and rule specification. For solving (b) we design GNO adhering to the Gruber's principle of the minimal ontological commitment [Gr93].

The reminder of this section verbally denotes the basic concepts which describe the

[3] The text in brackets is ours.
[4] OWL DL – the sub-set of the Web Ontology Language (OWL) based on the Description Logic. OWL Reference: http://www.w3.org/TR/owl-ref/
[5] Semantic Web Rule Language (SWRL): http://www.w3.org/Submission/SWRL/.

process of reaching agreements – i.e., a Negotiation Process[6]. The denoted concepts and the descriptive theory of their context [Er05a] are illustrated by the UML diagram[7] of the Generic Negotiation Ontology in Fig. 1. This ontology is than used to model specific types of negotiations. This usage is illustrated in section 3 by applying the ontology to modeling the Extended Iterative Contracting Negotiation dialogue [Er04a], [Er04b].

Definition 1: *Negotiation Process*. A Negotiation Process is the encounter in which several Negotiation Parties persuade the goal of reaching a common agreement on the Negotiation Issues comprising the Negotiation Set. The parties do it: (a) by applying their private rules to behave rationally; (b) by exchanging messages – uttering the performatives of the common communication language. These messages communicate offers, counter-offers, arguments, counter-arguments as their content. The Negotiation Process is governed by the Negotiation Mechanism. Negotiation Parties are committed to the Negotiation Mechanism.

The set of characteristics describing a Negotiation (type) and distinguishing it from another Negotiation (type) will further on be referred to as a *negotiation setting*. Negotiation Parties first need to agree upon and to commit to the Negotiation Mechanism and then use it for their negotiation.

Definition 2: *Negotiation Party*. Negotiation Parties are the participants of the Negotiation Process. Each party consists of an Actor (modeled by a software agent) having its own beliefs about the other parties. Negotiation Parties are assumed to be *rational*. Rationality in this context means that a party will not act in a way or agree upon an outcome which lowers its utility, even despite the fact that this action or outcome may bring extra utility to the other Parties. A Negotiation Party plays the role in the Negotiation Process. Negotiation Roles frame out the goals and the strategies of the parties.

Definition 3: *Utility*[8]. A Utility represents the motivations of Negotiation Parties. A utility function for a given party assigns a number for every possible Negotiation Outcome where a higher number implies that the outcome is more preferred. Utility functions may be either ordinal in which case only the relative rankings are important, but no quantity is actually being measured, or cardinal. Utilities are used in Negotiation Strategies to reason about possible Negotiation Outcomes.

Definition 4: *Negotiation Role*. A Negotiation Role models the behavior pattern of an abstract Permissible Participant of a Negotiation Process. A Negotiation Role shapes out the configuration of the party's Negotiation Strategy (or several strategies) and the attitudes to the other parties. These attitudes are often formulated in the terms of their pre-disposition to collaboration with the other parties. Of course a party (playing one role) is open to the collaboration with another party (playing another role) if their goals are coherent in some sense.

---

[6] Here and further on in the paper we highlight the concept names by capitalizing the first letters.
[7] The diagram is built in Argo UML v.0.20 beta: http://argouml.tigris.org/
[8] This definition follows the Game-Theoretical definition of a utility.

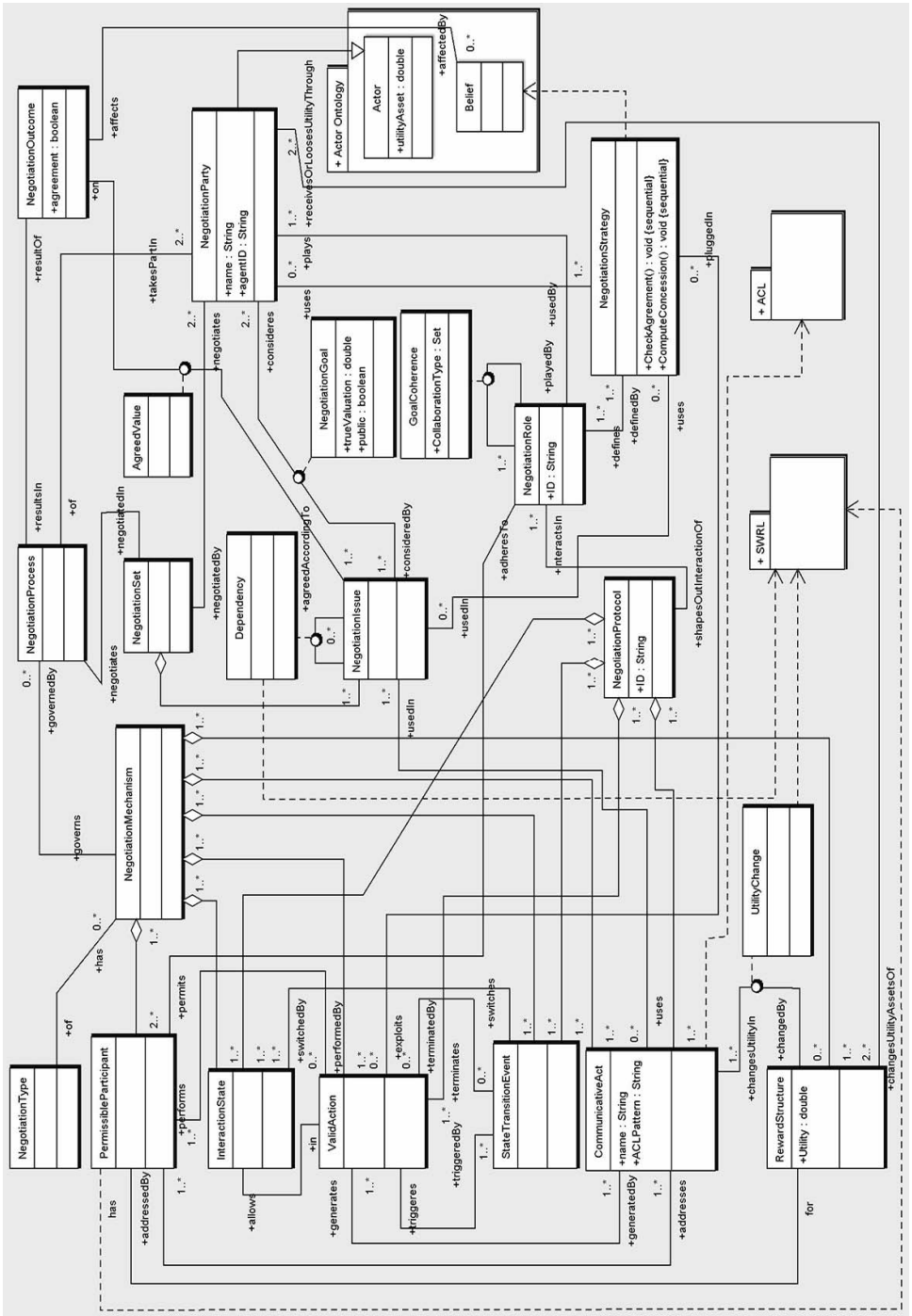


Fig.1.: The UML Diagram of the Generic Negotiation Ontology

Definition 5: *Negotiation Goal*. A Negotiation Goal is the goal pursued by a Negotiation Party in the Negotiation Process. In an abstract sense a Negotiation Goal is the state of affairs which: (a) *may be reached* in the Negotiation Process; (b) is specified in the terms of the *true valuations* of the Negotiation Issues comprising the Negotiation Set; and (c) *is most preferred* by the party among the other possible states of affairs (please refer to *Utility*). The specification of the goal is influenced by the Negotiation Role the party plays in the Negotiation Process.

In some negotiation settings the goals are not revealed to public (the other parties) and remain private. The example of such a setting is the Classical English auction[9]. In other settings the true valuations of the Negotiation Issues may be revealed through the bids. The example of such a setting is the Vickrey auction[10]. The goals of the different parties playing different roles may be *coherent*, *neutral*, or *conflicting* with respect to the goals of the opponents. A goal g_A of the party A is coherent to the goal g_B of the party B if they are mutually facilitating – i.e., the closer A is to reaching g_A implies that B is closer to reach its g_B as well. The distance between the goal and the current state of affairs in a Negotiation Process is evidently measured by the difference in utilities. A goal g_A of the party A is neutral to the goal g_B of the party B if the movement of A towards g_A does not affect the movement of B towards g_B. Finally, a goal g_A of the party A is conflicting to the goal g_B of the party B if the movement of A towards g_A affects negatively the movement of B towards g_B and vice versa – i.e. from the utilitarian viewpoint, reaching the state of affairs which is closer to g_A makes A better of at the expense of B which is worse of in this outcome.

Negotiation settings with respect to the parties involved may be *symmetric* and *non-symmetric*. Negotiation settings are symmetric with respect to the parties involved if their roles and the part of their beliefs revealed to the other parties is equivalent in some sense. An example of a symmetric negotiation setting is: two or more buyers negotiate on forming the group of buyers in order to get the wholesale discount. For example in the context of PSI a symmetric negotiation may occur when different designers discuss grouping around one Software Tool. It is also true to say that the parties of symmetric negotiations normally have *coherent* goals and their Negotiation Roles are *cooperative*. A non-symmetric negotiation setting assumes differences in the roles of the parties, the information they share with the others, or in both factors. For example in PSI a non-symmetric negotiation occurs when a Task Manager is willing to allocate a Task to a Designer [Go05] and negotiates with the group of possible Contractors [Er05b].

According to the number of parties in a negotiation the negotiation settings are classified as 1-to-1, 1-to-many, and many-to-many. A 1-to-1 negotiation occurs when 1 party negotiates with exactly 1 opponent. A 1-to-many setting stands for the situation when 1 representative of a role negotiates with the group(s) of the representatives of (an)other role(s). A negotiation setting is of type many-to-many if there are 2 or more parties represented by the groups of the participants playing different roles.

---

[9] See, e.g., http://www.gametheory.net/dictionary/Auctions/EnglishAuction.html

[10] Bidding true valuations is the dominant strategy in Vickrey auction – http://www.gametheory.net/dictionary/Auctions/VickreyAuction.html

With respect to the coherency in the goals Negotiation Parties may be characterized as *self-interested*, *neutral*, or *cooperative*. We shall say that the Parties are self-interested if the goals are conflicting. On the contrary, the parties are cooperative if some of their goals are *coherent* and the others are *neutral*. From the utilitarian viewpoint this means that there are the outcomes in which a party reaching its goal makes (an)other party(ies) better of as well. The parties are said to be *neutral* all their goal pairs are neutral.

Definition 6: *Negotiation Outcome*. A Negotiation Outcome is the result of the Negotiation Process. If negotiation is successful its outcome is the agreement on the Negotiation Issues comprising the Negotiation Set. This agreement is denoted as the set of the Agreed Values of the Negotiation Issues. If negotiation fails the outcome shows the gaps in the valuations of the Negotiation Issues between the last bids of the parties.

Definition 7: *Negotiation Set and Negotiation Issue*. A Negotiation Set is the set of matters or issues which are under negotiation in the given encounter. Negotiation Parties are trying to reach the agreement on the values of the Negotiation Issues in the Negotiation Set. A Negotiation Set may consist of 1 to several Negotiation Issues. A negotiation is said to be single-issue if the cardinality of its Negotiation Set is equal to 1. A negotiation is said to be multi-issue if the cardinality of its Negotiation Set is greater than 1.

Negotiation Issues may be dependent or independent of each other. An issue is said to be dependent on the other issues if the agreement on one of the latter affects the agreement on the former. An issue is independent from the other issues if the agreement on this issue doesn't affect the agreement on the other issues. If a Negotiation Issue is independent of the other issues in the Negotiation Set it can be negotiated separately. This supposition allows the decomposition of a multi-issue negotiation into the series of a simpler singe-issue Negotiations. However, such decomposition is impossible for the negotiations with the Negotiation Sets composed of dependent issues. Dependencies among Negotiation Issues may be formalized by rules coded in SWRL.

Definition 8: *Negotiation Mechanism*. A Negotiation Mechanism is the aggregation of the sets of its integral components that govern Negotiation Processes [DPJ03]. The sets are:

- *Permissible Participants*. Permissible Participants specify which Negotiation Roles are permitted to take part in the process. In an Auction the Permissible Participants may be buyers, sellers, or an auctioneer. The rules for the Permissible Participants constrain the permissions for the parties playing roles to undertake Valid Actions.
- *Interaction States*. Interaction States are the states of the parties in the negotiation process. At least the following four are the generic ones: initiate negotiation, perform a Valid Action, wait for a Communicative Act, terminate negotiation. These states may be refined in the specific negotiation types. For example, in an auction the following states may be valid: accepting bids (the refinement of performing a Valid Action), auction closed (the refinement of negotiation termination), etc.
- *State Transition Events*. State Transition Events cause the transitions of Interaction States. These events are either produced by the Valid Actions undertaken by the Permissible Participants – i.e. *internal* to the Negotiation Mechanism, or are *external*.

The examples of the internal events are a Communicative Act has been performed, a message has been received, a Valid Action has been accomplished. The example of an external event is: the auction is closed due to the electric power breakdown.

- *Valid Actions*. Valid Actions are the actions allowed to be undertaken by the Permissible Participants in a Negotiation Process. A Valid Action, when accomplished by a party, is followed by the performance of a Communicative Act which communicates the results of the Valid Action to the other parties. A Valid Action is the locus of the internal behavior of a Negotiation Party. This behavior becomes pro-active and flexible due to exploiting one or several Negotiation Strategies framed out by the Negotiation Role played by the party. Examples of a Valid Action for an auction are: compute the initial price, compute the bid, analyze the submitted bids.
- *Communicative Acts*. Communicative Acts are the means to construct the interaction (i.e. sending and receiving performative messages) among the Permissible Participants. It is assumed that the messages are coded in the FIPA Agent Communication Language (ACL) and standardized according to the FIPA Communicative Act Library Specification [Fo01b]. Communicative Acts are used to exchange the desired content among the parties. For example, an auctioneer may want to declare the initial price of a lot, auction participants may need to communicate their bids at negotiation rounds, etc.
- *Reward structures*: Reward Structures are the means which allow to share and re-allocate the Utility among the parties. For example, in a group-buy combinatorial auction [YS01] the utility of buying a certain volume of a lot is evenly shared among the members of the group of buyers.

Definition 9: *Negotiation Protocol*. A Negotiation Protocol is the sub-structure of a Negotiation Mechanism. It aggregates the components which govern the interactions between Negotiation Parties: Interaction States, State transition Events, Valid Actions and Communicative Acts.

Negotiation Processes using different Negotiation Protocols may be *one-shot* and *iterative*. A Negotiation Process is said to be one-shot if its outcomes are elaborated in one round. This round normally comprises a proposal by one party or a group of parties and the acceptance or rejection feedback by the counterparty(-ies). A Negotiation Process is said to be iterative if the outcomes are reached iteratively in a series of rounds. It is assumed that the parties in the iterative Negotiation Process concede towards the agreement on the Negotiation Issues comprising the Negotiation Set by turns in these iterations. These concessions are done in a rational way through reasoning undertaken in Valid Actions. This reasoning is guided by the Negotiation Strategy of the party. Negotiation Strategies are shaped out by the Negotiation Role a party plays in the process. Different parties may employ similar or different Negotiation Strategies.

Definition 10: *Negotiation Strategy*. A Negotiation Strategy is the specification of the internal behavior of a Negotiation Party playing a Negotiation Role in a Negotiation Process. A party employs the strategy to reason about its next move in the Negotiation Process: (a) computing how much to concede on the values of the Negotiation Issues; (b) checking out if the Negotiation Goals are reached; (c) finding out if it is rational to

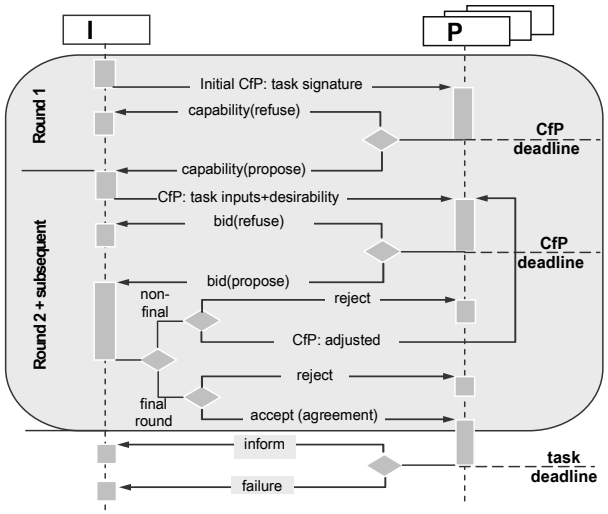


Fig. 2. Extended FIPA Iterated Contract Net protocol for outsourcing a task ([Er04a]).

continue or to terminate the Negotiation Process.

A reader may notice in Fig. 1 that GNO uses the Actor ontology and the ontologies of SWRL and ACL. The Actor ontology is the ontology of the PSI family of ontologies v.1.4 [Er06]. The simple ACL ontology was created by the authors for testing purposes and contains the minimal necessary set of the ACL performative patterns as strings. SWRL has been imported from the W3C source[11]. GNO[12] has been coded in OWL DL with the help of Protégé ontology editor[13]. Since that it may be used as the generic namespace for modeling different types of negotiation. We have started the evaluation of GNO by using it to model contracting negotiations on outsourcing a task in a design process. This type of negotiation has been chosen because it is the basic one we are implementing in the research prototype of PSI Dynamic Engineering Design Process agent-based planner and simulator software [Go05].

3 Modeling Contracting Negotiation Dialogue in PSI

One of the basic types of negotiation used in PSI is negotiation on out-sourcing a task. Such negotiation takes place each time: (a) an actor realizes, according to its knowledge of the task or because of the overload, that the task should be out-sourced to one of the fellow colleges; and (b) the actor believes that several appropriate candidates capable to perform this task are available. Negotiation is performed instead of a directive task assignment if an actor wants to make an optimal choice from the set of the possible contractors.

[11] http://www.w3.org/Submission/SWRL/swrl.owl
[12] http://kit.zsu.zp.ua/eva/psi-public/gno_v10_02-2006.owl
[13] Protégé ontology editor and knowledge acquisition system http://protege.stanford.edu/

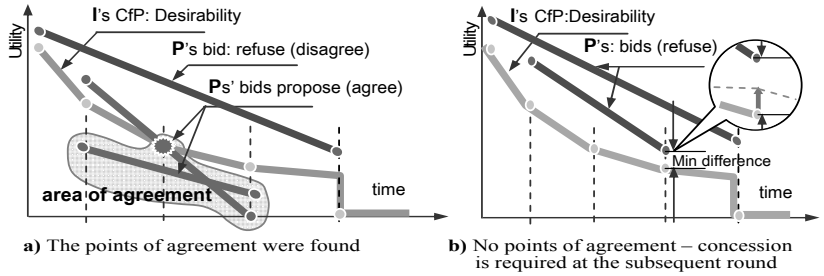


Fig. 3. Negotiation on outsourcing a task: agreement, disagreement, and concession ([Er04a]).

3.1 Contracting Negotiation Framework in PSI

In PSI the extension of the **FIPA Iterated Contract Net protocol (CNP)** [Fo01a] as the interaction protocol for this kind of negotiation (see Fig. 2) is used [Er04a]. The actor willing to out-source the task is considered to be the Initiator (**I**) in this encounter. The actors about which **I** believes that they are capable [Er04b] to perform the activity play the role of the invited Participant (**P**).

The first round of the process aims to find out if any of the known capable **P**s may agree to consider performing the task. Negotiation set for this round contains task signature only (for example, 'Code_GNO_in_OWL'). Negotiation at the second and the subsequent rounds is about the terms of the possible contract. **I** advertises the task inputs and the discrete desirability function. The desirability function specifies the dependency of the utility **I** is ready to pay over the possible time of the task accomplishment by the anticipated contractor. **I** than chooses the best proposal in case several proposals of the **P**s would result in the agreement (Fig. 3.a). The subsequent rounds are performed to adjust the task inputs or the desirability function in the case if no one of the **P**s has agreed on the previous round.

Ps refusals and proposals are shown in Fig. 3a. These feedbacks are formulated in a constructive way to allow **I** to adjust its CfP in the subsequent round. A feedback contains two (utility, time) bids defining the segment on which a possible agreement may be stricken. The area of agreement for the current round could be formally defined as the union of all those parts of the bid segments which are on and below **I**'s desirability function poly-line. All other points of the **P**s feedbacks indicate their disagreement with **I**'s offer at the current negotiation round.

In case at round *n* no agreements were detected by **I** it concedes just enough for not to concede at the next round *n*+1. The concession in utility value is computed as the half of the minimal difference in utility between the current desirability function and the current feedbacks of the **P**s. **I** may continue to concede in a series of rounds if:

– 1st, the **P**s concede accordingly in a monotonic way
– 2nd, the concession still makes the possible deal individual rational[14] for **I**

[14] A deal is individual rational if, having the deal, **I** is still better of or has the same utility as it was alone.

```
<NegotiationParty rdf:ID="Task_Manager">
   <rdfs:comment
   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
   This NegotiationParty is willing to out-source a Task
   to the one of the three Designers optimally chosen
   in the NegotiationProcess
   </rdfs:comment>
   <actor:Actor-name
   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
   Vadim Ermolayev
   </actor:Actor-name>
   <actor:Actor-UoWAssets
   rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
   200.0
   </actor:Actor-UoWAssets>
   <actor:Actor-description
   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
   Assoc. Prof., Dept of IT, ZNU
   </actor:Actor-description>
   <NegotiationParty-agentID
   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
   TM
   </NegotiationParty-agentID>
   <NegotiationParty-name
   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
   Task Manager
   </NegotiationParty-name>
</NegotiationParty>
```

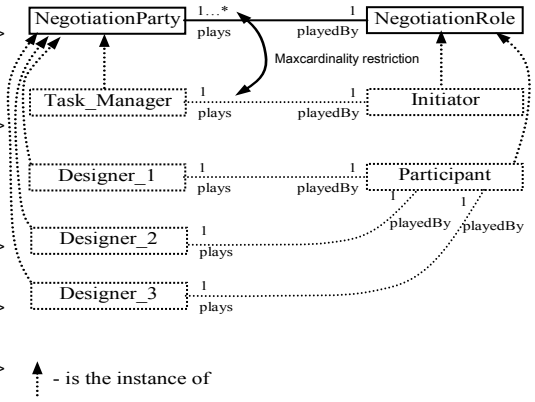


Fig. 4. Refining GNO with the instances and the property constraints.

I considers the negotiation round as final if it can accept one agreement of the **Ps** and strike the contract deal. The chosen **P** becomes the Contractor. **I** may declare the negotiation round as final by repeating the desirability function without concession. Hence, if the **P**s do not concede enough to make agreement in the last round, negotiation ends without reaching the agreement.

3.2 Using GNO to Modeling Contracting Negotiation

We shall model contracting negotiation process using GNO and following the methodology of [Ta05] by: (a) providing the instances of GNO concepts, (b) applying restrictions to some properties of GNO concepts and (c) adding new properties. We extend the approach of [Ta05] by allowing to: (d) create the subclasses of the GNO concepts. We shall consider that the modeled contracting negotiation process involves one Task Manager and three Designers as the parties.

Negotiation Roles. According to the description in section 3.1 we have the following two roles in contracting negotiation: the **Initiator**[15] and the **Participant**. These two instances of a NegotiationRole concept have been added to the ontology. These roles are played by the instances of a NegotiationParty concept.

Negotiation Parties. The Negotiation parties instantiated using GNO as the namespace were as follows: **Task_Manager**, **Designer_1**, **Designer_2** and **Designer_3**. OWL description of the **Task_manager** instance is given in Fig. 4. While creating the instance of the **Task_Manager** we had to add the restriction to the cardinality of its relationship to NegotiationRole instances as shown in Fig. 4. Indeed, according to the rules of the encounter (section 3.1) there may be only one **Task_Manager** playing the role of the **Initiator**.

[15] The instances and their property values are given in **bold**.

Negotiation Issues and Negotiation Goal. For the specific Negotiation Type we create the specializations (the subclasses) of the Negotiation Issues. For the specific Negotiation Process we create the instances of these subclasses and of the Negotiation Goal. Let the particular process be the negotiation on the out-sourcing of the task having the signature: Code_GNO_in_OWL. The inputs for this task are:

– The maximal price in the UoW[16] believed by the Task_Manager: 50
– The deadline believed by the Task_Manager: 05.02.2006
– The discrete desirability function for the task believed by the Task_Manager:

$$des_i(u_i, d_i, deadline) = \begin{cases} d_i \le deadline : (u_i, d_i) \\ d_i > deadline : (0, d_i) \end{cases} \quad (1)$$

where:

– u – the price to pay in UoW: (u_1=50, u_2=40, u_3=30, u_4=20)
– d – the date of task accomplishment: (d_1=30.01.2006, d_2=02.02.2006, d_3=04.02.2006, d_4=05.02.2006, d_5=06.02.2006, d_6=06.02.2006)
– *deadline* = 05.02.2006

The bids of the participants (please refer to Fig. 3) are specified as:

$$bid_j(u_1^j, d_1, u_2^j, d_2) = ((u_1^j, d_1), (u_2^j, d_2)) \quad (2)$$

So, the Negotiation Set is: {u, d, *deadline*, *des*, bid_1, bid_2, bid_3}. Negotiation Issues u, d, *deadline* are independent while *des*, bid_j depend on u, d, *deadline*. These dependencies are coded with SWRL rules. For example, we have coded the part of the desirability dependency as follows:

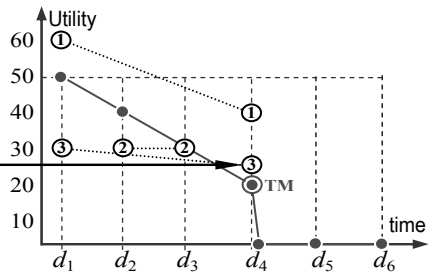
```
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#zeroDesirabilityRule"/>
  <owlx:Annotation>
      <owlx:Documentation>Ready to pay nothing (u=0)
    if an accomplishment date
      is greater than a deadline</owlx:Documentation>
 </owlx:Annotation>
 <ruleml:_body>
            <swrlx:datarangeAtom>
            <owlx:Datatype owlx:name="&xsd;double" />
                        <ruleml:var>u</ruleml:var>
            </swrlx:datarangeAtom>
            <swrlx:datarangeAtom>
            <owlx:Datatype owlx:name="&xsd;date" />

            <ruleml:var>accomplishDate</ruleml:var>
            </swrlx:datarangeAtom>
            <swrlx:datarangeAtom>
            <owlx:Datatype owlx:name="&xsd;date" />
                        <ruleml:var>dealine</ruleml:var>
            </swrlx:datarangeAtom>
            <swrlx:builtinAtom
            swrlx:builtin="&swrlb;#greaterThan">
                        <ruleml:var>d</ruleml:var>
                        <ruleml:var>deadline</ruleml:var>
            </swrlx:builtinAtom>
  </ruleml:_body>
  <ruleml:_head>
            <swrlx:datavaluedPropertyAtom
swrlx:property="desirability">
                        <ruleml:var>u</ruleml:var>
                        <owlx:DataValue
owlx:datatype="&xsd;double">0</owlx:DataValue>
            </swrlx:datavaluedPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

[16] The units to measure Utility in PSI are called the Units of Welfare (UoW) [Er04a, Er05b].

```
<NegotiationGoal rdf:ID="u-2-3">
   <trueValuation
   rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
   25.0
   </trueValuation>
   <public
   rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">
   false
   </trueValuation>
</NegotiationGoal>
<NegotiationGoal rdf:ID="d-4">
   <trueValuation
   rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
   05.02.2006
   </trueValuation>
   <public
   rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">
   false
   </trueValuation>
</NegotiationGoal>
```

a) The goal of **Designer_3**



b) The goals and the less preferred outcomes of the parties with respect to the desirability function

Fig. 5. Negotiation Goals.

The Negotiation Goals and the less preferred but acceptable outcomes in the terms of the true valuations of the (some of the) subclasses of Negotiation Issues are as follows:

- **Task_Manager**: The **Task_Manager** prefers to have the task accomplished on d_4 and to pay the minimum of u_4 thus maximizing his own utility in this encounter
- **Designer_1**: The **Designer_1** prefers and is capable to accomplish the task on d_1 and to receive $u_1^1 = 60$ for that. He will also agree to receive $u_2^1 = 30$ for finishing the task on d_3. The goals of the **Designer_1** are therefore: $u_1^1 = 60$, d_1.
- **Designer_2**: The **Designer_2** prefers and is capable to accomplish the task on d_2 and to receive $u_1^2 = 30$ for that. He will also agree to receive $u_2^2 = 30$ for finishing the task on d_3. The goals of the **Designer_2** are therefore: $u_1^2 = 30$, d_2.
- **Designer_3**: The **Designer_3** prefers and is capable to accomplish the task on d_4 and to receive $u_2^3 = 25$ for that. He will also agree to receive $u_1^3 = 30$ for finishing the task on d_1. The goals of the **Designer_3** are therefore: $u_2^3 = 25$, d_4.

The goals of the **Designer_3** have been coded as shown in Fig. 5a. The position of the goals with respect to the desirability function of the **Task_Manager**s is shown in Fig. 5b.

Negotiation Mechanism and Negotiation Protocol. The Negotiation Mechanism is modeled similarly to Negotiation Roles and Negotiation Parties by instantiating the Interaction States (**Initiate**, **Perform_Action**, **Wait_Message**, **Terminate**), the State Transition Events (**Participant_ShowedUp**, **Message_Sent**, **Message_Received**, **Contractor_Assigned, Participant_Left**), the Valid Actions (**Prepare_Initial_CfP**, **Prepare_Extended_CfP**, **Analyse_Bids_Concede**, **Analyse_Capability**, **Prepare_Bid**, **Analyse_CfP_Concede**), the Communicative Acts (**CfP**, **Announce_Bid**, **Reject**, **Accept**). One may notice that, according to Fig. 2, different parties may be in different subsets of the Interaction States, may have different subsets of the Valid Actions. These constraints are defined as SWRL rules in the frame of the Permissible Participant concept similarly to what we did with Negotiation Issues and Negotiation Goals. The

instances of the Reward Structure have been created for all the parties. The rules for changing the property values in the instances of a Reward Structure are also coded in SWRL. These rules use the content of the **CfP** and the **Announce_Bid** instances of a Communicative Act which comprise the instances of the *u* (the subclass of a Negotiation Issue) in their content part [Fo01b].

4 Discussion and Related Work

The major intention of the presented work is to develop the proposal of the standardized upper-level conceptual framework of rational negotiation subsuming different negotiation types. Of course we anticipate that this framework will be used in automated negotiation among the autonomous intelligent members of open heterogeneous software systems. In frame of PSI we prototype such a system using agent-based approach [Go05]. Therefore GNO is primarily oriented to facilitate to the negotiation among software agents.

In understanding the nature and the conceptual foundation of negotiation among software agents we base our work on the results from DAI and eCommerce domains. We adopt the theoretical basics of [DS83], [LWJ00] and [Je01] and build the Negotiation Mechanism part of GNO on them. In structuring and specifying the descriptions of the integral parts of a Negotiation Mechanism we use the results of the colleges working in the field of the computational mechanism design (a good survey may be found in [Hy05], a profound analytical review of the field is [DPJ03]). In this part GNO is the extension of the state of the art because it is the first to incorporate mechanism-based principles of specifying negotiation encounters. Our approach in designing GNO is close to the one of Tamma et al [Ta05] who propose an approach to automated negotiation that fully exploits the potential of open environments: agents should not be forced to commit to a single negotiation protocol, but should be able to choose the negotiation protocol which is most suitable to the type of interaction they participate in. The authors of [Ta05] then rightfully state that "this, in turns, implies also that agents should be able to tune their strategy to the specific protocol employed, in order to maximize their chances of success in the negotiation." However, the ontology proposed in [Ta05] adds nothing to this statement. GNO in this sense is a considerable step forward because it specifies Negotiation Strategies used as reasoning plug-ins in Valid Actions of a Negotiation Mechanism. Another advancement of GNO with respect to [Ta05] is the specification of Negotiation Goals in terms of the true valuations of the Negotiation Issues and the treatment of the coherence in goals in terms of Negotiation Types. GNO makes one more substantial step ahead in specifying that Negotiation Outcomes affect the Beliefs of an Actor which acts as a Negotiation Party. This allows making Negotiation Strategies more flexible and adaptive, for example using the technique proposed in [Er04b]. An example of using the similar ideas in modeling bargaining processes is [Sc05.]

5 Concluding Remarks

The paper presents our results in developing the generic conceptual framework for

modeling rational negotiation processes. The central piece of the presented work is the upper-level negotiation ontology. This ontology is designed to be capable to model arbitrary types of negotiation dialogues by providing the generic namespace for that. The models of the specific negotiation types refine this namespace by providing the instances of its concepts, applying additional restrictions on the properties, possibly adding new subclasses and properties. The approach to model negotiations with the Generic Negotiation Ontology (GNO) is initially evaluated by applying to contracting negotiation type. The choice of the negotiation type for the evaluation is suggested by the fact that contracting negotiations are typical encounters among agents taking part in the simulation of a DEDP. GNO is the formal ontology coded in OWL DL and may be used by software agents in their automated negotiation dialogues. Some simple negotiation dialogues are implemented in PSI DEDP-MAS software research prototype [Go05].

For the future work we plan the more thorough evaluation of the modeling framework based on GNO as well as the further development of the ontology itself. For better evaluation we'd like (a) to undertake the experiments in modeling different Negotiation Types, like meaning negotiation [Er05a] and (b) to use the presented framework in implementing more sophisticated negotiation dialogues in DEDP-MAS than we have today. In further development of GNO we intend to focus on the observation of [WK95] noticing that the dialogue types are rarely used in isolation – on the contrary, a dialogue may contain phases of different types. Another topic for further extension of GNO is the development of the refined representations for negotiation strategies.

Acknowledgements

We would like to acknowledge the substantial aid by all the members of the PSI project team in discussing and formulating the negotiation framework and shaping out the generic negotiation ontology in their current state. We would explicitly like to thank Vladimir Gorodetsky, Eyck Jentzsch, Oleg Karsayev, Wolf-Ekkehard Matzke and Vladimir Samoylov for their help in this work.

Bibliography

[DS83] Davis, R.; Smith, R. G.: Negotiation as a Metaphor for Distributed Problem Solving. Artificial Intelligence 20 (1983); 63–109

[DPJ03] Dash, R. K.; Parkes, D. C.; Jennings, N. R.: Computational Mechanism Design: A Call to Arms. IEEE Intelligent Systems 18 (6), 2003; pp. 40–47

[EKT02] Ermolayev, V.; Keberle, N.; Tolok, V.: OIL Ontologies for Collaborative Task Performance in Coalitions of Self-Interested Actors. In (Arisawa, H.; Kambayashi, Y.; Kumar, V.; Mayr, H.C.; Hunt, I. Eds.): Conceptual Modeling for New Information Systems Technologies. ER 2001 Workshops, Yokohama, Japan, 2001. Revised Papers - LNCS vol.2465, 2002; pp. 390–402

[Er04a] Ermolayev, V. et al: Agent-Based Dynamic Engineering Design Process Modeling Framework. Technical Report. Cadence Design Systems, GmbH, 2004; 29 p. http://ermolayev.com/eva_personal/PS/PSI-DEDP-MF-v10-Feb-2004.pdf

[Er04b] Ermolayev, V.; Keberle, N.; Kononenko, O.; Plaksin, S.; Terziyan, V.: Towards a framework for agent-enabled semantic web service composition. Int. J. of Web Services Research, 1(3), 2004; pp. 63–87

[Er05a] Ermolayev, V.; Keberle, N.; Matzke, W.-E.; Vladimirov, V.: A Strategy for Automated Meaning Negotiation in Distributed Information Retrieval. In (Gil, Y. et al. Eds.): Proc. 4th Int. Semantic Web Conference (ISWC'05), Galway, Ireland, LNCS vol.3729, 2005; pp. 201–215

[Er05b] Ermolayev, V.; Jentzsch, E.; Karsayev, O.; Keberle, N.; Matzke, W.-E.; Samoylov, V.: Modeling Dynamic Engineering Design Processes in PSI. In (Akoka, J. et al. Eds.): Perspectives in Conceptual Modeling, ER 2005 Workshops AOIS, BP-UML, CoMoGIS, eCOMO, and QoIS, Klagenfurt, Austria, LNCS vol. 3770, 2005; pp. 119–130

[Er06] Ermolayev, V.; Jentzsch, E.; Keberle, N.; Samoylov, V.; Sohnius, R.: The Family of PSI Ontologies. Reference Specification. PSI-ONTO-TR-2-2006. Cadence Design Systems, GmbH, 2006; 47 p.

[Fo01a] Foundation for Intelligent Physical Agents. FIPA Contract Net Interaction Protocol Specification. Ref. No XC00029E. 2001. http://www.fipa.org/specs/fipa00029/.

[Fo01b] Foundation for Intelligent Physical Agents.FIPA Communicative Act Library Specification. Ref. No SC00037J. 2001. http://www.fipa.org/specs/fipa00037/

[Go05] Gorodetsky, V.; Ermolayev, V.; Matzke, W.-E.; Jentzsch, E.; Karsayev, O.; Keberle, N.; Samoylov, V.: Agent-Based Framework for Simulation and Support of Dynamic Engineering Design Processes in PSI. In (Pechouchek, M.; Petta, P.; Varga, L. Z. Eds.): Proc. 4th Int. Central and Eastern European Conf. on Multi-Agent Systems (CEEMAS'05), Budapest, Hungary, LNAI vol.3690, 2005; pp. 511-520

[Gr93] Gruber, T. R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Int. J. Human-Computer Studies, 43(5/6), 1995.

[Hy05] Hyafil, N: Computational Mechanism Design. Depth Examination Report. Dept of Computer Science, Univ of Toronto, March 9, 2005, http://www.cs.toronto.edu/~nhyafil/Papers/Hyafil-depth.pdf

[Je01] Jennings, N.R.; Faratin, P.; Lomuscio, A.; Parsons, S.; Sierra, C.; Wooldridge, M.: Automated negotiation: prospects, methods and challenges. Int. J. of Group Decision and Negotiation, 10 (2), 2001; pp.199–215

[LWJ00] Lomuscio, R.; Wooldridge, M.; Jennings, N. R.: A Classification Scheme for Negotiation in Electronic Commerce. In (Dignum, F.; Sierra, C. Eds.): Agent-Mediated Electronic Commerce: A European Perspective. Springer Verlag, Berlin Heidelberg, New York, 2000; pp. 19–33

[Sc05] Schewe, K.-D.: Bargaining in E-Business Systems. In (Akoka, J. et al. Eds.): ER Workshops 2005, Klagenfurt, Austria, LNCS vol. 3770, 2005; pp. 333 – 342

[Ta05] Tamma, V.; Phelps, S.; Dickinson, I.; Wooldridge, M.: Ontologies for supporting negotiation in e-commerce. Int. J. of Engineering Applications of Artificial Intelligence 18, 2005; pp. 223–236

[WK95] Walton, D. N.; Krabbe, E. C. W.: Commitment in Dialogue. State Univ. of New York Press, 1995

[YS01] Yamamoto, J.; Sycara, K.: A stable and efficient buyer coalition formation scheme for e-marketplaces. In (André, E.; Sen, S.; Frasson, C.; Müller, I.P. Eds.): Proc. 5th Int Conf. on Autonomous Agents, Montreal, Quebec, Canada, 2001; pp. 576 – 583