[3] A. Abdollahzadeh Barfouroush, B. Masoumi, M. R. Ayatollahzadeh Shirazi, "An Introduction to Distributed Artificial Intelligence," Tehran: Jelveh Publishing, January 2006.

[4] I. Rahwan, S. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons and L. Sonenberg, "Argumentation-Based Negotiation," *Knowledge Engineering Journal*, December, 2003.

[5] A. R. Lomuscio, M. Wooldridge, and N. R. Jennings, "A classification scheme for negotiation in electronic commerce", *International Journal of Group Decision and Negotiation.* 12(1):31-56, January 2003.

[6] C. Bartolini, C. Preist, N. R. Jennings, "A Generic Software Framework for Automated Negotiation," *in Proc. of AAMAS '02*, July 15-19, Bologna, Italy, 2002.

[7] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons, "A Framework for argumentation-based negotiation," in M. Singh, A. Rao and M. Wooldridge, editors, Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages (ATAL-1997), volume 1356 of Lecture Notes in Artificial Intelligence, pages 177-192, Springer Verlag, 1998.

[8] J. B. Kim, A. Segev, "A Framework for Dynamic eBusiness Negotiation Processes," in *Proc. of the IEEE International Conference on E-Commerce Technology (CEC'03)*, p. 84, 2003.

[9] X. Zhang, V. Lesser, "Solving negotiation chains in semi-cooperative multi-agent systems," 2006, Submitted to AAMAS06, Available: http://mas.cs.umass.edu/paper/414..

[10] S. Parsons, C. Sierra, and N. R. Jennings, Agents that reason and negotiate by arguing, Journal of Logic and Computation, 8(3), 261-292, 1998.

[11] Xu, H., S. M. Shatz, S.M, An agent-based Petri net model with application to seller/buyer design in electronic commerce, Proceedings of 5th International Symposium on Autonomous Decentralized Systems, 2001.

[12] L. Amgoud, S. Parsons, and N. Maudet, Modelling dialogues using argumentation, In the proceedings of the 4th International Conference on Multi-Agent Systems ICMAS'2000, Boston 2000.

[13] M. Dastani, J. Hulstijin, van der Torre L., Negotiation Protocols and Dialogue Games, Proceedings of the fifth International Conference on Autonomous Agents, Montreal, Canada, 2001.

[14] P. Yolum, M. P. Singh, Flexible Protocol Specification and Execution: Applying Event Calculus Planning using Commitments, Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS). July 2002.

[15] S. Kraus, K. Sycara, and A. Evenchik, Reaching agreements through argumentation: A logical model and implementation, Artificial Intelligence, 104(1–2):1–69, 1998.

[16] F. Sadri, F. Toni, and P. Torroni, Abductive logic programming architecture for negotiating agents, Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA-2002), volume 2424 of Lecture Notes in Computer Science, pages 419–431. Springer Verlag, Berlin, Germany, 2002.

[17] P. Yolum and P. M. Singh, Commitment Machines, Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL-01). Springer, 2002.

[18] M. R. Ayatollahzadeh Shirazi, A. Abdollahzadeh Barfouroush, "Flexible Approaches to Automated Negotiation," Technical Report, Computer Engineering and IT Department, Amirkabir University of Technology, December 2005.

R4: **if** $V^t_{a\leftrightarrow b}$ = offer (Customer, Service Provider, value(m,x)) and IsNear(GainedBenefit(x), Expectedbenefit) **then** accept(Service Provider, Customer,value(m,x)) **where** m $\in O$, x $\in N$, //Acceptance of offer

R5: **if** $V^t_{a\leftrightarrow b}$ = offer (Customer, Service Provider, value(m,x)) and $x^t=x^{t-1}$ **then** offer(Service Provider,Customer,newobject(o),value(o,z))

R6: **if** $V^t_{a\leftrightarrow b}$ = reject (Customer,Service Provider, value(m,x)) and x-y>s **then** offer(Service Provider, Customer,value(m,x-y)) **where** $m\in O$, $x\in N$, y=n<x

R7: **if** NegotiationTime > ExpectedNegotiationTime **then** withdraw(Service Provider,Customer,negotiation(m))

Customer strategy

R1: **if** $V^t_{a\leftrightarrow b}$ = reject (Service Provider, Customer, value(Service,Price)) **then** **propose**(Customer, Service Provider,value(Service, Price+y)) **where** 0<y<n, Price+y< b

R2: **if** $V^t_{a\leftrightarrow b}$ = propose (Service Provider, Customer ,value(Service, Price)) and VeryHigher(Price,reservedprice(Customer)) **then** **reject**(Customer, Service Provider,value(Service, Price)

if VeryLower(b,MarketValuation(Service)) and Need(Customer, Service) **then** Equal(b,Near(MarketValuation(Service)))

R3: **if** VeryLower(b,MarketValuation(Service)) and CannotAfford(Customer,MarketValuation(Service)) **then** Withdraw(Negotiation(Service))

R4: **if** $V^t_{a\leftrightarrow b}$ = propose (Service Provider, Customer ,value(Service,x)) and x>MarketValuation(Service) **then** **reject**(Customer, Service Provider,value(Service, Price)

R5: **if** $V^t_{a\leftrightarrow b}$ = propose (Service Provider, Customer ,value(Service,x))and IsNear(x,MarketValuation(Service)) and IsVeryLower(x,b) **then** **accept**(a,b,value(Service,x))

R6- **if** $V^t_{a\leftrightarrow b}$ = propose (Service Provider, Customer ,value(Service,x)) and IsLower(x,b) **then** **accept**(a,b,value(Service,x))

R7- **if** $V^t_{a\leftrightarrow b}$ = propose (Service Provider, Customer ,value(Service,x)) and x<b and $x>x^{t-1}$ **then** propose(a,b,value(Service,x+y)) **where** 0<y<n, x+y< b and x^{t-1} is the proposed value for Service in previous round.

In the above strategy, if a Service Provider agent receives same price offers in two consecutive offers, it will try to introduce a new object in negotiation process to make a progress in process. Of course, this strategy is one of the possible ways to break the locks in the negotiation process. Another possibility of changes in negotiation object set is a request from other party (Service Provider or Customer agent) to add or remove a negotiation object.

Also, during negotiation, the Service Provider agent may gain knowledge about reserved price of Customer agent. In this case, Service Provider agent changes its strategy and tries to offer prices close to this reserved price and not to offer any price that is much lower than this bound. If the Service Provider agent finds that the highest price that Customer can pay for the service is lower than the minimum price that Service Provider can sell the service then the Service Provider agent will withdraw the negotiation on this service.

4. Conclusion

Negotiator agents need to negotiate with each other to manage their task interdependencies, share the resources and obtain the needed resources for performing their tasks or reaching their goals. Our experiences in developing multiagent systems show that negotiation exposes itself as a system. Negotiation protocol, strategy and architectural aspects of negotiator agents are the main elements of this system. Design of these elements is under influence of many parameters of the application domain and negotiation scenario. For example, every application domain has characteristics such as style of interactions and nature of negotiation objects in that domain that have impact on the design of a negotiation protocol and every negotiation scenario is one of the possible negotiation spaces that are generated by a number of negotiation parameters and these parameters are important factor that influence the design of negotiation protocol and strategy. Therefore, it is very important to identify the relationships among parameters of the application domain and negotiation scenario and the elements of the negotiation system. Thus, two things are important: a process for constructing the negotiaiton system and a framework that represent the main concepts and entities in development process and their relationships with each other.

In this paper, we presented the process and a conceptual framework for modeling automated negotiation systems. The framework has been used in our work for evaluating various negotiation scenarios for a flexible negotiation system as a template for capturing and representing the requirements of a practical negotiation system. It serves as a medium for communication among analysts and developer(s) of a computational negotiation system. It is also used for representing human negotiations scenarios that we want to analyze and simulate with a multiagent system.

5. References

[1] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra and M. Wooldridge, "Automated Negotiation: Prospects, Methods and Challenges," *International Journal of Group Decision and Negotiation*, 2000.

[2] P. Faratin, "Automated Service Negotiation Between Autonomous Computational Agents," Ph.D. Dissertation, University of London, Queen Mary College, Department of Electronic Engineering, 2000.

agent A in a specific time *t* and *p* is a propositional formula.

- Commitment: Each agent commits that if it offers a proposal it will not start a negotiation about the same object until it receives accept or reject response.

- *Env* is a type of environment that negotiation takes place. In this scenario *Env* = {static}.
- *Ochar* is a characteristic of the objects over which the agents negotiate. The object is used for private use and is discrete.
- *E* defines some regulations such as validity and visibility of proposals, timeouts and schedules in negotiation. In this scenario, negotiation should reach to a conclusion in a specified time period.

B. Representation of a Negotiation Protocol

According to the definition of negotiation protocols in this scenario the elements of negotiation protocol are defined as follow:

Negotiation action set is defined as A= {propose, accept, reject, withdraw}.

Negotiation object set is defined as *O*={DSL service}.

Final state of negotiation is defined as *S*={Closed, Withdraw}.

Agent roles can be defined as *R*= {Service Provider, Customer}.

Behavior and state mapping of the negotiation protocol is defined by using the state machine in Figure 3.

C. Representation of a Negotiation Strategy

In all negotiation scenarios, negotiation strategy plays an important role. Negotiation strategy is the decision making mechanism that determines the next action that should be done in current situation of negotiation process. In the defined scenario there two parameters that affect design of a negotiation strategy:

Negotiation object: In some situations such as lack of progress in negotiation, agents may decide to introduce a new object in negotiation object set or remove an object from negotiation. For example, in a negotiation for buying a car, Service Provider agent may offer a one-year insurance for the car in order to satisfy the Customer agent to buy the car. In some cases, the objects that are added or removed are not under direct negotiation and their role is to speed up reaching to an agreement or conclusion in negotiation.

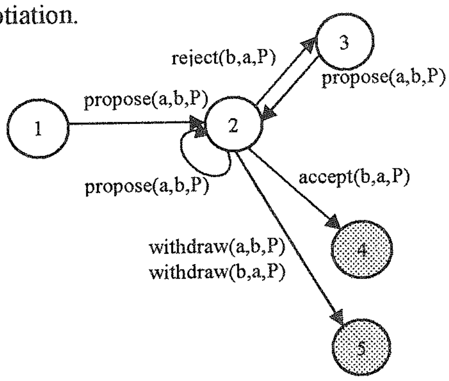

Figure 3: Representation of negotiation protocol with a Finite State Machine (FSM).

Knowledge of agents: During negotiation knowledge of participant agents about each other and about the negotiation object may change. For example, in negotiation for buying a product Customer agent, Service Provider agent or both of them may gain knowledge about the reservation price of each other. This new knowledge changes the way that agents negotiate with each other. The main change is in negotiation strategy of agents: one or both of negotiator agents may terminate the negotiation, one of agents may change the quality or quantity of negotiation object(s) or negotiator agents may change their strategy in negotiation.

As can be seen, negotiation strategy of each agent should be designed in such a way that agent can handle these changes in negotiation parameters. In the following lines, we define a negotiation strategy for agents that can handle changes in negotiation objects and knowledge of agents during negotiation. In this scenario, the Service Provider agent will be informed about the highest price that the Customer agent can pay for a service. In the following lines, we represent part of negotiation strategies of Service Provider and Customer agents. In representation of this strategy we have used predicates of predicate logic. We assume that agents have a belief base in which they can store factual information about the world and we have included some fuzziness in decision making rules of agents to better encode the reasoning rationale of human decision makers in the negotiator agents.

Service Provider strategy

R1: **if** $V^{t}_{a\leftrightarrow b}$ = offer(Customer,Service Provider, value(m,x)) and Verylower(x,s) **then** reject(Service Provider, Customer, value(m,x) **where** $m \in O$,$x \in N$ //The received offer is very lower than the minimum price that Service Provider can sell its service. Thus, Service Provider rejects the offer.

R2: **if** $V^{t}_{a\leftrightarrow b}$ = offer(Customer,Service Provider, value(m,x)) and IsNear(x,s) and Believe(Service Provider, reservedprice(Customer,b)) **then** offer(Service Provider, Customer,value(m,Near(x,b))) **where** $m \in O$, $x \in N$ //Service Provider knows the reserved price of the Customer and tries to offer proposals that are near this reserved price

R3: **if** $V^{t}_{a\leftrightarrow b}$ = offer(Customer,Service Provider, value(m,x)) and x>s and Believe(Service Provider, reservedprice(Customer,b)) and **not** IsNear(x,b) **then** offer(Service Provider, Customer,value(m,Near(x,b))) **where** $m \in O$, $x \in N$ //Service Provider knows the reserved price of the Customer the offered price of Customer is not near this value and Service Provider tries to offer proposals that are near this reserved price.

INTERACTION CARDINALITY AND ITS EFFECT ON THE DESIGN OF NEGOTIATION PROTOCOL.

Agent Role	Numbe r	Negotiation Protocols	
Customer	one	Bargaining	Reverse auction, tendering, RFQ
	many	One-to-one negotiation with each Customer, auctions	Double auctions, market-based negotiations, stock exchanges
		Auctions	
		one	many
		Service Provider	

In the next section, we show how the above mentioned framework can be used in developing a negotiation system for reaching agreement on providing a specific service.

3. An Example of the Framework Usage

As a sample, we consider a possible negotiation scenario in the domain of DSL Service Provisioning. The general characteristics of this application domain were mentioned in Table 1 in section 1. In this domain, Service Provider agent(s) wants to sell one or more DSL service to Customer agent(s) directly or through broker agents. Each agent has a valuation function to compute the value of the received proposal. Service Provider agent has a resistant point or reservation price. This parameter is the minimum value that Service Provider assigns to the service. Customer agent has a resistant point. This parameter is the maximum value that Customer assigns to the service. x* is final agreement on the service. In negotiation, if b<s, then there is no area of agreement. If b>s, then there is an area of agreement in which parties may reach to an agreement. In this example, we should define negotiation action set, negotiation object set, final state of negotiation and behavior of the protocol based on a representation such as a Finite State Machine (FSM). The syntax of actions in negotiation action set is represented as *locution* (σ,ρ,Π) in which:

– σ is sender of locution so that Agents $\sigma \in$ Agents.

– ρ is receiver of locution so that $\rho \in$ Agents.

- *locution* is one of negotiation actions so that *locution* $\in$ A

- If L is a propositional language then Π is any formula or any set of formula in L.

In the scenarios in which some information (justification, reasoning, support, etc.) is passed along with the locutions (argumentation-based negotiation), the syntax of actions will be extended to include a support for a statement that is uttered by the sender agent. This means that the syntax of negotiation action will be *locution*(σ,ρ,Π,Σ) in which:

– σ is sender of locution so that Agents $\sigma \in$ Agents.

– ρ is receiver of locution so that $\rho \in$ Agents.

- *locution* is one of negotiation actions so that *locution* $\in$ A.

- If L is a propositional language then Π is any formula or any set of formula in L and Σ is a support for Π. Agent can send explanations in locutions that support its utterances. These explanations can be any formula or any set of formula in L.

The negotiation strategy for a given scenario is defined based for each negotiation party. We use a rule-based representation for defining the rules that an agent use for making decision about the next action that should be done in the present situation. Strategy rules define the way an agent interpret an incoming message and generate an appropriate action. Each rule has the following representation:

if $V^{t}_{a\leftrightarrow b}$ = *locution* (σ,ρ,Π) **then** *locution* (ρ,σ,Π) = Evaluate (Π)

$V^{t}_{a\leftrightarrow b}$ is the message that is sent by agent a to agent b at time t. Evaluate(Π) is an interpretation function of agent that 'interprets the statement of the other agent and generated an appropriate message for sending to the receiver.

A. Defining a sample Negotiation Scenario

In this section, we define a negotiation scenario in the DSL service provisioning application domain based on the specified parameters. Here we consider a more complex case in which the parameters of a negotiation scenario change during negotiation process. The main causes of changes in these parameters are nature of environment, (e.g., environment openness), solving problems in negotiation process (e.g., removing deadlocks) and agent's negotiation strategy. We define this scenario based on the following parameters:

- *Ocard* is a changing parameter in this scenario. At the beginning, negotiation is done over one object. During negotiation, this parameter changes. This means: $Ocard^{t0} = 1$ and $Ocard^{t0} \neq Ocard^{t'}$
- *Icard* = one-to-one. This means that negotiation is done between one Service Provider and one Customer.
- *AChar* defines characteristics of agents. Characteristics of agents depend of:

- Agent role: This parameter specifies the role that each agent plays in negotiation. In this scenario *Achar.Role* = {Service Provider, Customer}.

- Agent rationality: In this scenario agents have bounded rationality, *Achar.Rationality* = Bounded.

- Agent knowledge: This is a changing parameter. At the beginning of negotiation process agents do not know anything about "reservation price" and utility function of each other. During the negotiation process one party gains an estimation of "reservation price" of other party. This means that:

$K^{t'}_{A} = K^{t}_{A} \cup p$, where K^{t}_{A} is a knowledge base of intervals.

Negotiation protocol specifies, at each stage of negotiation process, who is allowed to say what. Thus, a negotiation protocol is a formal set of conventions governing the interaction among participants [1,4].

DEFINITION 3. Negotiation protocol is defined by a tuple $P = <A,\pi,I,S,O,R>$, where:

- A is a set of valid actions that participants can perform in certain situations.
- $\pi : A \rightarrow 2^A$ is a protocol mapping function in which 2^A is all subsets of the set of valid actions, A.
- S is a set of negotiation states.
- I: $A*S \rightarrow S$ is a function that determines the next state.
- O is a set of negotiation objects.
- R is a set of negotiation rules.

Negotiation strategy plans the action sequences of agents during negotiation.

DEFINITION 4. A very simple and abstract definition of negotiation is a function S: $2^A \rightarrow A$ such that if $A \subseteq T$ then $T \in S(T)$.

Another point that should be mentioned is the role of formal framework that is chosen in development of a negotiation system. For representing the elements of a negotiation system, existing formal representation techniques can be used. For example, a variety of representation techniques have been used for representing negotiation protocols. The examples of such representation techniques are: Finite state machines [10], Petri Net [11], dialogue games [12,13], event calculus [14], logic programs [15], Abductive Logic Programs [16], and extended propositional logic [17]. Some of these representation techniques such as finite state machine or Petri Net are not flexible because they model protocols in terms of legal sequences of actions. In this way, agents do not compute transitions during negotiation process and follow the formalism. Thus, this kind of formalisms is dependent on the previous action and strictly enforces a certain behavior. We can say that flexibility of the agents in executing these protocols is limited and protocols are over constrained [18]. Approaches such as dialogue games provide clear and precise declarative semantics of the interactions by stating the pre- and post-conditions of each locution as well as its effects on agents' commitments. In this way, agents that follow the protocol, use the action semantics of protocol and reason about the next action.

Design of negotiation strategy depends on characteristics of the application domain and negotiation scenario among agents. For example, if the style of interaction in a specific application domain is based on utility maximizing and self-interest, then the developer should design of competitive negotiation strategy.

D. Relationships among the framework elements

The parameters of a negotiation scenario affect the design of negotiation protocol, negotiation strategy and negotiator agents. Cardinality of interactions determines the negotiation protocol that governs the interaction. By detail analysis of some key parameters such as cardinality, events, information and allocation we define the negotiation system elements. For example, consider a classic buying-selling negotiation. Table 1 shows the role of interaction cardinality on the chosen protocol for negotiation. This table shows possible negotiation protocols that can be used in different buying-selling negotiation scenarios. Cardinality of the negotiation objects can determine the design of a utility function that is used to relate different object in negotiation.

Depending on how knowledge of agents about the negotiation objects is distributed, agents may choose different a negotiation strategy. Thus, knowledge of agent is crucially important to design the agent's strategy. Dynamicity of the environment that negotiation takes place in that might affect the design of the agent's preferences in decision making about a proposal (negotiation strategy). In a dynamic environment, it is better for agents to adopt their preferences as they progress in the negotiation process.

Events parameters forms an important part of negotiation protocol specification. Event parameters mainly determine the rules for validity and visibility of proposals. Also they determine termination rules of any negotiation [6].

If the agents decide to have some pre-negotiation before starting the negotiation, this information parameter will have impact on the design of the negotiation protocol and negotiation strategy. For example, in reference [9] Zhang et al. introduce a pre-negotiation phase in semi-cooperative negotiation chains that allows agents to transfer meta-level information. Using this information, the agent can build a more accurate model of the negotiation in terms of modeling the relationship of flexibility and success probability. This more accurate model helps the agent in choosing a good negotiation solution. If some information such as description about the proposal, justification for the offer or reasoning behind the offer should be exchanged in addition to the proposals, then designer should consider protocols for argumentation-based negotiation [4] and techniques for interpretation and evaluation of arguments in the negotiation strategy [4]. In other words, we can say that the following relationships exist among the negotiation parameters and elements of negotiation system:

$P = \mathcal{F}(Icard, Ochar, E, I, A)$, where P is a negotiation protocol.

$S = \mathcal{F}(Ocard, Icard, Achar, Env, I)$, where S is a negotiation strategy.

TABLE 3

issues(s)	
Attributes of object	Bandwidth, quality of service (QoS), delivery time, cost, duration, security, start-time.
Possible roles of agents	DSL service provider, Customer, broker, value-added service provider.
Cardinality of negotiation objects	Negotiations can range over a number of objects and multiple attributes of the objects. For example, negotiating over many instances of a DSL service and attributes such as price, QoS, etc.
Style of interaction	Negotiations involve self-interested, utility maximizing agents. Agents may share the same system goal but have different individual preferences.
Agent organization	Among individual agents or group of agents. Thus interaction cardinality can be one-to-one, one-to-many or many-to-many.
Constraints	Time is a critical factor. Timings are important on two distinct levels: (a) the time it takes to reach an agreement must be reasonable; and (b) the time by which the negotiated service must be executed is important in most cases and crucial in others.

B. Negotiation Scenario

As shown in Figure 1. after specifying common characteristics of the application domain in which negotiation is done, we should specify and represent the parameters of the negotiation space in which agents currently interact with each other. As mentioned in [5], every negotiation scenario is defined based on a number of parameters which generate a space of possible negotiation scenarios. We define a negotiation scenario according these parameters.

DEFINITION 1. A negotiation scenario is defined by a tuple NGSC = <*Ocard, Icard, AChar, Env, Ochar, E, I, A*>, where:

- *Ocard* defines the cardinality of the negotiation object set.
- *Icard* defines the cardinality of the interaction that take place among agents.
- *AChar* defines characteristics of agents. Characteristics of agents depend of their key attitude, goal, motivation, role, rationality, knowledge, commitment, social behavior, trust and openness, predictability, aggressiveness and decision making strategy.
- *Env* is a type of environment that negotiation takes place. Environment can be static or dynamic.
- *Ochar* is a characteristic of the objects over which the agents negotiate.
- *E* defines some regulations such as validity and visibility of proposals, timeouts and schedules.
- *I* defines the information other than proposals that may pass among the negotiation participants.
- *A* is applied in many-to-one and many-to-many negotiations and governs the winner of a negotiation when more that one agent has shown interest in the object.

Many design aspects of the practical negotiation system that can work in this setting is determined by the parameters of the negotiation scenario. Developer can use this definition as a guideline for representing the current negotiation space. After performing this task, developer should specify the elements of the negotiation system.

C. Negotiation System

Negotiation system defines main elements of a computational system that should be implemented to achieve the objectives of negotiation among agents in the specified application domain. Negotiation protocol, negotiation strategy and agents that participate in negotiation are main components of this system.

During design of negotiation system, two other factors are important: Negotiation approach and evaluation criteria. There are three main negotiation approaches: Game-theoretic, Heuristics-based and argumentation-based approach. Each of these approaches has their own features and is used in particular circumstances due its underlying assumptions. The chosen approach has direct impact on the design of elements of a negotiation system.

Also, during the specification phase, designer should decide about the evaluation criteria of negotiation process and negotiation product. These criteria (e.g., Pareto efficiency, optimality, social welfare, negotiation success, etc.) will be used during execution of the deigned negotiation system for testing and evaluating the elements of negotiation system and output of negotiation.

DEFINITION 2. We represent a negotiation system *NS* with a tuple NS = <*Agents, Roles, R, P, S, FF, Time*> , where:

- *Agents* is a set of negotiating agents.
- *Roles* is a set roles that agents play in negotiation.
- *R: Agents * Agents → Roles* is a function that assigns a role to an agent.
- *P* is a negotiation protocol.
- *S* is a negotiation strategy.
- *FF* is a formal framework is used to model negotiation protocol and strategy. This framework can be based on defeasible logic propositional logic, predicate logic or multi-modal BDI logic.
- *Time* indicates a set of ordered discrete time

evaluation and interpretation mechanisms. These related works are domain and/or approach specific and the components of these models can be used to implement real world applications. A more related work is a framework that is presented in reference [8] for negotiation process in dynamic e-business environments. The framework provides the foundations for constructing dynamic negotiation processes in five components: 1) negotiation requirements, negotiation structure, negotiation process, negotiation protocol and negotiation strategy. The work reported in this paper, is the first work that analyzes the relationships that exists among general characteristics of application domain, negotiation scenario parameters and elements of a negotiation system and their effects on each other in a single conceptual framework. In this framework, these elements are defined and represented.

This paper has been organized as follows. In section 1, automated negotiation and main research problems in this field are discussed. Then, as a basis for forming the elements of the conceptual framework, a process for development of a negotiation system is given. Section 2 introduces the framework and its elements. Section 3 discusses the relationships among the framework elements. In section 4, for evaluating the usage of the framework, all elements of the proposed negotiation framework are defined based on a classic distributed bargaining scenario in the domain of DSL service provisioning. In section 5, our conclusions are presented.

2. A Conceptual Framework for Automated Negotiation

In a multiagent system, in which agents should be equipped with negotiation mechanisms, we need a framework and guideline that help us to specify all the main parameters that are important in our decisions during design of negotiation mechanisms. This framework should specify what aspects of negotiation should be considered and what information is needed for giving an overall design view of those aspects. Also, this framework should determine how these aspects relate to each other. For defining such a framework, we first considered the necessary steps for designing negotiation component of a multiagent system. This process was discussed in section 1. Then, we tried to detail the activities of each step to see what concepts and elements should be defined in each activity.

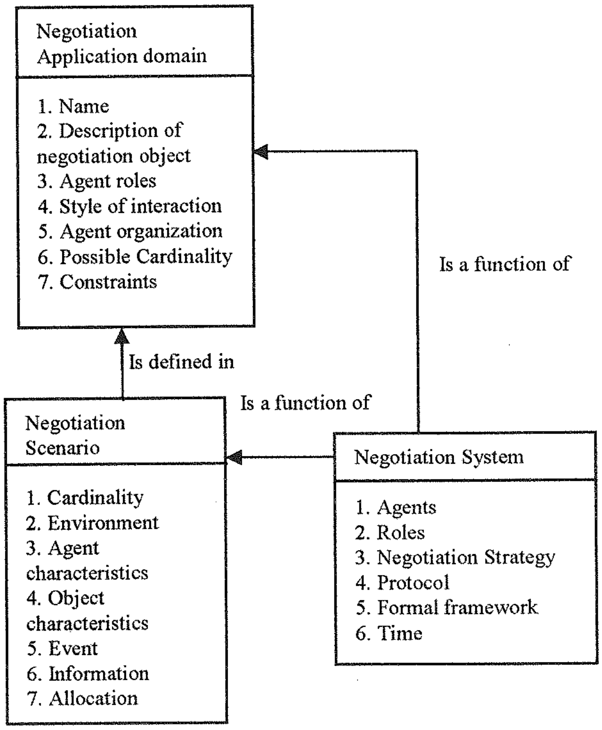

Figure 2: A Model of framework that represents its elements.

The result of this study is the conceptual framework that has shown in Figure 2. In the following subsections we define each element of this framework.

A. Application Domain

The application domain has some general characteristics that determine nature of many aspects of negotiation such as negotiation object(s), characteristics and attributes of negotiation objects and rules that govern the negotiation. For example, consider a multiagent system that works in the domain of telecommunication service provisioning in which two agents should negotiate with each other over a Digital Subscriber Line (DSL) service. Table 2 shows the general characteristics of this application domain that are likely common to a wide range of negotiation scenarios in this domain.

Table 2
Characteristics of the DSL Service Provisioning domain.

Characteristic	Description
Problem scenario	The provisioning scenario starts when a customer with a specific need contacts the service provider and asks for a DSL service. The scenario ends successfully when customer and service provider reach to an agreement on a service. More description about the scenario should be provided separately.
Negotiation object(s) or	DSL service

practical and computational negotiation system is to describe the application domain in which agents should negotiate with each other. Nowadays, multiagent systems are being used in an increasing wide variety of application domains such as industrial, commercial, medical, networking and educational domains. Each of these domains has specific characteristics. These general characteristics affect design of the negotiation system that is developed for the multiagent system that works in the application domain.

TABLE I
MAIN CHALLENGES IN AUTOMATED NEGOTIATION RESEARCH.

Research problems	Related subproblems/questions
Negotiation frameworks: - What are the frameworks that are needed to conduct automated negotiation? What are the elements of these frameworks? -What are the representation techniques for representing elements of negotiation frameworks?	a. Communication language and Domain language: Providing rich communication languages with clear semantics. Building common, standardized domain languages in order to use in heterogeneous environments.
	b. Negotiation protocol: - Specification technique - Design of negotiation rules. - Conformance checking of negotiation protocols - Evaluation of negotiation protocols
	c. Negotiation strategy
Approaches to automated negotiaiton and their computational assumptions	Game-theoretic Heuristic-based Argumentation
Architecture for negotiator agents	What are the effects of agents' capabilities on a chosen negotiation mechanisms?
Techniques and tools for analysis the behaviour of a negotiation system	What are the needed tools for verification and validation of the developed negotiation protocol, negotiation strategy and negotiator agents and overall behavior of the system according to the evaluation criteria such as Pareto efficiency, optimality, time, etc.
Evaluation criteria for negotiation systems	What are the criteria for measuring and testing negotiation systems in each negotiation approach?

After defining related characteristics that have implicit or explicit effect on the negotiation system, the developer should define specific parameters of a negotiation scenario. A negotiation scenario that takes place among agents is an instance of possible negotiation scenarios that can exist in an application domain. This negotiation scenario is defined by certain negotiation parameters [5]. Parameters of negotiation scenario are more specific and are directly related to the space that negotiation system should work in it. After representing the scenario, we define the appropriate negotiation system based on its elements. The elements are negotiation protocol, negotiation strategy and related architectural blocks of negotiator agents. After these activities, we should implement and test the negotiation system.

In this paper, we describe a conceptual framework for representing all the needed descriptions, concepts, entities, and information for development of a negotiation system. Main elements of this framework are negotiation application domain, negotiation scenario and negotiation system. This representation framework is used as a template for capturing and representing the requirements of a practical negotiation system. It serves as a medium for communication among analysts and developer(s) of a computational negotiation system.

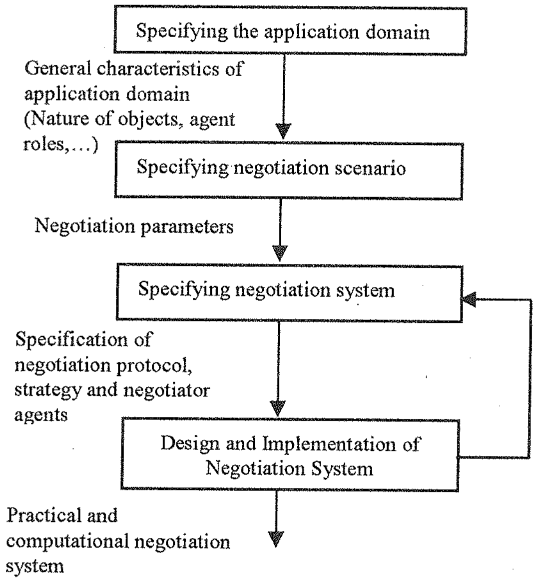

Figure 1: Main steps for developing a negotiation system.

Researchers in the field of automated negotiation have proposed similar frameworks. For example, Bartoloni et al. at HP Laboratories have created a general framework for automated negotiation dedicated to market mechanisms [6]. They present an abstract architecture for the negotiation framework in which they just focus on the shared protocol, not the negotiation strategy. Sierra et al. have devised a general framework for negotiation through argumentation [7]. In this framework, they use Dialogical Frameworks to define shared ontology, social relations, communication and protocol. Also, they define argument

A Framework for Representation of Negotiation Scenarios

Mohammad Reza Ayatollahzadeh Shirazi[i]; and Ahmad Abdollahzadeh Barfouroush[ii]

ABSTRACT

One of the main research problems in multiagent systems is designing effective and efficient negotiation mechanisms to help the agents to negotiate with each other in order to come to a mutually acceptable division/exchange of scare resources and manage their task interdependencies. For constructing these mechanisms, the developer should determine and specify all the necessary elements, concepts and parameters that are important for negotiation and then based on this specification build the negotiation constructs for performing negotiation in the multiagent system. In this paper, we present a conceptual framework for modeling and developing automated negotiation systems. This framework represents and specifies all the needed concepts and entities that are crucial for developing a negotiation system and existing relationships among these concepts. In addition to the role of this framework in development of a computational negotiation system, it is used in modeling human negotiations scenarios for analysis these types of negotiations and simulating them with multiagent systems. The work reported in this paper is the first unified framework that considers all the needed elements for modeling and developing negotiation systems and existing influence relationships among them.

KEYWORDS

Agent, automated negotiation, multiagent system, negotiation scenario

1. INTRODUCTION

In multiagent systems, a group of agents interact with each other to fulfill their objectives or improve their performance. One of the main problems in multiagent systems is how these agents interact with each other and what interaction mechanisms are needed [1,2,3]. Different types of interaction suit different types of environments and applications. These interaction mechanisms facilitate communication, coordination, cooperation and negotiation among agents. The most powerful mechanism for managing interdependencies among agents at run-time is automated negotiation [2]. Automated negotiation is a form of interaction in which a group of agents with different objectives and a desire to cooperate try to come to a mutually acceptable agreement [1,4]. This typically occurs when agents have competing claims on scare resources (anything that is needed to achieve something), not all of which can be simultaneously satisfied. There are a number of research challenges related to the development of suitable negotiation mechanisms [4]. We have summarized the main challenges in negotiation research in Table 1. As shown in this table, one of the problems in automated negotiation research is developing an appropriate negotiation system that can appropriately work in a given negotiation scenario. For solving this problem, we need a process to specify all necessary activities for constructing a negotiation system as a part of development of the multiagent system. We believe that for creating a computational negotiation system we should follow the steps depicted in Figure 1.

As shown in Figure 1, the first step in developing a

[i] M. R. Ayatollahzadeh Shirazi is a PhD student at the Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran (e-mail: ashirazi@ce.aut.ac.ir).

ii A. Abdollahzadeh Barofouroush is with the Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran (e-mail: ahmad@ce.aut.ac.ir).