

Intelligent Application Implementation Model for Automated Agent Negotiation

Wrya Anwar Hayder

IT Department
Kalar Technical Institute
Sulaimani Polytechnic University
Sulaimani, Iraq
wrya.hayder@spu.edu.iq

Mohammed Hassan Husain

IT Department
Kalar Technical Institute
Sulaimani Polytechnic University
Sulaimani, Iraq
mohamad.hassan@spu.edu.iq

Abstract -In the last few years multi-agents communication has seen an increased concern. The first main driver was intelligent agent is claimed to be the next generation model for complex and distributed system engineering. Second the agents can automate process and communication on behalf of the agent's users for proper actions and decisions. There are many concerns about how to implement communications among different agents in term of applied technology. In this paper we propose a REST- API and JSON technology as implementation tools that can be applied to implement intelligent web-based applications. REST is stands for Representational State Transfer which is allowing communication between web-based agents. JSON (JavaScript Object Notation) is a lightweight data format makes it easy for agents to parse and interchange data. This study has merged REST-AP and JSON technology into an automated negotiation model, and then a prototype is developed to successfully apply the proposed model. This model can facilitate application in business application negotiations and even internet of things (IOT) where HTTP protocol is used.

Keywords: multi-agent system, automated negotiation, web-based agent technology, e-commerce automated, intelligent application, negotiation web agents

1. INTRODUCTION

Intelligent agents are being promoted to be the future trend in distributed computation theory of complex systems. Agent study has seen a great concern from various perspectives and levels. Agent has been defined as a computer or machine embedded in some environment, that capable of autonomous actions in response to specific inputs perceived from external environment [1].

1.1. Intelligent Agent's Properties

A set of characteristics have been appointed, by which an agent should be determined. Firstly, agents should have well defined boundaries and clearly defined problem solving entities. Second, it must suit precise domain or embedded in a particular environment. Thirdly, designed to fulfill specific purpose objectives and achieve specific goals. Last but not least, It also should capable of specific behavior in pursuit of their need in order to respond to the environment requests [2][3]. In term of agent properties there are numbers of

important properties that have been appointed, that should be performed by the agents including:

- **Reactiveness:** the ability of agents to change their behavior according to a new condition in the environment, such as identifies changes in users' needs.
- **Pro-activeness:** This means that agent should act autonomously according to the user's goals. By autonomously it meant agent should act on our behalf.
- **Social ability:** social behavior of agent tends to be the ability of several heterogeneous agents to interact with each other. The social behavior of intelligent agent has been defined by three main activities such as:
 - **Cooperation:** as the name proposes is the ability of agent to cooperate with another agent to achieve specific object.
 - **Collaboration:** suggests that agents complete each other tasks to perform desired actions.
 - **Negotiation:** is the ability of agents to negotiate on behalf of their owners about a specific negotiation object, in order to achieve agreement according to their design specifications [4].

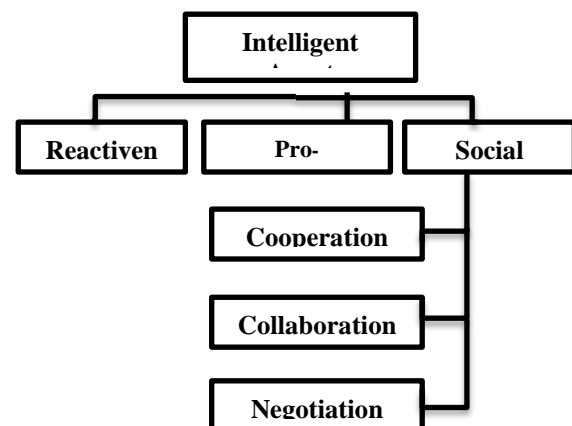


Figure 1: Intelligent Agent's Properties

Agents' negotiation includes various forms of interactions starting from simple form of data interchanges to a complex contract agreement. Applying automated negotiation requires the intelligent application to perform process of data exchange so that it can achieve acceptable agreement. Even though web

applications have been widely available, still not many researches have studied how to make these applications more intelligent to support automated negotiation. In this paper we propose using REST-API and JSON as web implementation technology to be used, which may leads to advance the future of web applications. As well as promoting business applications ability for automated negotiation.

1.2. REST-API and JSON Web Services

REST-API key concept is by representing information as resource; this resource can be text information, document or image. These resources then can be easily transferred and communicated by applications using standard HTTP protocol. Each resource is identified by a resource identifier URI which will be available for any http application request. For instance *http://bookstore/book/{price}*, in this example the book information such as name, price or any other attributes are represented as resource that can be requested and exchanged by http request[5].

The second main principle of REST architecture is the methods or function that can be performed on those resources. REST is using methods that are defined in standard HTTP protocol, the main methods are:

a. GET

Get function is to retrieve resources. Whenever a client agent wants an information resource it can execute Get method for such purpose.

b. POST

Post method can be used by client agent to add resource information to a server agent, not just retrieving.

c. PUT

Put method is to update resource information. By using this method a client agent can put a resource message with a new content to replace resource on the server agent.

d. DELET

Delete is to remove an existing resource. When a client agent calls delete method all member of that resource will be removed [6][7][8].

Third main concept this web service is the format of the resource information being exchanged. The resource data required to be in a standard format. JavaScript Object Notation (JSON) is a standard data exchange format between agents. JSON stores data in a key-value format which is easy to understand by all the object oriented programming languages [9]. For instance:

```
Book info= {
"bookName": "Java How to Program",
"Author": "Paul Deitel"
"originalPrice": "$164.60"
"bookRate": 3.8/5
}
```

2. LITERATURE REVIEW AND RELATED WORKS

Electronic negotiation system is defined as process between two or more agents by intelligent software and

applications [10]. Agent Negotiation has been studied in literature in three main domains includes:

2.1. Negotiation strategies and techniques

There are three common negotiation techniques that have been widely adopted in literature for automated negotiation support. Most of the techniques are adopted from artificial intelligent domain, which can be summarized as following:

- **Game Theoretic Techniques**

Originally game theory is concerned with the study of computational game design, for example chess game. Researchers had mentioned Negotiation and bargaining in the literature of game theory before agent studies have been emerged. Then it rapidly became obvious that the strategies of game theories can be applied between intelligent agents. From this point the game computational theory adopted by automated negotiation researchers since it is applicable between two self-interested agents.

In computer science this domain has been covered by computational theory studies. It can be applied in a particular automated negotiation scenario.

- **Heuristic Techniques**

These techniques determine the computation associated with the negotiation space, by costs. As result the priority is for the optimal solution instead of the good one. It minimizes the complexity of the argument behind the computation decision. In their simplest protocol agents can only accept or reject proposals.

- **Argumentation-based techniques**

The main motivation for argumentation-based technique is to permit extra data to be communicated regarding the negotiation proposals. Thus, in many cases it is important to provide some additional data to establish the agreements. For example rewards can be provided for other agents if they accept the proposal [11][12].

2.2. Negotiation models

The majority of the research studies try to propose theoretical model that deal with specific scenario. It is difficult to cover all models that covered by other researchers, but some related models can be mentioned. The models that going to be covered are about commercial models to automate negotiation between seller agents and buyer agents.

- **Rule-Based Automated Negotiation**

Rules are very promising technique for agent negotiation. The model defines agent roles by using rules that specifically organized to perform a particular response. For instance, rules that support negotiation admission. While others for proposals validity checking, finally rules for other negotiation protocol aspects [13].

- **Learning-Based Negotiation agents**

It is described as learning model framework that allows the agents investing in deeper analysis of the past experience interactions. In this model an agent uses its past interaction experience to learn about other agents.

Then depending on the learned model the agent makes an effective interaction strategy for the future [14].

- **Multi-attributes-Based Negotiation Agent**

The aim of this model is to provide the client agents with extra attributes to negotiation upon for a proposed product. This leads to better satisfying the need of clients and better understanding of agreements proposals [15].

- **Ontology-Based approach**

Intelligent agent supposed to perform on behalf of the owners. To achieve this purpose agent supposed to be supported with negotiation knowledge. It is better to organize this knowledge, in order to facilitate the agent negotiation decision making ability. The ontology-based model applied for such purpose. These agents are using specific protocols that help them to properly organize knowledge. The knowledge can be separated into public and private knowledge experience to preserve the privacy of strategic knowledge during negotiation [16].

- **Combined negotiation agent**

Combined negotiation is a model that suitable for agents which interested in variety of products. As result they can be engaged in many negotiations simultaneously. These concurrent negotiations are independent of each other's as it is with the products. This model tends to help user's agent to be involved in more than one negotiation at once [17].

2.3. Negotiation issues

- **Protocol issues**

Building autonomous agents to perform sophisticated negotiation is very broad area. Assumed the wide range of possibilities, it is difficult to employ universally purpose protocol for each automated negotiation [18].

- **Negotiation objects**

It is the domain that the agents negotiate about, to reach a specific agreement. Each domain can be determined by a single attribute for instance price, while other domains can be determined by hundreds of attributed to be negotiated such as quality, terms and conditions, etc. [1].

- **Agents' Decision Making ability**

In order to meet the negotiation objectives, agents supposed to be design to make decisions based on agreement argumentation. The decision can be at once or through many rounds.

- **Negotiation Technology application**

The limitation use of technology systems to support automated negotiation has been assigned to some factors including:

- The complexity of the electronic commerce applications considered to be a challenge for researchers. The new web-based systems made millions of user exposed to online commerce negotiation practices. This development of new technologies leads to difficulties in managing user's data variety and communication.

- The limitations of information communication protocols that facilitate standard data exchange between the agents.
- Computational and resource constraints that required to manage the delegation of negotiation process [19].

Developing automated negotiation system requires technologies to be implemented and applied. Even technology implementation is an important aspect of applying negotiation agent, but it has been widely covered by the researchers. Most of the research works concerned with negotiation strategies and models. The development of REST-API and JSON for web application created more promising future for implementing intelligent web-based applications. By using JSON, nowadays applications can store and process data more effective and efficient. The REST-API in other hand made it easier to communicate and exchange data.

RESTful application can perform four main functions of data communication using HTTP requests. These four functions are (GET) to retrieve information from other agents. (POST) HTTP request can be used to pass json data to other agents. (PUT) request create and/or update the data on other agents. (DELETE) request makes it possible to delete data on remote agents. Using REST requests the agents can perform all four necessary operations that required in automated negotiation, which they are read, create, update and delete data.

In this paper work we propose an implementation model that emerge REST-API as communication protocol and JSON as data format to perform automated negotiation. The implementation will be for a simple scenario of a seller agent (SA) and client agents (CA). As it has been mentioned this scenario has been covered by other researchers in term of negotiation strategy, protocol and design [20][21][22], while in this paper we propose the technology implementation practice of how to apply this scenario in developing web application.

3. PROPOSED IMPLEMENTATION MODEL

We propose an automated negotiation implementation model that provides the behavioral architecture in form of web application services. This work service is easy to implement framework for completely intelligent application system. It facilitates the basic operations needed by e-commerce platform as well as it is a cost saving technology of development.

The working process: seller agent (SA) provides a book selling service; it sells books with reduction ration depending on client agents (CA) attributes. For instance: client age, profession, quantity, original price and book rate. Many other attribute can be added according to different negotiation situation.

The (CA) look for specific book of his interest. Once the book is matched the SA and CA are going to communicate with each other to start negotiation. The CA has it is JSON data object and apis codes for request behavior, while the SA has its own data retrieved from database stored in JSON format. In addition the SA has

its own API codes for behavior based on number of issues taken into consideration. Here we took five attribute in consideration to demonstrate the implementation work, since the principle is understood it can be applied to any situation and number of attributed.

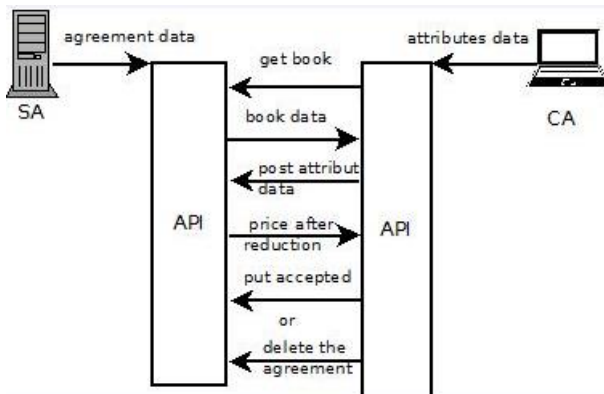


Figure 2: Proposed Automated Negotiation implementation model

As it the figure depicts the client agent (CA) initiate the request for the book using GET API. If the request matched existing book form a book seller agent, then the CA uses POST API to pass attributes in JSON format to the SA. Depending on the CA attributes the agreement data the SA calculate the amount of reduction from the original price and respond it to the CA. finally, if the CA accept the new price it uses PUT API to save the agreement deal or uses DELETE API to delete and cancel the agreement. The implementation algorithm process is as following:

Client Agent (CA): makes the initial request

```
//Protocol: GET
//Content-Type: application/json
url=localhost/SellerAgent
```

//Requested data:

```
From (url) Get book data where
{
  "bookName": "Java How to Program"
  "Orginal Price": "prince"
}
```

Seller Agent (SA): returns book based on get request

```
//Api: GET Api
HTTP 200 OK
{
  BookName: "bookname",
  BookPrice: "prince"
}
//return book data from seller agent databse
query= select bookname, orginal price, author, bookrate
from books
where bookName=BookName
data=execute(query);
```

```
//send back data to CA
// client recives JSON-Data
```

```
{
  "bookName": "Java How to Program",
```

```
"Author": "Paul Deitel",
"orginalPrice": "$164.60",
"bookRate": "3.8/5"
}
```

CA: client agent posts attributes data as JSON

```
//Protocol: POST
//Content-Type: application/json
Attributes={
  "age": 20,
  "profession": "student",
  "quantity": 2,
  "maxprice": 35,
  "minprice": 15
}
```

```
send(Attributes)
```

SA: receives client attributes and calculate reduction

```
// Return a value based on Api get request
HTTP 200 OK
```

```
{
  "studentAge": "age",
  "studentProfession": "profession",
  "requestQuantity": "quantity",
  "maxprice": "maxprice",
  "minprice": "minprice"
}
```

```
// Return reduction ratio
Select reduction_attributes from book_reduction
Where reduction_attributes=CA_attributes
```

```
//calculating reduction
Reduction_price=Orginal_price*Σk=0n attributes
```

```
//response will be
send(reduction_Price)
```

CA: Final Confirmation

// if the reduction price suits the Client agent preference then the final confirmation will be ok

```
//Protocol: PUT
//Content-Type: application/json
```

```
confirm={
  "id": "client hashing id",
  "confirmState": "agreed"
}
open("PUT", url)
send(confirm)
```

//Otherwise agreement is canceled

```
confirm={
  "id": "client hashing id",
  "confirmState": "canceled"
}
open("DELETE", url)
send(confirm)
```

4. PROPOSED APPLICATIONS PROTOTYPE

The result of the proposed model and code of the above section simply developed into a bookstore business intelligent applications. The code in the above section represents the basic principles of the applied prototype implementation; the detailed implementation process can easily be understood by other researchers or developers. In this study work the model developed into a simple prototype application to illustrate how such model can be applied. As it is mentioned earlier there are going to be a books seller agent application which offers books with negotiating price and criteria as it shown in figure below.

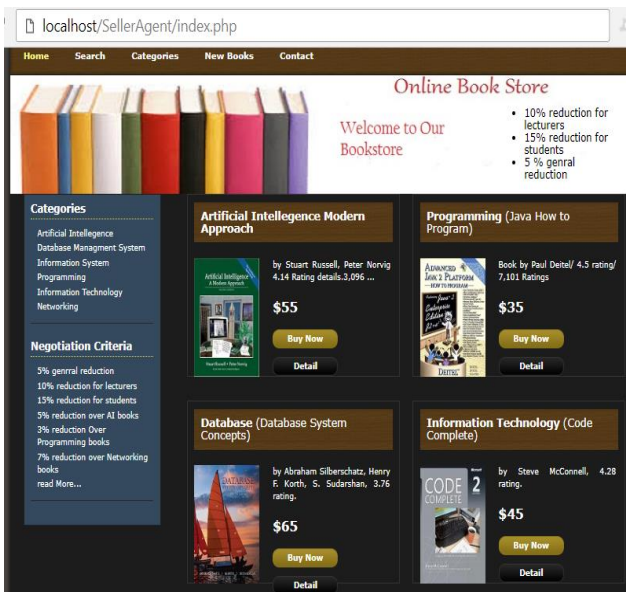


Figure 3: Seller Agent Application

At the left side of the seller application there are negotiating criteria as preferred by seller agent for instance:

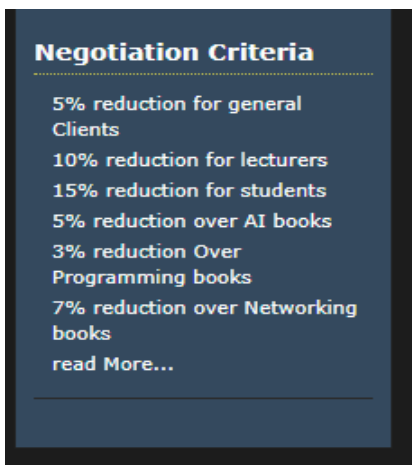


Figure 4: Seller Agent Negotiating criteria

In the other hand there are clients' applications, where buyers can open and sign up there information attributes in order to use these services, as it is shown by the next figure.

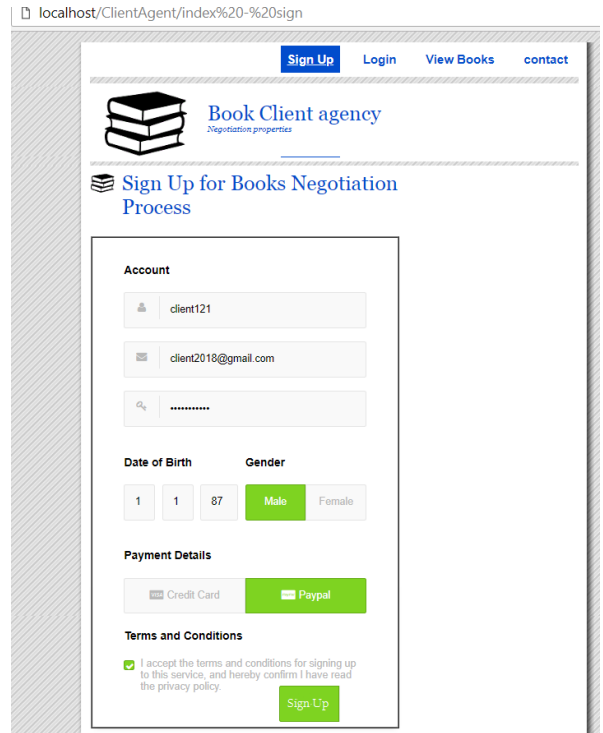


Figure 5: Sign Up for Buyer Agent Application

After client verification then a buyer client can set up his negotiating criteria for his desired books. Then the application can negotiated on his behalf from many seller agents by sending and receiving information using REST-API and JSON data exchange as it shown below:



Figure 6: Buyer Agent Application

As it can be seen from the above main home page of buyer application, a client can add his favorite books with his suitable criteria as below:

Negotiation Criteria

Book Name

Job Profession

Copies

Min. Price

Max. Price

Book Category

[List of Seller Agences](#)

[Add a Seller Agent](#)

[Prices in your area](#)

[Tips about buying](#)

[Terms & conditions](#)

Figure 7: Buyer Agent Negotiation Criteria

The application then can use REST API and JSON data form to communication information with seller agencies and propose list of result to the buyer clients, as below:

Welcome client2018@gmail.com

Your Current List of Negotiated Books

1- Artificial Intellegence Modern Approach

by Stuart Russell, Peter Norvig ,
4.14 Rating details.3,096

1. Seller Agent
[localhost/SellerAgent](#)

2. Original Price \$55

3. Reduction Negotiation

- 15% reduction for student
- 5% reduction over AI books
- 2% reduction for 3 copies

4. Price after negotiation is \$12.1

2- Java How to Program

Book by Paul Deitel/ 4.5 rating/
7,101 Ratings

1. Seller Agent
[localhost/SellerAgent](#)

2. Original Price \$35

3. Reduction Negotiation

- 15% reduction for student
- 3% reduction over Programming books
- 1% reduction for 2 copies

4. Price after negotiation is \$6.65

Figure 8: Buyer Agent Application Negotiation List

5. EVALUATION COMPARISON OF REST-API AND OTHER WEB SERVICES

Even though other web service technologies are exist, but REST-API and JSON are newer technology and have a qualified performance metrics for this purpose. Other familiar technologies that exist for data communication between agents' platforms are:

- **SOAP** (Simple Object Access Protocol): is a standard messaging protocol used to machine-to-machine communication [23].
- **AMQP** (Advanced Message Queuing Protocol): is an open standard messaging middleware to support communication among different platforms [9].
- **CORBA** (Common Object Request Broker Architecture): is a standard protocol which intends to facilitate message communication between different machine platforms [24].

There are metrics which qualifies REST and JSON performance to be suitable for such research purpose. These metrics are proposed for software application evaluation as following [8].

Metrics	REST	SOAP	AMQP	CORBA
Simplicity	Simple to design	More code specification required	Not straight forward	Not much familiarized
Data Format Support	Support JSON and XML	Does not Support JSON	Requires other software tools	Additional library required
Cost	Free-api Open source	Maintenance cost is higher	Software broker tool is required	Commercial implementat ion cost
Execution Time	Shorter response time	Slower response	Execution time consuming	Execution time consuming
Resource consumption	Simple http requests, less resource consuming	More memory consuming	Longer process more resources required	memory leak issue in CORBA program

Table 1 REST-API Performance Preferences

Generally, REST-API became preferable by developers because it is more light weighted, easy to develop and integrated with existing web applications. In addition it uses less bandwidth which makes it preferable for less memory devices such as mobile application [25].

6. DISCUSSION

REST-API and JSON differ from other negotiation services in many ways, as it has evaluated and discussed by other researchers.

REST-API and JSON services are relatively new and open source, which means it has no cost and license restriction for any company. It also a standard format, it can be implemented by any applications using HTTP

protocol and object oriented programming. Other important value of REST-JSON is simplicity in comparison to the other distributed complex processing. It is easy to implement applications using REST with JSON services, because it is closely related to the HTTP protocol. REST is a light-weight alternative to applications with limited resources and bandwidth as it is only using URL to represent information resources.

Finally, we used this web service because despite its simplicity and powerfulness, it has not been implemented for such purpose. Our work contribution intends to apply this technology in intelligent negotiation domain. Such technology can transform the way current online business system work.

7. CONCLUSION

This paper reviews many aspects of multi-agents system in particular the online commerce system. There are many researches have covered the strategy and models of automated negotiation system, but only few research work concerned with implementation practices of how to apply this concept in term of application development. In this work propose a very easy and cost effective technology to implement intelligent web application in order to perform online automated negotiation system. Even though the model is applied for one case scenario, but it can be easily applied for a huge complex e-commerce system.

8. FUTURE WORK

Agents are getting their way through in intelligent commerce applications. There are still much works need to be done as future work plan to make it practically applicable. One important aspect need to be studied is expanding the HTTP protocol behavior for e-commerce domain in precise. Then creating more advanced REST-API routines that can handle more complex and intelligent auctions.

REFERENCE

- [1] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge, and C. Sierra, "Automated negotiation: prospects, methods and challenges," *Group Decision and Negotiation*, vol. 10, no. 2, pp. 199–215, 2001.
- [2] S. Kraus, "Automated negotiation and decision making in multiagent environments," in *ECCAI Advanced Course on Artificial Intelligence*, 2001, pp. 150–172.
- [3] P. Huang and K. Sycara, "A computational model for online agent negotiation," in *System Sciences*, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on, 2002, pp. 438–444.
- [4] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [5] D. Konecny, *Enhanced rest services with custom data*. Google Patents, 2017.
- [6] S. Jade, "REST Services." Software Corporation Limited, Jade Corporation Copyright-2018.
- [7] D. IATA, "RESTful Architecture." IATA documentation. Version 1.5, 24-Feb-2015.
- [8] J. Tihomirovs and J. Grabis, "Comparison of soap and rest based web services using software evaluation metrics," *Information Technology and Management Science*, vol. 19, no. 1, pp. 92–97, 2016.
- [9] J. L. Fernandes, I. C. Lopes, J. J. Rodrigues, and S. Ullah, "Performance evaluation of RESTful web services and AMQP protocol," in *Ubiquitous and Future Networks (ICUFN)*, 2013 Fifth International Conference on, 2013, pp. 810–815.
- [10] I. Rahwan, R. Kowalczyk, and H. H. Pham, "Intelligent agents for automated one-to-many e-commerce negotiation," in *Australian Computer Science Communications*, 2002, vol. 24, pp. 197–204.
- [11] C. Beam and A. Segev, "Automated negotiations: A survey of the state of the art," *Wirtschaftsinformatik*, vol. 39, no. 3, pp. 263–268, 1997.
- [12] G. Lai and K. Sycara, "A generic framework for automated multi-attribute negotiation," *Group Decision and Negotiation*, vol. 18, no. 2, p. 169, 2009.
- [13] C. Bădică, M. Ganzha, and M. Paprzycki, "Rule-Based automated price negotiation: overview and experiment," in *International Conference on Artificial Intelligence and Soft Computing*, 2006, pp. 1050–1059.
- [14] M. Oprea, "The use of adaptive negotiation by a shopping agent in agent-mediated electronic commerce," in *International Central and Eastern European Conference on Multi-Agent Systems*, 2003, pp. 594–605.
- [15] [1]W. Kim, J. S. Hong, and Y. U. Song, "Multi-attributes-based negotiation agent and E-marketplace in customer-to-customer electronic commerce," in *International Symposium on Methodologies for Intelligent Systems*, 2003, pp. 608–612.
- [16] G. Wang, T. Wong, and X. Wang, "An ontology based approach to organize multi-agent assisted supply chain negotiations," *Computers & Industrial Engineering*, vol. 65, no. 1, pp. 2–15, 2013.
- [17] M. Benyoucef and R. K. Keller, "A conceptual architecture for a combined negotiation support system," in *Database and Expert Systems Applications*, 2000. Proceedings. 11th International Workshop on, 2000, pp. 1015–1019.
- [18] M. Louta, I. Roussaki, and L. Pechlivanos, "An intelligent agent negotiation strategy in the electronic marketplace environment," *European Journal of Operational Research*, vol. 187, no. 3, pp. 1327–1345, 2008.
- [19] E. Chen, R. Vahidov, and G. E. Kersten, "Agent-supported negotiations in the e-marketplace," *International Journal of Electronic Business*, vol. 3, no. 1, pp. 28–49, 2005.
- [20] M. Ströbel, "Design of roles and protocols for electronic negotiations," *Electronic Commerce Research*, vol. 1, no. 3, pp. 335–353, 2001.
- [21] M. I. Bala, S. Vij, and D. Mukhopadhyay, "Intelligent agent for prediction in e-negotiation: an approach," in *Cloud & Ubiquitous Computing & Emerging Technologies (CUBE)*, 2013 International Conference on, 2013, pp. 183–187.
- [22] P. Braun et al., "E-negotiation systems and software agents: Methods, models, and applications," in *Intelligent decision-making support systems*, Springer, 2006, pp. 271–300.
- [23] M. W. Khan and E. Abbasi, "Differentiating parameters for Selecting Simple Object Access Protocol (SOAP) vs. Representational State Transfer (REST) based architecture," *Journal of Advances in Computer Networks*, vol. 3, no. 1, pp. 63–6, 2015.
- [24] S. Buciu, *Communications using the common object request broker architecture (CORBA)*. Google Patents, 2017.
- [25] M. W. Khan and E. Abbasi, "Differentiating parameters for Selecting Simple Object Access Protocol (SOAP) vs. Representational State Transfer (REST) based architecture," *Journal of Advances in Computer Networks*, vol. 3, no. 1, pp. 63–6, 2015.