Multi-Agent Auction Based Simulation Tool for an Insurance Policy Market

Egons Lavendelis[1], Janis Grundspenkis[2], [1-2] _Riga Technical University_

Abstract – **The paper presents a simulation tool for automated interactions between insurance companies and their clients during the travel insurance buying process. Insurance deal evaluation model using price and insured risks has been developed based on the study of the Latvian insurance market. The proposed model is used together with well-known agent auction protocols, thus providing a multi-agent negotiation protocol. It allows automating one-to-many negotiations between client and insurance companies simulating electronic insurance policy marketplace. The simulation tool has been developed using the MASITS methodology and tool, thus providing a case study for the methodology and tool for a new type of systems.**

Keywords – **Auction, intelligent agent, insurance information system, MASITS methodology, multi-agent system.**

I. INTRODUCTION

The objective of the paper is to represent research results increasing the interaction automation level between Latvian insurance companies and their clients. At the moment, Latvian companies sell insurance policies on the Internet for the insurance types that allow automatic risk evaluation like travel insurance, civil liability insurance of vehicle owners and some other types. Therefore, one-to-one interactions between a client and a single insurance company are automated. Still, at least in the case of the travel insurance it is possible to view offers of insurance companies only one by one. The potential client must fill in information about him/herself, his/her intended trip and the risks he/she is willing to insure at the website of each company he/she is willing to receive offer from. Thus, finding out offers of all existing companies is time consuming.

Intensive research is conducted to automate different markets during the last years. Still our search for solutions of automation of the insurance market failed because we could not find any approach. Majority of studies from general market automation research direction try to create electronic marketplaces by using multi-agent paradigm [1] - [3]. Traditionally, marketplaces are built so that each stakeholder is represented by an intelligent autonomous agent. In the case of insurance, it means that each insurance company is represented by its agent as well as the client is represented by his/her agent. The use of intelligent agents is motivated by their capabilities to automatically represent their owners and do proactive actions to obtain the user's goals as well as by the possibility to implement the whole insurance system as a set of agents [4]. In this case, the agent that represents the company in the electronic marketplace would also be a participant of the multi-agent system that implements the insurance information system of the corresponding company.

In such a marketplace, there is a need for negotiation protocols to automate interactions between client agents and company agents. Negotiation protocol is defined as a set of rules, which govern the interaction [5]. In case of civil liability insurance of vehicle owners, the negotiations have one attribute, because there is only one criterion, price, in particular, whose value must be negotiated about. On the contrary, markets for other insurance types like travel insurance are more complex, because there is more than one attribute to negotiate about, namely, the price, insured risks and the liability limits. Thus, to automate one-to-many negotiations between insurance companies and their clients there is a need for multi-attribute one-to-many negotiation mechanism. To the authors' knowledge, at the moment there are no one-to-many agent interaction protocols that would be suitable for travel insurance marketplace automation. The paper outlines the analysis of Latvian travel insurance policy market as well as proposes model based on this study for insurance deal evaluation using multiple criteria. The model can be used together with traditional agent auction protocols to implement an automated negotiation mechanism between a client agent and insurance company agents. A simulation tool for such an electronic market has been developed, showing the potential of agent and auction usage to automate the insurance market. The development has been done using the MASITS agent oriented software engineering methodology, which initially is proposed for the development of agent based intelligent tutoring systems. Therefore, the paper serves as a case study for this methodology demonstrating its usability for other type of system – electronic marketplace.

The remainder of the paper is organized as follows. Section II outlines the related work in the area of multi-attribute agent negotiation protocols paying special attention to the multi-attribute or multi-criteria auctions. Section III describes the study of the Latvian travel insurance market and suggests a domain specific model for proposal evaluation in the travel insurance domain. Section IV describes the multi-agent system used to develop the simulation tool and outlines the development process of the tool. Section V gives a detailed description of the developed tool functionality. Section VI concludes the paper and outlines the possible directions of future work.

II. RELATED WORK

According to [6], multi-attribute (multi-issue, multi-criteria) negotiation protocols have been studied widely and represent an important challenge in the multi-agent system community. Several general multi-attribute agent negotiation models exist [7] - [9]. General negotiation protocols are usually based on

multiple negotiation rounds and the notion of concession. The main disadvantage of such protocols is the complexity of agent's individual strategies, because usually these protocols have no dominant strategy. It usually can be solved in case of one-to-one negotiation settings, for example, by using search techniques [9], but to the authors' knowledge the one-to-many setting is too complex for existing techniques, and the current research concentrates on one-to-one interactions [8]. Still, many domains require one-to-many negotiations or even many-to-many negotiations, including the insurance policy market that requires automation of the negotiations between one client and many insurance companies. The problem of agent strategy building in one-to-many negotiations is sometimes solved by involving the user into multi-agent negotiations, like in the architecture proposed in [7]. Still this approach disables building of completely autonomous negotiations. In case of negotiations between the user and insurance companies, the potential client would be involved in the strategic decisions during the negotiations; therefore, the concluding deal would depend on the client's negotiation skills that is undesirable.

A simpler option from the individual agent strategy viewpoint is multi-agent auctions that are one of the most explored and applied forms of one-to-many negotiations. The traditional auctions are based on one criterion or one attribute of the deal, which in most cases is price [10]. However, several auction models for multi-attribute deals have been proposed. Some of these approaches concentrate on one particular auction protocol, for example, David et al. [11] extend the English auction protocol to multi-attribute applications. At the same time, other authors concentrate on general mechanisms that can be used with any auction protocol [12], [13], [14], [15]. Still the key idea behind both approaches is to use a domain specific model for deal evaluation by the auctioneer. Bichler et al. [12], [13] propose using a mediator (named E-Broker) based solution. The mediator is put between the auctioneer and the bidders and the latter two do not communicate directly. The mediator computes a winner and generates a contract based on the bids made.

The objective of the paper is to provide a completely distributed marketplace for autonomous negotiations. Thus, the involvement of mediators in the interactions is not desired. One completely distributed solution is described in [14], [15]. The solution is domain specific for transportation and logistics domains because it is based on the domain specific deal evaluation model. Still, this solution is the closest to the needs of an insurance policy, because it offers autonomous, multi-attribute one-to-many interactions between one client and many sellers.

III. THE MODEL OF THE MARKET

The client's decision to choose a particular insurance deal must be made not only based on the price, but also based on the insured risks. The Latvian insurance market has been analysed to determine the risks that Latvian insurance companies offer for the analysed type of the insurance deals –

travel insurance. The insurance companies collect data about the client that will be insured (mainly the data include name, surname, personal ID number and some status information, for example, if the client is a student) and the planned trip (start and end dates as well as destination). The possible destinations are categorized in 3-4 groups (depending on the company) that have the same price. Additionally, the risks that the client wants to be insured are collected. The study of the offers given by Latvian insurance companies allows concluding that there are four groups of risks that can be insured in travel insurance, namely:

- Health insurance or medical insurance. This group may contain the following risks: medical expenses, repatriation, medical evacuation, dentist services and accidents.
- Risks related to aviation, namely, flight delay, luggage loss and luggage delivery delay.
- Risks related to sports and active recreation. These risks must be chosen to cover expenses that occur due to accidents during the sports events like winter sports (skiing, snowboarding), summer sports, motorsport and other kinds of amateur sports. These risks do not add new insurance liabilities but remove constraints related to the accidents that occur during sports activities and are otherwise not compensated by the insurance company.
- Other risks that do not fit in the groups listed above. The most common risks in this group insured by Latvian companies are loss or theft of the passport, money and personal belongings, civil liability, legal aid and replacement of the client due to his/her health situation.

Scale for the values for every risk has been defined, including the measurement units, minimum value and maximum value. Majority of risks have the scale in EUR, meaning the maximum insurance liability that can be paid out if the risk occurs. Thus, the scale is from zero EUR (risk is not insured) to the maximum insurance liability amount for a particular risk that can be insured by Latvian companies (differs for various risks). The only exceptions are the risks of the group about the sports events. These risks have a binary scale – they either are included in the insurance or not. Additionally, some risks may have deductibles, i.e., the amount for which the client is liable on each loss.

Different clients evaluate particular criteria with different preferred values and importance. Thus, the proposed model allows clients to specify two values for each criterion, namely, the preferred value and the importance of the particular criterion. The specified values allowed measuring the difference between the preferred values and the values proposed by the company. The bigger the difference between the preferred value and the proposed value (only risks where the proposed value is smaller than the preferred one are considered) is, the worse the deal for the client is. This is true for all insurance risks and the price, whose preferred value is as small as possible (0 in an unreal case). This allows calculating the utility U of each deal proposed by the

insurance company to the particular client according to (1). The equation includes two more coefficients expressing the importance of the criterion in the insurance market and importance of the criterion to a particular client.

$$U = \sum_i \frac{x_i - x_i^{pref}}{x_i^{max}} * k_i^{crit} * k(pref_i), \qquad (1)$$

where

x_i – the actual value of the i-th criterion (attribute);

x_i^{pref} – the preferred value of the i-th criterion, that is input by a client;

x_i^{max} – the difference between the maximal and the minimal value (usually 0) of the i-th criteria;

k_i^{crit} – the importance coefficient corresponding to the i-th criteria. The coefficient is needed to deal with the fact that some criteria are more important than others in general. The values of this coefficient are defined by an expert and are in the interval from 0.5 to 2;

$k(pref_i)$ – the coefficient of the priority level specified by the client for the i-th criterion. The model assumes that there are fixed priority levels given to the client to choose from. Current implementation offers priorities from 1 (the lowest) to 5 (the highest). The coefficients have been determined empirically and are in the boundaries from 1 to 10.

The proposed approach allows adapting the model to different users and use cases in the following way. First, a client can specify his/her preferred values together with their importance. Second, the model can be calibrated by changing the coefficients and, as a consequence, making the model suitable for different users and also other market situations.

In the current implementation, the following indexes of criteria are used. Price is criterion number 0, and, in the current model based on the study of the Latvian insurance market, criteria 1-19 are different insurance risks, but the model can be extended with new insurance risks or other parameters of insurance deals like client service or reliability of the insurance company.

The proposed model for evaluation of different insurance deals gives the utility of each proposed deal enabling the comparison of the deals and finding the best deal.

Additionally, the proposed model can be used when the agent knows the utility of the bid that it should make, for example, the utility of the current highest bid in the English auction. The model allows finding the price P that must be offered by the company for its deal with known all other values of the criteria for the offer to have a particular needed utility value for the client. It can be done using (2).

$$P = \frac{(U_{pref} - \sum_{i=1}^{n-1} \frac{x_i - x_i^{pref}}{x_i^{max}} * k_i^{crit} * k(pref_i)) * p^{max}}{k_p^{crit} * k(p)}, \qquad (2)$$

where

n – the number of criteria (attributes) in the model;

p^{max} – the maximum value of the price;

x_i^{pref} – the needed value of the utility of the bid;

k_p^{crit} – the importance coefficient for the price;

$k(p)$ – the coefficient of the priority level of the price.

Thus, the model can be used as a basis for automated negotiations between insurance companies and possible clients, allowing finding the most appropriate deal in the whole insurance market. The multi-agent system that implements automated negotiations is presented in the following section.

IV. THE MULTI-AGENT SYSTEM

The authors propose to implement automatic negotiations between the potential client and insurance companies by means of the multi-agent system. Each insurance company and a client are represented by particular autonomous agents, namely, the client agent and company agents. Additionally, the market monitoring agent is added to monitor the actions done by the agents of insurance companies so ensure that they comply with the legislation. In this case, all messages sent by the client agent and the company agents are also visible to the market monitoring agents.

The agents can automate the negotiations between the client and insurance companies by using some agent interaction mechanism. According to [10], auction mechanisms are simple negotiation mechanisms that are used for one-to-many negotiations that have some common known evaluation mechanism to evaluate the deal for the negotiation participant that is negotiating with other agents to buy some goods. Auction mechanisms have been used in various domains like logistics [14], [15], electronic procurement [13], and labour management [16]. Majority of these researches use auctions to determine one attribute of the deal, while [14], [15] use auctions to determine more than one attribute of the deal. Similarly to these studies, the paper proposes the usage of agent auctions together with the domain specific utility model that allows determining which deal is better for the auctioneer, which in our case is the client agent, while the company agents are bidders. The four most popular auction protocols, according to [10], have been implemented. There are two main differences from the classical implementation of these auction protocols. Firstly, in the insurance policy buying case, the auctioneer has to minimize the price and the bidders prefer to maximize it not vice-versa as it is in traditional auctions. Secondly, the total utility is used instead of just price to evaluate bids. Thus, the auction protocols work in the following way:

- The English auction is an ascending open-cry auction. The auctioneer starts the auction by announcing a new auction and giving the preferred values of all criteria as well as the initial value of the utility that is very bad for the client. The bidders can respond either by refusing to participate or by giving offer that has better utility for the client. The bids are visible to all other participants. The following offers must have better utility than the previous ones. The auction ends when no agent is willing to submit any bid with better utility as the current bid has.
- The Dutch auction is an open-cry descending auction. The auctioneer starts the auction by announcing the starting bid with very good utility for the client, in

particular the preferred values of all insurance risks and an artificially low price. If no agent is willing to propose the deal with the same utility, the client agent increases the price (so decreasing the utility) and asks for new bids. The auctioneer continues to increase the price until one of the company agents gives a bid with at least the same utility.

- The first-price sealed bid auction starts with the auctioneer announcing the preferred values of all criteria. Each agent is allowed to submit one offer. Offers are not visible to other participants. After all participants have submitted their offers (or the end time has come), the auctioneer chooses the offer with the best utility.
- The Vickrey auction is very similar to the first-price sealed-bid auction only it is the second-price auction. In our case, it means that the utility of the winning deal is changed to be the same as the utility of the second best offer. It is done by increasing the price.

Thus, the multi-agent system consists of 3 types of agents – there is a single client agent, multiple company agents and single monitoring agent. The corresponding structure of the multi-agent system is given in Figure 1.
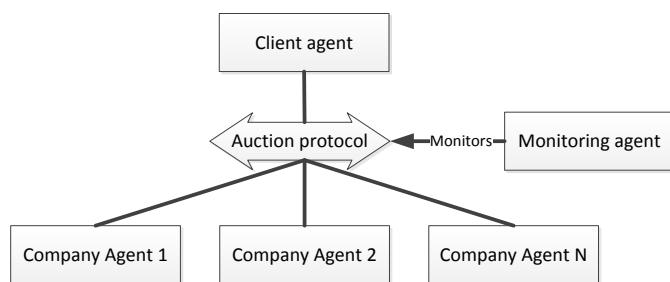

Fig 1. The structure of the multi-agent system.

The agents and their interactions have been defined at the very early stage of the development of the system. The MASITS methodology [17] is initially built for agent based intelligent tutoring system development. One of the main characteristics of its current version is the fact that it is suitable for the development process that is started with the known set of agents. Thus, this methodology has been selected to support the software engineering process for the simulation tool. Therefore, the paper gives one more contribution. It provides a case study for the usage of the MASITS methodology for a new type of systems that have a set of agents known before the beginning of the development process.

The MASITS methodology defines the following phases of the development process: requirements analysis, design (this phase is split into two stages, namely, external design of agents and internal design of agents), implementation and deployment. The development starts with the requirements analysis phase that consists of two steps, namely, the goal modelling and the use case modelling. The goal modelling starts with the definition of the main goal. In our case, it is "Automated policy selling". The process continues with iterative definition of subgoals of the main goal. The goal

hierarchy is defined as a result of this step. The MASITS tool [18] was used for practical development. In the requirements analysis and design phases its main functionality is diagram creation. The tool also provides a solution to consistency checking among the models, thus ensuring that no important parts of one model are missing in the following models. The interface of the tool with the created goal hierarchy is given in Figure 2. Textual description was also created for each goal. The goal diagram together with the descriptions creates a goal model that is the first artefact of the development process.
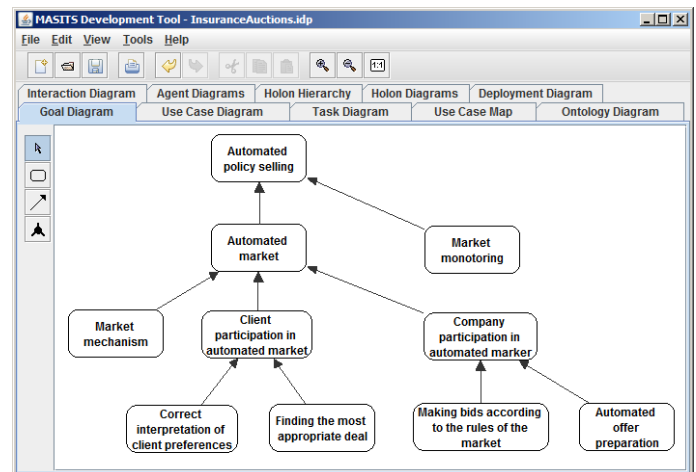

Fig 2. The MASITS tool with the goal diagram.

The use case model is created based on the goal model. The model includes three actors, namely, a client, insurance companies and monitoring institutions. There is one central use case – policy buying that includes the following use cases: priority input, offer preparation, choice of best offer and deal saving. Additionally, there is one more use case for the monitoring institutions – market monitoring. Similarly to the goals, a description is created for each use case. This concludes the requirements analysis phase.

The external design stage (the first part of the design phase) is the next step in the MASITS methodology after the requirements analysis stage. This stage of the MASITS methodology requires the known set of agent classes. The multi-agent system given in Figure 1 is used for this purpose. Thus, there are three classes of agents: a client agent, monitoring agent and company agent. The latter will have multiple instances (denoted with company agents 1…N in Fig. 1). A particular stage of the design answers the question about the functionalities of each agent and the interactions among agents. The functionality of agents is specified in terms of tasks that each particular agent has to do. The tasks assigned to each agent are grouped into hierarchies. The following tasks have been assigned to the client agent:

- Representation of the client in the market – the main task.
- Collecting the client's preferences – the first-level subtask.
- Participation in the auction – the first-level subtask and all the tasks listed below are subtasks of this task.
- Finding the company agents. This is needed to know the potential participants of the auction.

- Announcement of the auction – sending the initial information about the auction.
- Receiving and collecting offers made by company agents.
- Choice of the most suitable offer for the client.

Figure 3 shows the task hierarchy of the client agent that is an example of task hierarchy building in the MASITS tool.
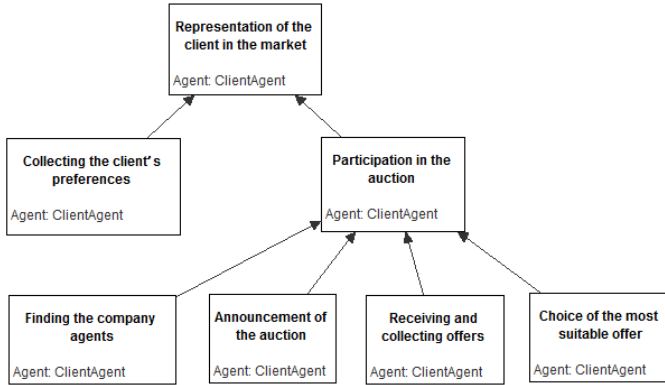

Fig 3. The task hierarchy of the client agent.

The company agent has the following hierarchy of tasks assigned:
- Representation of the company in the market – the top level task.
- Making the final contract – the first-level subtask.
- Participation in the auction – the first-level subtask. All the tasks listed below are subtasks of this task.
- Service registration to the Directory Facilitator agent to make the agent discoverable for the auctioneer.
- Automated offer preparation.
- Making the bids.

The monitoring agent has only one task – market monitoring to ensure that all the offers made comply with the rules of the market.

After defining the hierarchy of tasks for each agent, the interactions among agents become the central concept of the external design. Two diagrams are created iteratively: the messages sent among agents are added to the interaction diagram as well as the concepts and predicates used in the communication among agents are added to the ontology diagram. After the end of the external design, the ontology diagram contains a class hierarchy of the domain concepts (see Figure 4) and predicates (see Figure 5). The hierarchies shown in the figures are refined in the detailed design step, because these initial versions contain only the concepts needed in the interactions, but do not contain concepts and predicates used only inside agents. The interaction diagram specifies all the messages sent among agents. The messages are designed according to the FIPA auction protocols [19], [20]. The client agent sends the following messages to the company agent:
- Call for Proposals (CFP) message containing a predicate NewAuction. This message invites company agents to participate in the auction.

- Call for Proposals (CFP) message containing a predicate NextBid that asks agents to submit next bid that has a better utility than the previous one.
- The accept message with a predicate ChooseDeal confirming that the user accepted the winning proposal.
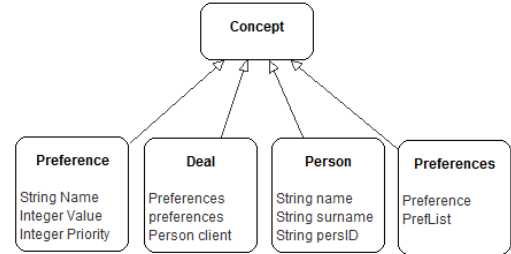

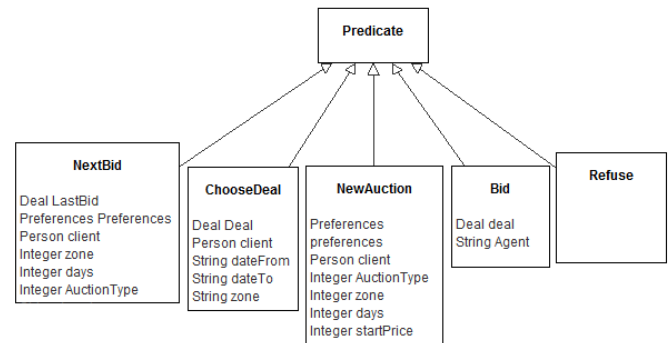Fig. 4. Concepts defined during the interaction design.

Fig. 5. Predicates defined during the interaction design.

The company agents respond with the following messages to the client agent:
- Propose a message containing a predicate Bid with the offer made by the agent.
- Refuse a message stating that the agent is not willing to make a bid.

Additionally, all the messages sent between client and company agents are also sent to the monitoring agent making it capable to monitor if the auction protocol and market rules have been followed by all the participants.

The final part of the interaction diagram is interactions between the client agent and user interface. The MAISTS methodology designs this kind of interaction in the following concepts: the events passed from the interface to the agent and the interface methods invoked by the agent. The interface of the simulation tool passes the following events to the client agent:
- Event InputPreferences contains the client's preferences that have been input in the user interface and inform the client agent that it should start the auction.
- Event ChooseDeal informs the client agent that the client has approved the offered deal.

The only interface method that is called by the client agent is showOffer(), which is used to show the winning offer of the auction to the client.

The detailed (internal) design of each agent in the MASITS methodology answers the question, how the agent will achieve the specified functionality and defines the components of the agent. Agents of the same type are specified as one class of agents. This class implements actions done by agents. The MASITS methodology follows the JADE agent model [21] of single thread per agent. Agent's actions are implemented as behaviour classes that are scheduled for the execution in the agent's thread.

In case of the simulation tool, all company agents are specified as one agent class. This class implements common functionality of the company agents and default methods for all other actions. Common functionality of all company agents is the following:

- Registration of services and cancellation of the registration. The auction mechanism is open for new company agents. This means that the client agent must be capable to find all company agents that currently are present in the system. The directory facilitator service is used for this purpose. Thus, all company agents must register their services.
- Message receiving and adding of corresponding behaviours as well as calling the methods that implement the corresponding actions to process them.
- Participation in new auction that includes specification of the offer by finding the most appropriate values of risks and making the bid according to the strategy. Separate methods have been defined and described below for latter two.

Still, there are actions that are executed differently from one company to another, because the companies do these things differently, for example, the price calculation is done completely differently in each company. Thus, a particular company agent is implemented as a new subclass of the general class of company agents that redefines the methods and/or behaviour classes for the actions that differ company by company. Finding the best combination of risk values and price calculation is the only method that is defined as an abstract method and, thus, must be separately implemented in all subclasses. Some other actions have default implementations and the developer can choose either to use the default implementation of the general CompanyAgent class or to redefine them. The actions that can be redefined are the following:

- Method applyStrategy. This method implements agent's strategies for participation in auctions, i.e., knowing the current bid and its utility for the client as well as the minimum price that can be offered to the client, the method calculates the price that should be included in the bid or decides not to make any bid.
- Method findOptimalValue that finds an optimal value for every risk. It is determined by minimizing the utility

decrease caused by the price increase and the difference between the preferred and actual values of the criterion. In other words, it returns the value that has the best combination of criterion's value and the cost of this value according to the client's priorities.

- Behaviour SaveDeal that saves the details of the deal and gives the insurance policy to the client. Its default implementation saves the deal in the database and creates a document containing general information about the insurance deal.

To summarize, if there is a need to add new agents for new companies that are willing to come into the market, the developer must take the following steps:

- Create a new subclass of CompanyAgent class;
- Implement the method findBestDeal that creates the insurance offer and calculates the minimum price for that offer;
- Implement the actions whose default implementations do not fit the needs of a particular company;
- Compile the agent class and add the agent to the system.

In the implementation phase, the MASITS tool was used to generate the Java code of the agent and behaviour classes that implement the system in the JADE Agent Development Kit [21]. The main task of the programmer was to implement action methods of behaviour classes. A majority of other code was generated. The developed tool is used for simulation purposes and, thus, it is deployed on a single computer making the deployment phase simple, because all agents are deployed in a single JADE container.

V. THE DEVELOPED SIMULATION TOOL

The simulation tool for the travel insurance policy market was implemented as a result of the agent oriented software engineering process using the MASITS methodology. Usage of the tool consists of the following steps:

1. Collecting the information from the client;
2. The client must select the auction type and start the execution of the auction. The remaining process of the auction is done automatically by agents.
3. Data output;
4. If the client is not satisfied with the offer, he/she can change some input data and go through steps 2 and 3 again.

During the first step, the interface of the tool collects information about the person willing to buy a travel insurance policy and about the planned trip as well as the preferences about price and insured risks. According to the previously defined model, client's preferences are collected in the form of the preferred value and its importance. This pair of values is

Fig. 6. The interface of the simulation tool.

collected for all the risks mentioned above. It is done in the left half of the tool's main form shown in Figure 6. The tool's user interface is in Latvian, because the Latvian insurance market is modelled. The information about the client and travel dates must be entered in the upper left area denoted with 1 in Figure 6. The area denoted with 2 in Figure 6 is foreseen for input of preferred values of other criteria. The tabs in this area split the input of criteria into groups that match the groups of risks defined in Section III. Additionally, the first tab (general) is used to input data about the destination and importance of lower price.

Preference input example is given in Figure 7 for input of preferences related to medicine. The left column contains the preferred value of the preference. For example, the entered preferred value of the first criterion "General Medical Expenses" ("Medicīniskie izdevumi") is 80 000 EUR. The right column is used for input of the preference level.

Input of other 3 groups of criteria (criteria related to aviation, sports and others) is done in the same form as the input of medical criteria.

After inputting the data described above, the user can choose the auction type and press the button "Atrast piemērotāko piedāvājumu" to start the second step (see the part of the interface denoted with 3 in Figure 6). The interface passes the auction starting event together with the collected data as parameters. The client agent does the following actions to start the auction of the corresponding type:

1. Finds the company agents currently available at the market. These agents will be invited to participate in the auction.
2. Forms the predicate NewAuction that contains the auction type (four above-mentioned auction types: English, Dutch, First-price sealed bid and Vicrey auctions are available) and the information received

from the interface module, namely, client data, needed insurance policy data as well as client's preferences.
3. Sends the Call for Proposals (CFP) message to the available company agents, thereby starting the auction.

Fig. 7. Preference input of the category "Medicine".

After receiving the CFP message all company agents find deals that most of all fit the client's needs and comply with the business strategy of the corresponding insurance company. Additionally, the minimum price that is satisfactory for the company is calculated. The preference values together with the calculated price form the marginal deal for the company agent. The utility of the marginal deal is the company agent's private evaluation of the auction subject. After finding the marginal deal, the agent is ready to participate in the auction according to its strategy. The default strategy implemented in the simulation tool depending on the auction type is the following:

- In the English auction upon receiving CFP or message with other agent's bid, the agent should make a bid if its private evaluation is better than the current best bid. According to the dominant strategy, the bid should be slightly better than the current best bid, but it is not evident how much better. In the current implementation, the price is calculated in two steps. First, the price that together with optimal values of all criteria would result in the same utility for the client as the current best bid has is calculated by applying (2). Second, the price is reduced by 5% to make the deal slightly better for the client.
- According to [10], there is no dominant strategy in the Dutch auction. Currently, the default strategy implementation is the following. In response to the received message with a current price, the agent checks if the deal is good enough to accept. Firstly, it finds the utility of the deal if the price in the marginal deal is increased by 10%. Secondly, it compares, if such a deal has at least the same utility as the current offer by the auctioneer. If it has, the company agent accepts the proposal by finding the price of the marginal bid that gives the same utility as the current offer by the auctioneer.
- In the first-price sealed bid auctions, the company agent increases the price of the marginal bid by 10% and submits it. This number similarly to the price increase in the English and Dutch auctions is empirically determined and appeared to be suitable for a particular auction protocol.
- In the Vicrey auctions, truth telling is the dominant strategy. Thus, the agent submits its marginal bid.

As described above, the strategy module is one of the components of the company agents that can be changed if the designer of the agent wishes to do so.

When the termination condition of the auction is reached, the data output step (Step 3) starts and the client agent finds the winner according to the rules of a particular auction. The winning proposal is displayed in the right side of the user interface (marked with 4 in Figure 6). The following data are displayed:

- The name of the insurance company whose agent won the auction;
- The price;
- Information about the client (name, surname, personal ID);
- Information about the planned trip (the destination and dates);
- All the insured risks and the values of maximal liability that can be paid out in case of the corresponding risk comes true. The risks that are not insured are excluded from the list.

Alternatively, if no bids are received from company agents (there is no agent willing to propose the deal with the corresponding characteristics), there is no winner and the corresponding message box is displayed.

After the end of data output step, the user can start Step 4. He/she can accept the winning proposal or reject it. If the user accepts it, he/she has to press the button "Noslēgt līgumu". In that case, the interface passes the confirmation to the client agent that sends the corresponding message to the company agent that issues a policy and opens it in a new RTF document. On the other hand, if the client is not satisfied with the result of the auction, he/she can change the preferred values and/or the auction type and start a new auction, i.e., the next iteration that consists of the same steps as the first iteration.

VI. CONCLUSIONS AND FUTURE WORK

Insurance domain specific multi-attribute deal evaluation model has been proposed and implemented in the simulation tool. The implemented tool demonstrates two things. Firstly, it validates the use of auction protocols together with domain specific multi-criteria offer evaluation methods. It has proved that the combination of domain specific model and one of the traditional auction protocols gives a sufficient and at the same time simple automated interaction protocol to automate the interactions in an electronic marketplace. The term "simplicity" here is used to point out the easiness for the participants to find the strategy in the interaction protocol. The auction protocols either have a clear dominant strategy (Vicrey auction in particular) or it is obvious what an agent should do and only the numbers (for example, price increase in the English auction) may vary. Secondly, it demonstrates the possibility of automating interactions between insurance companies and their clients in the travel insurance policy market. The tool would allow simulation of the situation if the automated interaction mechanisms were deployed in the real market. The developed tool shows the possibility of implementing automated interaction mechanisms with multiple companies and also manifests the potential of autonomous agents to save the time and workload of clients and insurance company employees. It also supports studies that determine which auctions give better results for clients and which give better results for companies.

The accomplished research has also achieved its secondary objective – it has served as a case study for the MASITS methodology and MASITS tool, thus testing the possible usage of the methodology in a new area. The development process of the simulation tool has shown that the methodology is sufficient for the auction based system development. The only significant limitation is the interaction design level of the MASITS methodology, where it is not possible to design the sequence of messages or interaction protocols. The main conclusion is that the methodology can be successfully used for other types of systems not only restricted to intelligent tutoring systems, in cases the system meets the following requirements:

- A set of agents must be known before the development process.
- The interaction design at the level of sent messages must be sufficient.

This leads to the new tasks of the future work – related to the extension of the MASITS methodology to fit the needs of

systems that do not comply with above-mentioned limitations. The following main tasks must be fulfilled:

- Introduction of agent definition mechanisms to define the agent classes in case they are not known at the beginning of the design process.
- Extension of interaction design mechanisms to support the interaction protocol design.

The second direction of future work is to carry out the experiments using the developed simulation tool in order to find which auction protocol leads to better deals for clients. It is also planned to develop new strategies for auction participants and to investigate which strategy gives better results for insurance companies in the auctions that have no theoretical dominant strategies.

REFERENCES

[1] E. David, C. Kiekintveld, V. Robu, O. Shehory and S. Stein Eds. Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets. *AMEC and TADA 2012, Valencia, Spain, June 4th, 2012, Revised Selected Papers.* Lecture Notes in Business Information Processing. Volume 136, 160 p. 2013.

[2] J. Debenham and S. Simoff, "Three Technologies for Automated Trading" in *Artificial Intelligence in Theory and Practice*. IFIP International Federation for Information Processing Volume 217, 2006, pp. 405-414. http://dx.doi.org/10.1007/978-0-387-34747-9_42

[3] K. Liu, "A Model of Protocol for Automated Negotiation System", in *Proceedings of the 2nd International Conference on Green Communications and Networks 2012 (GCN 2012)*: Volume 3. Lecture Notes in Electrical Engineering Volume 225, 2013, pp. 407-412.

[4] E. Lavendelis, "Multi-Agent Architecture for Intelligent Insurance Systems" in: *Advances in Intelligent Systems and Computing: 10th International Symposium on Distributed Computing and Artificial Intelligence*, Spain, Salamanca, 22-24 May, 2013. Amsterdam: Springer International Publishing, 2013, pp.439-448. http://dx.doi.org/10.1007/978-3-319-00551-5_53

[5] N.R. Jennings, S. Parsons, P. Noriega and C. Sierra, "On argumentation-based negotiation" in *IWMAS98 submission 1* pp. 68–70, 2000.

[6] M.A. López-Carmona, I. Marsá-Maestre and J.R. Velasco, "Constraint Based Automated Multi-attribute Negotiations", in *Multiagent Systems*, S. Ahmed and M.N. Karsiti (Eds), 2009, I-Tech, Vienna, Austria, pp. 21-54.

[7] C.M. Jonker and J. Treur, "An Agent Architecture for Multi-Attribute Negotiation", in *Proceedings of the 17th International Joint Conference on AI*, IJCAI'01, 2001, pp. 1195-1201.

[8] G.Lai, C.Li, K.Sycara, and J.A. Giampapa. *Literature Review on Multi-attribute Negotiations*, Technical Report CMU-RI-TR-04-66, Carnegie Mellon University, 2006.

[9] G. Tesauro, "Efficient Search Techniques for Multi-Attribute Bilateral Negotiation Strategies", in *Proceedings of the 3rd Internation AI Symposiumon Electronic Commerce*, 30-36. IEEE Computer Society, 2002.

[10] M. Wooldridge, *An Introduction to MultiAgent Systems,* Second Edition, John Wiley and Sons, 2009 p. 484.

[11] E. David, R. Azoulay-Schwartz and S. Kraus, "An English Auction Protocol for Multi-attribute Items" in *Agent-Mediated Electronic Commerce IV*. Designing Mechanisms and Systems. Lecture Notes in Computer Science Volume 2531, 2002, pp 52-68.

[12] M. Bichler and J. Kalagnanam, "Configurable offers and winner determination in multi-attribute auctions" in *European Journal of Operational Research*, Volume 160, Issue 2, 2005, pp. 380-394. http://dx.doi.org/10.1016/j.ejor.2003.07.014

[13] M. Bichler, M. Kaukal and A. Segev, "Multi-Attribute Auctions for Electronic Procurement" in *Proceedings of the First IBM IAC Workshop on Internet Based Negotiation Technologies*, Yorktown Heights, NY, 1999, 13 p. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.8499 [Accessed: October 3, 2013].

[14] J. Grundspenkis and E. Lavendelis, "Multiagent Based Simulation Tool for Transportation and Logistics Decision Support". *Proceedings of the 3rd International Workshop on Computer Supported Activity Coordination – CSAC* May 16-19, 2006, Cyprus, Paphos, pp. 45-54.

[15] E. Lavendelis and J. Grundspenkis, "Simulation Tool for Multicriteria Auctions in Transportation and Logistics Domain", *in Proceedings of the International Conference on Computer Systems and Technologies –* CompSysTech'06, June 15-16, 2006, Veliko Tarnovo, Bulgaria, ACM Bulgaria, 2006, pp. IIIB.17-1 - IIIB.17-6.

[16] K. Sycara, "Problem Restructuring in Negotiation", in *Management Science*, Vol. 37, No. 10, October, 1991. http://dx.doi.org/10.1287/mnsc.37.10.1248

[17] E. Lavendelis and J. Grundspenkis. "MASITS – A Multi-Agent Based Intelligent Tutoring System Development Methodology" in *Proceedings of IADIS International Conference „Intelligent Systems and Agents 2009"*, 21-23 June 2009, Algarve, Portugal, pp. 116-124.

[18] E. Lavendelis and J. Grundspenkis, "MASITS - A Tool for Multi-Agent Based Intelligent Tutoring System Development", in *Proceedings of 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*, Salamanca, Spain, 25-27 March 2009. Advances in Intelligent and Soft Computing Vol. 55., Springer, 2009, pp. 490-500.

[19] "FIPA Dutch Auction Interaction Protocol Specification", Specification number: XC00031F, 2000, 6 p. [Online]. Available: http://www.fipa.org/specs/fipa00032/XC00032F.pdf [Accessed: October 3, 2013].

[20] "FIPA English Auction Interaction Protocol Specification", Specification number: XC00032F, 2000, 5p. [Online]. Available: http://www.fipa.org/specs/fipa00031/XC00031F.pdf [Accessed: October 3, 2013].

[21] F.L. Bellifemine, G. Caire and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley Series in Agent Technology, 300 p, 2007. http://dx.doi.org/10.1002/9780470058411

Egons Lavendelis has studied at the Faculty of Computer Science and Information Technology of Riga Technical University (Riga, Latvia) since 2001 and received his Dr.sc.ing in 2009. The topic of the Doctoral Thesis was "Open Multi-Agent Architecture and Methodology for Intelligent Tutoring System Development"
He started to work as a Researcher in 2005. Since 2010, he has been a Senior Researcher and since 2013 he has been an Assistant Professor and Senior Researcher at Riga Technical University, the Department of Systems Theory and Design. His research interests are agent technologies, multi-agent systems, agent oriented software engineering and intelligent tutoring systems.
E-mail: egons.lavendelis@rtu.lv

Janis Grundspenkis is a Professor at Riga Technical University. He is the Dean of the Faculty of Computer Science and Information Technology, the Head of the Institute of Applied Computer Systems, and the Head of the Department of Systems Theory and Design. He received his Dr.sc.ing. in 1972, Dr.habil.sc.ing. in 1993 both from Riga Technical University. His research interests include artificial intelligence, agent technologies and computer-based tutoring systems.
He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM). He is a full member of the Latvian Academy of Sciences.
E-mail: janis.grundspenkis@rtu.lv