A GPSS SIMULATION MODEL OF INTERACTIONS IN A MARKET-BASED MULTI-AGENT SYSTEM

Eugenia Kalisz and Adina Magda Florea
Department of Computer Science
"Politehnica" University of Bucharest
Spl. Independentei, 313, Bucharest, 77206
Romania
E-mail: {ekalisz,adina}@cs.pub.ro

KEYWORDS

Multi-agent system, GPSS model, agent abilities.

ABSTRACT

Academic and industrial system designers who consider using agent technology to solve application problems are faced with a wide variety of agent system paradigms. Choosing a paradigm or another is not an easy endeavour, as little knowledge or not too many guidelines exist on how to make the good choice. The paper presents a multi-agent system simulation framework for testing different time and load performances of agents depending on some essential characteristics of the agent model, e.g., abilities, negotiation time, and response time to requests. Agents are self-interested and cost-based, as in a market-like model. Their behaviour is driven by the necessity of achieving their own goals while minimising the payment for actions executed by other agents on their behalf. A system facilitator is supporting agent interactions. Starting from a core GPSS model, we have developed several simulation scenarios to simulate and evaluate interaction performances in a multi-agent system, under different assumption of agent behaviour and the tasks assigned to the facilitator. Some comparative results of several simulation experiments are presented and discussed.

INTRODUCTION

Academic and industrial system designers who consider using agent technology to solve application problems are faced with a wide variety of agent paradigms (Muller 1998). Once a given paradigm is chosen, the designer must develop the application, either experimentally or following an existing methodology (Iglesias et. al. 1998). In both cases, agent and system modelling is an essential part of this process. The data and world conceptual model and the agent identification for a particular application are some of the difficult tasks that are faced by a multi-agent systems (MAS) application designer, for which little knowledge or not too many guidelines exist. The work undertaken aims to contribute in solving part of these difficulties.

The paper presents a MAS simulation framework for testing different time and load performances of agents depending on some essential characteristics of the agent model, e.g., abilities, negotiation time, and response time to requests, and on the system architecture.

Starting from a core GPSS model, we have developed several simulation scenarios to simulate and evaluate interaction performances in a multi-agent system, under different assumption of agent behaviour and facilitator assigned tasks. The simulation framework includes a set of parameters that may be set by the user to describe the system under evaluation, while the obtained results and statistics help the user to select the appropriate paradigm.

CONCEPTUAL MODEL

The multi-agent system we consider is formed of a society of self-interested agents that are acting in a distributed environment. Each agent is pursuing its own goals but, in order to achieve them, it has to cooperate with other agents in the system. An agent is endowed with a set of characteristics (abilities) that are relevant for the simulation model:

- the actions the agent is able to perform;
- a cost associated to each action, that may be either how much it is willing to pay for this action to be executed by another agent or how much it would like to receive if it is requested to perform the action;
- a time delay in which one action is to be executed.

All the other characteristics of the agents are hidden in the simulation model (as being irrelevant for our focus of interest) and they may range from simple internal representations and negotiation strategies to more elaborate ones. The only impact these internal characteristics have on the model is at the level of the time necessary to issue a request and to take a decision following an answer to a request. The characteristics to be considered in the simulation framework are selected from the agent model presented in (Florea 1999) but they are quite common to many market-based models of MAS.

The agent behaviour is driven by the necessity of achieving their own goals while minimising the payment for actions executed by other agents on their behalf, as in a market-like environment. The agents are interacting by sending request messages in which they ask for the execution of a given action and by receiving answer messages of acceptance or rejection from the other agents in the system. The system comprises a facilitator that supports agent interactions, as described bellow. The facilitator has knowledge about the agents in the system (their names) and about their associated abilities: actions, with associated costs and time delays, specific to each agent.

One of the main aims of the simulation is to evaluate individual and global system performances under different negotiation schemes (***single-party*** and ***multi-party*** negotiation) and different roles of the facilitator as regarding agent interactions. These roles correspond to different task load balances among the facilitator and agents. Starting from an initial model (called *sp1* in the sequel) that corresponds to a centralised interaction control for a single-party conversation, we have developed other simulation models, namely partially distributed single-party (*sp2*), fully distributed single-party (*sp3*), and centralised multi-party (*mp*). In the initial model, the facilitator operates as follows:

- Receives a request from a source agent for an action to be performed, within specified time and price limits - usually this represents confidential information, needed by the facilitator during the next step. We considered that the source agent sends the request to the facilitator even if it is able to perform the requested action. This behaviour is typical for agents hoping that somebody else could perform the action under better terms. Consequently, the source agent would have the opportunity of good profit with reduced effort.
- Selects the best candidate to perform the requested action. The selection criteria could be different from case to case. In our first model, we considered the same criterion for all requests: time delay inferior or equal to the requested time limit, and lowest price, inferior or equal to the requested price limit. Other selection criteria could be easily modelled, even as functions of the particular source agent and/or the action to be performed.
- Sends a message to the selected candidate. There are several situations to be considered here:
  1. the selection process failed (no appropriate candidate was found); in the initial model we consider that in such cases the request is simply rejected and the source agent is notified accordingly;
  2. if the best candidate is the source agent itself, it is notified by a specific message;
  3. if the requested action should be performed by another agent than the source, the facilitator sends a message to the selected agent.

At this moment, the facilitator becomes free to process other requests, but it will gather answers to the messages issued for the third case above. The legal answers of the agents which received a request are *Yes* (accepted), *Maybe* (start negotiate), or *No* (rejected). The best answer received by the facilitator will be sent to the source agent, which, in case of acceptance, will start communicating with the selected agent to commit it to perform the requested action.

VARIANTS OF THE INITIAL MODEL

The initial model (*sp1*) is that of a centralised ***single-party*** conversation MAS, where the facilitator handles both best candidate selection and negotiation (which, for the moment, is treated as a simple request-answer exchange). It collects statistics concerning the request flow from the moment of its reception by the facilitator until the source receives an answer to it. Two other single-party conversation models were considered, with decreasing degrees of decentralisation:

- the facilitator receives a request from the source agent, selects the best candidate (as in the initial model) and then sends the result of the selection to the source, that further handles the negotiation; this corresponds to model sp2;
- the facilitator receives a request from the source agent and returns at source the list of all candidates for performing the required action; in this case the source handles both selection and negotiation (sp3).

A fourth model (*mp*) that we developed considers a ***multi-party*** conversation between the source and the candidates. After receiving a request from the source agent, the facilitator inspects all candidates, which could perform the required action and sends request's copies to those able to satisfy the action price and time limits. For answer processing it was considered the quite common case of answer reception deadline, as measured by a timer set when sending the copies, i.e., answers arriving after a deadline are simply ignored. From the answers received in time, the facilitator selects the best one, and sends the result back to the source, as presented in Fig. 1. The behaviour of these four models was studied by conducting simulation experiments under several different experimental conditions (sets of parameters), applied to each of them.

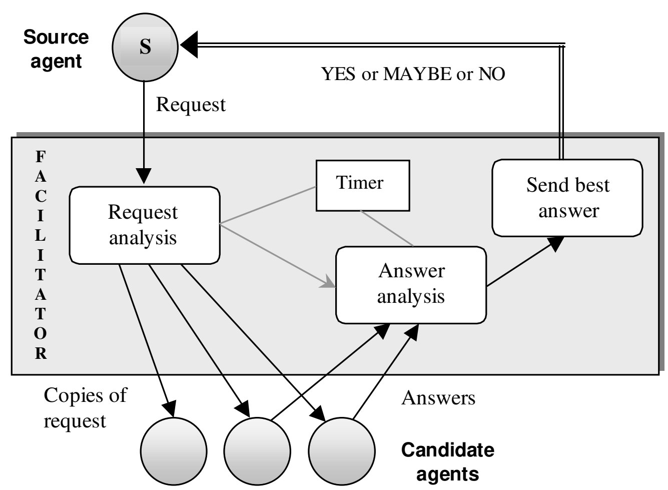

Figure 1. The multi-party conversation conceptual model.

GPSS SIMULATION MODEL

The different versions of the multi-agent system were modelled and analysed using GPSS/H (Banks et. al. 1989), which provides a natural, powerful and flexible modelling framework for discrete-event systems. The GPSS/H simulation model presented in Fig. 5 is that of the multi-party conversation MAS, which includes the facilitator and a number of &NAG agents, able to perform one or more of the &NACT possible actions. The facilitator, as well as the agents, are modelled as simple resources (named facilities in GPSS terminology). In fact, the facilitator is modelled by means of two separate resources: request analyser and answer analyser. Statistics on the waiting queues formed for seizing these resources are collected by the corresponding GPSS statistic entities (queues). Both resources and associated queues have numerical identifiers: 1 to &NAG for the agents, &RAI=&NAG+1 and &AAI=&NAG+2 for request and answer analysers.

Requests are represented by dynamic entities (called transactions in GPSS) with several attributes (parameters). Some parameters are describing the request: it's unique identifier RNO, source agent SAG, requested action ACT, time and price limits - RTL, RPL, and others are describing the request processing status, during the simulation process: number of candidates NC, offer index AA, candidate index XAG, answer code XX. Request copies are modelled by clones of the dynamic entities, which return to the facilitator as candidate answers and are discarded after their analysis.

In this model the facilitator applies the first come - first served rule for requests, but gives higher priority to candidates answers processing. It was assumed that arriving answers are handled by a collateral process and do not imply facilitator's special attention before the start of answer analysis process. It is triggered at time-out or even earlier, in case that all candidate answers arrive before time-out.

The main parameters of the proposed model are represented by the number of agents and their abilities: actions and the price and time associated to each action. These parameters are entered from an user specified input file, along with several time parameters, expressed in conventional time units: communication time &CT (from or to facilitator), candidates processing time spent by the facilitator for the analysis of each candidate and answer, namely &RT and &AT, time-out interval &DL.

Other model parameters within the GPSS model are defined as expressions or functions, which could be easily changed in other versions of the simulation model. For example, the user may set the distribution of answer types or the logical conditions specifying the best candidate. In the current models, the candidate answer type distribution was defined by ANSTYPE function as: 5% *No*, 25% *Maybe* and 70% *Yes*, but it could be changed in a multitude of ways, even by replacing the constants with function calls.

COLLECTED STATISTICS

During a GPSS simulation experiment various statistics are collected, either automatically or according to model specifications. The most *significant statistics collected* by the simulation model are:

- Automatically collected statistics on GPSS resources: &NAG+2 agents and two facilitator's components - request and answer analysers.
- Statistics on waiting queues formed by requests/answers at different points of their flow through the system. There are &NAG+2 queues in our model: &NAG agent queues, and two queues for the facilitator's components.
- Information about request processing time, from its reception by the facilitator until the source receives an answer to it. Such statistics are gathered in tables TAR and TTT, concerning the accepted requests and all requests processed, respectively.
- Information about the number of answers (*No*, *Yes*, *Maybe* and total, respectively), received for each category of requested action, collected in the 4 x (&NACT+1) matrix MANS, where column index identifies the action, except the highest one, which identifies the total column.

All these statistics are available at the end of each simulation experiment, in a GPSS standard output file.

SIMULATION RESULTS

The goal of the simulation process is to gather information about how the model parameters and the components' behaviour affect the overall system performance. The few results presented in what follows are extracted from the GPSS standard output files of *several simulation experiments*.

The experiments were performed on the MAS models proposed in the paper (single-party models *sp1*-*sp3* and multi-party model *mp*), for different values of time ratio R = &CT/&RT, but with &AT = &RT. All simulation experiments used similar data files, and collected statistics for 1000 processed requests.

Table 1. Statistics collected for the multi-party model.

Statistics on model components		R = 1	R=0.5
Request Analyser	load (%)	19.7	39.4
	queue maximum contents	3	5
	queue average contents	0.025	0.131
	queue average time	0.125	0.668
Answer Analyser	load (%)	26.5	53
	queue maximum contents	3	5
	queue average contents	0.023	0.261
	queue average time	0.166	1.855
Agents	average load (%)	27.9	27.9
	queues maximum contents	3	3
	queues average contents	0.059	0.061
	queues average time	1.057	1.088
MAS	mean of request proc.time	10.06	14.61

Table 1 presents part of the statistics collected during simulation experiments on *mp* model.

Tables 2 and 3 present results concerning the request processing time distributions, as collected in GPSS table *TTT*, during simulation experiments executed for different models and/or different values of *R*. These results are illustrates by charts in Fig. 3 and 4.

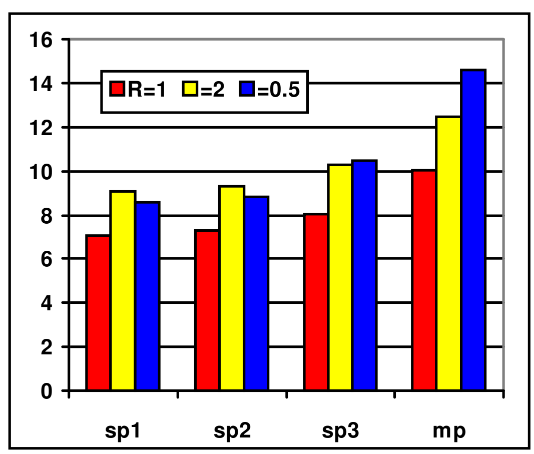

Figure 3. Chart of results presented in Table 2.

Table 2. Mean of request processing time as function of model and R

	model			
R	**sp1**	**sp2**	**sp3**	**mp**
= 1	7.04	7.32	8.02	10.06
= 2	9.04	9.31	10.27	12.48
= 0.5	8.59	8.83	10.45	14.61

Table 3. Request processing time distributions (number of requests observed for each time class).

time class	**model , R**					
	mp, 1	mp, 2	mp,0.5	sp1, 1	sp2, 1	sp3, 1
(0-5]	287	285	247	382	382	350
(5-10]	72	2	40	189	173	210
(10-15]	482	277	167	415	407	388
(15-20]	135	347	286	13	35	47
(20-25]	23	81	172	1	3	5
(25-30]	1	8	62			

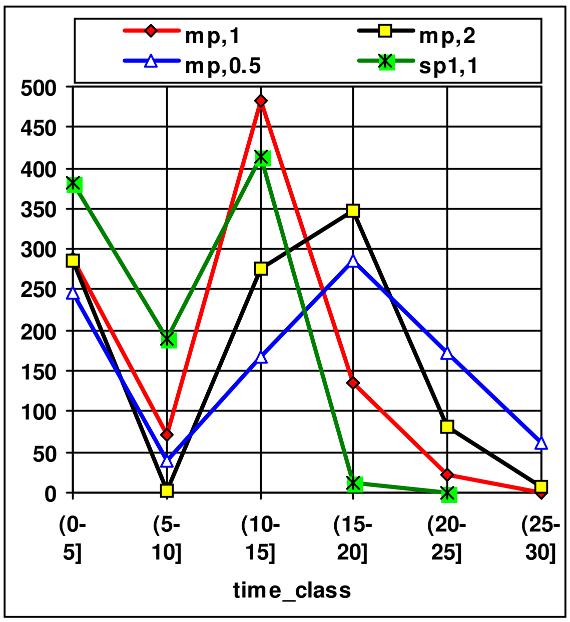

Figure 4. Partial chart of results in Table 3.

CONCLUSIONS AND DEVELOPMENTS

We have presented a MAS simulation framework for testing different time and load performances of agents depending on essential characteristics such as actions that the agents are able to perform and associated costs and time intervals for these actions. The behaviour of the agents is mainly motivated by the gain they may obtain while fulfilling their goals and by the necessity to cooperate with other agents for achieving these goals. A system facilitator is supporting agent interactions. Starting from a core program written in GPSS, we have developed four simulation models to evaluate interaction performances. From the point of view of the global request processing time, the centralised model of single-party conversation (*sp1*) showed to yelled the best results on the test cases considered. The time performances of the multi-party conversation model (*mp*) are not significantly worse than those of the single-party case, which justifies considering also this paradigm, as different negotiation strategies may optimally select the partner in this case. Besides, the number of accepted requests in the *mp* case is significantly higher than that of all *sp* models.

The simulation framework includes a set of parameters that may be set by the user to describe the system under evaluation while the obtained results and statistics help the user to select the appropriate paradigm.

The MAS model could be developed in various ways, by including other types of messages and/or different models of agent behaviour. Some of the possible model enhancements are presented in what follows.

- Besides the action requests, the facilitator could receive messages informing about changes in agent abilities. For instance, an agent could announce that it is willing to add a new action to its abilities or that it made some changes to the attributes of an existing ability.
- If the facilitator or source fails in the selection of the destination agent, it could try to apply other selection criteria.
- Candidate answer selection may vary from one agent to another, either according to global criteria or to agent specific ones.
- Negotiation is performed in multiple steps, candidate agents proposing modifications of the request parameters, as in (Florea 1999, Florea and Cristea 1999).

Of course, many other situations are possible and will lead to new versions of the model. An extended version of the paper and the GPSS simulation programs for all the four models presented in the paper may be found at the URL address http:\\turing.cs.pub.ro\ai_mas\publications.

ACKNOWLEDGEMENTS

This work has been supported by the National Council of Academic Research (CNCSIS) and the Ministry of National Education under Grant No.51, 1999-2001.

REFERENCES

Banks, J.; J.S. Carson II; and J.N. Sy. 1989. *Getting Started With GPSS/H*. Wolverine Software Corporation, Annandale, Virginia.

Florea, A. "Cost-based cooperation of self-interested agents." In *Proceedings of CSCS 1999, the 12th International Conference on Control Systems and Computer Science* (Bucharest, 26-28 May). 62-67.

Florea, A. and P. Cristea. "Multi-Agent Model of Manufacturing Systems." In *Proceedings of WMC 1999, the Second World Manufacturing Congress* (Durham, UK, 27-30 September). 282-288.

Iglesias, C.A.; M. Garijo; and J.C. Gonzalez. "A survey of agent-oriented methodologies." In *Proceedings of ATAL 1998, the 5th International Workshop on Agent Theories, Architectures and Languages* (Paris, 4-7 July). 185-197.

Muller, J.P. "The right agent (architecture) to do the right think." In *Proceedings of ATAL 1998, the 5th International Workshop on Agent Theories, Architectures and Languages* (Paris, 4-7 July). 1-16.

```
        SIMULATE                    -- Multi-Agent System -- Multi-Party  --

*- 1: Facilitator sends request copies to candidates;
*- 2: collects "yes" and "maybe" answers until deadline or all answers received;
*- 3: selectes best answer and sends it to the source

*- model parameters
         INTEGER       &NAG,&NACT,&N    no.of agents,actions,existing abilities
         REAL          &DL,&CT,&RT,&AT  time parameters
         VCHAR*20      &FILENAME        data file name
         GETLIST       &FILENAME        enter data file name
DATA     FILEDEF       &FILENAME
         GETLIST       FILE=DATA,&NAG,&NACT,&N,&CT,&RT,&AT,&DL
         INTEGER       &RAI,&AAI        facilitator's components identifiers:
         LET           &RAI=&NAG+1          request analyzer
         LET           &AAI=&NAG+2          answer analyzer
         INTEGER       &I,&AI,&AY       auxiliary variables
*-- Facilitator information
         REAL          &T(20),&P(20)   info on existing abilities: time, price,
         INTEGER       &A(20),&L(11)     agent, list of action_ability indices
*- action x could be performed by agents &A(&L(x)) .. &A(&L(x+1)-1)
*- agent's &A(j) ability is defined by &T(j) and &P(j)
         GETLIST       FILE=DATA,(&L(&I),&I=1,&NACT)
         LET           &L(&NACT+1)=&N+1
         GETLIST       FILE=DATA,(&A(&I),&T(&I),&P(&I),&I=1,&N)
*- selection conditions
ABLE     BVARIABLE      (&T(PB(XX))'LE'PL(RTL))_      Acceptable time
                       *(&P(PB(XX))'LE'PL(RPL))       and acceptable price
SELECT   BVARIABLE      (&AY'E'0)+(&AY'E'PB(XX))_     First answer
                       *(&P(PB(AA))'L'&P(&AI))_       or better than
                       +(PB(XX)'E'1)*(&AY'E'2)        previously selected
TXX      BVARIABLE     PB(AA)'G'0                     At least one candidate
*-- distribution definitions
REQINT   VARIABLE      RVEXPO(10,5)        Request arrival interval
SOURCE   VARIABLE      RN(2)*&NAG/1000+1   Source agent [1..&NAG]
ACTION   VARIABLE      RN(3)*&NACT/1000+1  Required action [1..&NACT]
TIMELIM  FVARIABLE     FRN(5)*8+2          Time limit (2..10)
PRICELIM FVARIABLE     FRN(6)*10+5         Price limit (5..15)
RPD      FVARIABLE     RVNORM(7,5,1)       Request Processing at Destination
ANSTYPE  FUNCTION      RN(9),D3            Answer type:
0.05,0/0.3,2/1,1                            5% no / 25% maybe / 70% yes
*-- statistics
TAR      TABLE         M1,5,5,12            processing time for accepted req.
TTT      TABLE         M1,5,5,12            total processing time
MANS     MATRIX        MH,4,&NACT+1         answers(type,action)

*-- Request flow --*
         GENERATE      V(REQINT),,,,,6PB,2PL,1PF request arrival
         ASSIGN        RNO,XID1,PF          set request identifier
         ASSIGN        SAG,V(SOURCE),PB          source agent,
         ASSIGN        ACT,V(ACTION),PB          type of action,
         ASSIGN        RTL,V(TIMELIM),PL         time limit,
         ASSIGN        RPL,V(PRICELIM),PL        price limit
         QUEUE         &RAI                 wait for request analysis
         SEIZE         &RAI                 begin analysis
         DEPART        &RAI
*-- Request analysis - request copies will be sent to candidates
         LOGIC S       COPY
         ADVANCE       &RT                  begin request processing
         ASSIGN        XX,&L(PB(ACT)),PB    start with first candidate
         ASSIGN        AA,0,PB              no candidate yet
SELCAND  TEST L        PB(XX),&L(PB(ACT)+1),ENDSEL  while there are candidates
         TEST E        BV(ABLE),1,NEXTC     able to perform requested action?
         ASSIGN        XAG,&A(PB(XX)),PB    store candidate index
         TEST E        PB(XAG),PB(SAG),OTHER  candidate = source?
         ASSIGN        AA,PB(XX),PB         consider as best (for the moment)
         TRANSFER      ,NEXTC
OTHER    ASSIGN        NC+,1,PB             one more copy
         SPLIT         1,RCOPY              prepare request copy
NEXTC    ASSIGN        XX+,1,PB             consider next candidate
         TRANSFER      ,SELCAND
*-- all copies prepared --
ENDSEL   LOGIC R       COPY                 signal "all copies ready"
         RELEASE       &RAI                 request analysis ended
```

```
         ASSIGN        XX,BV(TXX),PB        update "answer" info
         TEST G        PB(NC),0,SENDA       any copies sent?
         SPLIT         1,TIMER              start timer
         PRIORITY      2                    give priority to answer processing
WAIT     LINK          WA,FIFO              put request in waiting_answer list
TIMER    ADVANCE       &DL                  wait for deadline
         UNLINK        WA,LANS,1,(RNO)PF,PF(RNO)         start answer analysis
         TERMINATE                          timeout processed
*-- copies flow
RCOPY    ASSIGN        AA,PB(XX),PB         duplicate offer index
         PRIORITY      1
         GATE LR       COPY                 wait for "all copies ready"
         MARK                               mark sending time
         ADVANCE       &CT
*-- request copy received by destination agent
         QUEUE         PB(XAG)              enter destination queue
         SEIZE         PB(XAG)              request received by destination
         DEPART        PB(XAG)              leave destination queue
         ADVANCE       V(RPD)               processing request
         ASSIGN        XX,FN(ANSTYPE),PB    mark answer type
         RELEASE       PB(XAG)              candidate agent sends answer
         ADVANCE       &CT
*-- answer received by facilitator: discard or put in Received Answers List
         TEST G        M1,&DL,INTIME        arrived after deadline?
         TERMINATE                          discard
INTIME   PRIORITY      2                    higher priority to answer processing
         UNLINK        WA,ANS1,1,(RNO)PF,PF(RNO)  find corresponding request
         TEST E        PB(XX),0,ERAL        request rejected?
         TERMINATE                          ignore
ERAL     LINK          RAL,FIFO             enter Received Answers List
*-- update request status
ANS1     ASSIGN        NC-,1,PB             less answers expected
         TEST LE       PB(NC),0,WAIT        no more answers expected?
*-- facilitator has all answers or those arrived until deadline
LANS     QUEUE         &AAI                 wait for answer analysis
         SEIZE         &AAI                 begin answer analysis
         DEPART        &AAI
*-- start Received Answers Analysis
         LOGIC S       RAA
         BLET          &AI=PB(AA)           store initial offer index
         BLET          &AY=PB(XX)                 and type (yes/maybe)
         UNLINK        RAL,ANS2,1,(RNO)PF,PF(RNO),ENDYP   first answer => ANS2
         GATE LR       RAA                  wait for end of RAA
         ASSIGN        AA,&AI,PB            get best answer (if any)
         ASSIGN        XAG,&A(&AI),PB       store answering agent index
         ASSIGN        XX,&AY,PB            and answer type
ENDYP    RELEASE       &AAI                answer analysis ended
         TRANSFER      ,SENDA               send answer to source
*-- candidate's answer processing
ANS2     ADVANCE       &AT
         TEST E        BV(SELECT),1,NEXTA   better offer?
         BLET          &AI=PB(AA)           store offer index
         BLET          &AY=PB(XX)                 and type
NEXTA    UNLINK        RAL,ANS2,1,(RNO)PF,PF(RNO),ENDY   next answer => ANS2
         TERMINATE                          end of current answer analysis
ENDY     LOGIC R       RAA                  signal end of RAA
         TERMINATE                          end of last answer analysis
SENDA    ADVANCE       &CT                  send answer to request's source
*-- answer received by source - update statistics
         TEST E        PB(XX),1,XXX         request accepted?
         TABULATE      TAR
XXX      MSAVEVALUE    MANS+,PB(XX)+1,PB(SAG),1,MH
         MSAVEVALUE    MANS+,4,PB(SAG),1,MH
         MSAVEVALUE    MANS+,PB(XX)+1,&NACT+1,1,MH
         MSAVEVALUE    MANS+,4,&NACT+1,1,MH
         TABULATE      TTT
         TERMINATE     1

*-- experiment control
         START        1000  simulation stops after processing 1000 requests
         END
```

Figure 5. GPSS/H simulation model of the multi-party conversation MAS.