Cognitive Agents Based Simulation for Decisions Regarding Human Team Composition

Bruno Gonçalves
We, the Body and the Mind Research Lab,
ADETTI/ISCTE
Av. das Forças Armadas 1649-026 Lisboa, Portugal
+351 21 782 64 80
Bruno.goncalves@we-b-mind.org

Sofia Esteves
We, the Body and the Mind Research Lab,
ADETTI/ISCTE
Av. das Forças Armadas 1649-026 Lisboa, Portugal
+351 21 782 64 80
Sofia.esteves@we-b-mind.org

António Lopes
We, the Body and the Mind Research Lab,
ADETTI/ISCTE
Av. das Forças Armadas 1649-026 Lisboa, Portugal
+351 21 782 64 80
Antonio.lopes@we-b-mind.org

Luís Botelho
We, the Body and the Mind Research Lab,
ADETTI/ISCTE
Av. das Forças Armadas 1649-026 Lisboa, Portugal
+351 21 782 64 80
Luis.botelho@we-b-mind.org

ABSTRACT

This paper proposes to use cognitive multi-agent based simulation to facilitate human team formation decision processes. The models of organizational actors are acquired by a data-gathering monitoring system that stores all the interactions that take place in the selected cooperative task. In this knowledge acquisition process, only a relatively small number of team compositions are monitored. The extracted models are used in the agent-based simulation of all the possible team compositions. Simulated team compositions are evaluated and only a small set of the most promising ones is submitted to the human decision maker. An emergence model overrides the multi-agent based simulation in localized situations for which the involved set of agents should not be trusted. The model of the simulation agents is a motivated case-based reasoning system, in which the regular functioning of the CBR is biased to fulfil the agent's motives. The motivation model includes individual motivation and social influence.

General Terms

Algorithms, Experimentation, Theory.

Keywords

Multi-agent simulation, agent motivational model, brainstorms.

AAMAS'06, May 8-12, 2006, Hakodate, Hokkaido, Japan.

1. INTRODUCTION

This paper addresses the KnowledgeAndCulture.org project. The main objective of the project is to use multi-agent simulation of teams of organizational human actors to assist the decision process involved in selecting the best human team for a specific cooperative task. The chosen cooperative task to be studied is the brainstorm, which frames a broad set of specific cooperative processes in a large range of problems and domains.

In some resourceful advanced organizations team formation involves highly specialized psychologists that manually create cognitive maps of all the available people [9]. This approach is a long and very expensive process. In order to improve this extremely resource consuming decision process we propose to use multi-agent based simulation.

The project has reached the first year of its duration therefore no simulation results have yet been produced. However, several other intermediate concrete results have been achieved. This paper presents some of those achieved results and contributions.

1.1 General Approach

Our approach (see Figure 1) is divided in two main stages: organizational knowledge acquisition stage; and simulation and evaluation stage.

In the first stage, a reduced set of human teams is observed while performing the cooperative task (brainstorms, in our case study). The interactions among the human team members are extracted through a computational system and stored into a repository of interaction cases. Only a few team compositions are monitored.

The stored interactions combined with a motivation model used to determine the agents courses of action based on their motives, allows us to create an artificial cognitive agent for each available organizational actor.

In the second stage, different teams of agents are simulated and evaluated according to defined evaluation criteria. Since there are agent teams that have not been observed, we've decided to use models of emergent social behaviour as a way of generating plausible emergent interactions among the agents that represent people that have not been observed working together.

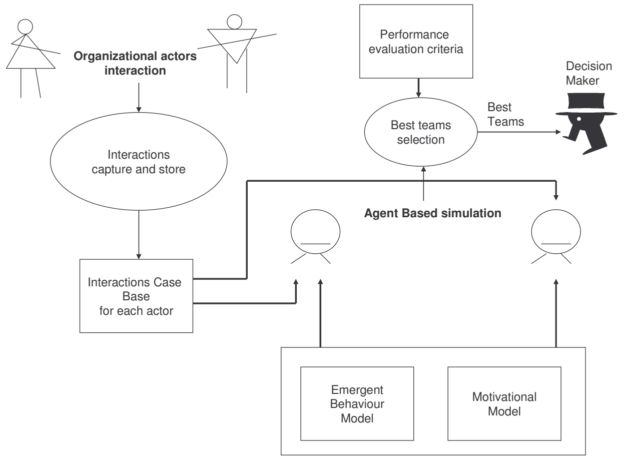

Figure 1. Cooperative work simulation model

The result of the simulation process is the set of a few best teams, which are then considered by the human decision makers. This approach will help focus the human decision process on more promising choices.

1.2 Contributions

The main contributions of the project stem from its decision to use multi-agent based simulation for real concrete organizational problems instead of relying upon toy problems. The requirement to use multi-agent simulation of real organizational problems places constraints on the design of the simulation process and on the computational and conceptual tools to be used. The satisfaction of such constraints continuously produces results for similar future simulation processes.

The first and maybe the most innovative contribution of the project is the designed simulation process. The proposed design has not produced simulation results yet. However the new proposed ideas contribute to and can be essayed in other simulation processes. The proposed simulation design makes two contributions regarding the multi-agent based simulation mainstream: the use of knowledge extracted from actual simulated organizational actors monitored during real interactions in the simulated task, and the integration of cognitive simulation agents with a model of emergent behaviour. In traditional multi-agent based simulation, the agents of the simulation are endowed with pre-defined mental models that are assumed to capture each simulated human actor. In the proposed simulation process design, such kind of pre-defined models are integrated with knowledge extracted from actual organizational actors. Besides, when the behaviour of the simulation agents is not believed to accurately mimic the simulated actors, the model of emergent behaviour takes control over the agents.

The second contribution of the project is the rigorous formalization of brainstorms, which is the cooperative task considered in the project. The formalism models all possible brainstorm interactions between participants.

The third contribution is the definition and implementation of a computational system that supports brainstorm interactions based on the mentioned brainstorm formalization [8], avoiding the complexity of natural language processing. This system monitors the interactions and stores them in a case base for each actor.

The fourth contribution is the definition and partial implementation of a motivated case based reasoning system (CBR). The motivated CBR system integrates a model of motivated behaviour defined in the project with traditional case based reasoning techniques. The flat CBR system (CBR without motivation) has been defined and implemented. Several contributions regarding the use of CBR in large real-life problems have been made. The model of motivated behaviour has been defined and implemented.

The fifth main contribution is the design of two sets of evaluation criteria, domain dependent and domain independent, to be used during the simulation of the cooperative task for ranking the teams according to their performance in the task.

In section 2, we describe the brainstorm formalization. In section 3, we present the model of the simulation agents. Section 4 presents the evaluation criteria used to rank simulated teams. Section 5 presents the chosen simulation scenario. Section 6 provides an overview of related work. Finally, section 7, presents our work's conclusions and plans for future developments.

2. BRAINSTORM FORMAL MODEL

Brainstorms are cooperative tasks in which a set of participants proposes a problem to be solved for which they try to find a satisfactory solution. There are two aspects to be considered regarding brainstorms: the process consisting of proposing a problem and the components of its solution, and the decision process used by the set of participants to accept the proposals being made.

The decision process depends on the structure of the particular group of participants. There are autocratic decision processes, democratic decision processes and hybrid decision processes. The steps of the decision process are not considered in this paper. Final decisions regarding the several alternatives being proposed and discussed during the brainstorm are modelled as if they were the responsibility of a particular virtual actor hereafter referred to as the *Brainstorm Decision Maker*. This option does not commit to a particular decision process. The decisions are made according to the group decision structure but they are represented as if the virtual *Brainstorm Decision Maker* made them.

This section describes the brainstorm processes involved in the proposal of the problem to be solved, the proposal and discussion of the several components of the problem solution. This process will be called hereafter as the *Brainstorm Problem Solving Process*. The formalization of the *Brainstorm Problem Solving Process* encompasses two issues: the formalization of the general problem solving process, and the representation of the individual proposals made by the participants.

2.1 General problem solving process

Traditionally, Artificial Intelligence and Operations Research view problem-solving processes as search problems in the problem space or in the solution space. These search problems are formally represented as graphs in which nodes represent states, and hedges represent actions causing state transformations.

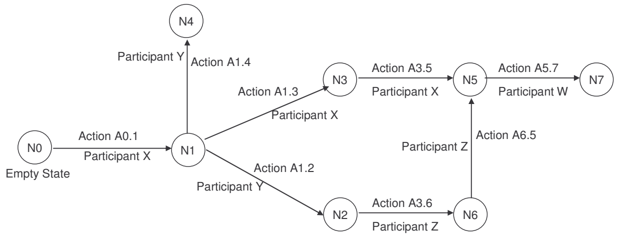

Figure 2. Brainstorm search graph

We propose to represent the general problem solving process used in brainstorms as an acyclic directed graph. In general, nodes represent partial solutions to the proposed problem; and arcs represent actions performed by brainstorm participants that transform the partial solution to which they were applied into a new partial solution.

In Figure 2, the arc from node *N1* to node *N2* is the action *A1.2* performed by participant *Y* in the state contained in node *N1*, which originates the new state contained in node *N2*.

Brainstorm actions always add new elements to brainstorm partial solutions therefore it is impossible to have closed loops in a brainstorm graph. Although it is possible to remove elements from a given partial solution, the removed element is not erased; instead, it is marked as removed. This way, it is possible to capture all the proposals being put forth by all participants during the brainstorm. The result of this policy is that the brainstorm problem solving process is represented by an acyclic graph. Brainstorm graphs are branching structures because it is possible that two or more different actions are performed by possibly different participants in the same partial solution. Different actions performed in the same partial solution give rise to different alternative approaches to solve the same problem.

Brainstorm graphs are not trees because, as shown in Figure 2, it is possible that two alternative courses of action meet in the same brainstorm state (node *N5*, in this case).

By convention, the initial node in a brainstorm graph is the empty state. Usually, the first step in a brainstorm is the definition of the problem to be solved. Problems are introduced in the brainstorm by the addition of a goal to the brainstorm. Among other possibilities, goals may be satisfied by plans of action, by systems design, and by strategy / policy definition, depending on the goal's nature. These elements that directly or indirectly contribute to achieve a brainstorm goal (actions, propositions, objects) are called solution elements. Besides goals and solution elements, brainstorm states may include documents, arguments, different kinds of compositions of those elements, and individual and collective approvals and rejections of proposed elements. As previously explained, collective approvals or rejections are achieved through group decision processes whose model will not be addressed in this paper.

There are brainstorm actions to add and remove (mark as removed) goals, to decompose a goal in sub goals, to add and remove solution elements of any kind (e.g., relationships, classes and objects), to relate existing brainstorm elements (e.g., "this element contributes to this goal", or "from those two elements, it is possible to derive this new element"), to add and remove documentation about any brainstorm element, to add and remove arguments against or in favour of any brainstorm element, to individually and collectively accept or reject any solution element.

2.2 Brainstorm state description

Nodes in the brainstorm graph represent brainstorm states. The formalism used to represent brainstorm states is a propositional version of a SNePS like language [13], which is a specific semantic network representation language. Currently, we use the SNePS graphical notation for being seen by people and an equivalent textual notation to be processed by agents. We are considering using only the graphical notation in future versions of the system if an automated reasoning mechanism can be created that takes advantage of the properties of the graphical notation to improve efficiency.

The brainstorm state representations are used in agent-based simulations. The simulation agents have to mimic the behaviour of the actual organizational human actors therefore they don't need to understand the detailed contents of brainstorm states' propositions. They only need to learn when and how to use the same propositions. Hence propositions are always represented as propositional symbols. However using only propositional symbols is not enough for automated processing therefore the formalism uses classes (i.e., attribute-value descriptions) to better describe propositions.

Since human participants must understand the detailed contents of propositions, the brainstorm state representation language allows adding free text annotations to propositional symbols. Such annotations are often natural language and / or predicate logic based versions of the propositional symbols. Besides being necessary to improve human readability, textual annotations may be used in future applications of the brainstorm representation for instance for the automated generation of meeting minutes or for more demanding automated reasoning systems.

The graphical notation of the brainstorm state representation is a graph in which nodes represent atomic and compound concepts, and arcs represent structural relations between nodes. There are two kinds of nodes: atomic and molecular nodes. Molecular nodes are nodes with arcs to other nodes and represent compound concepts. Atomic nodes do not have arcs to other nodes.

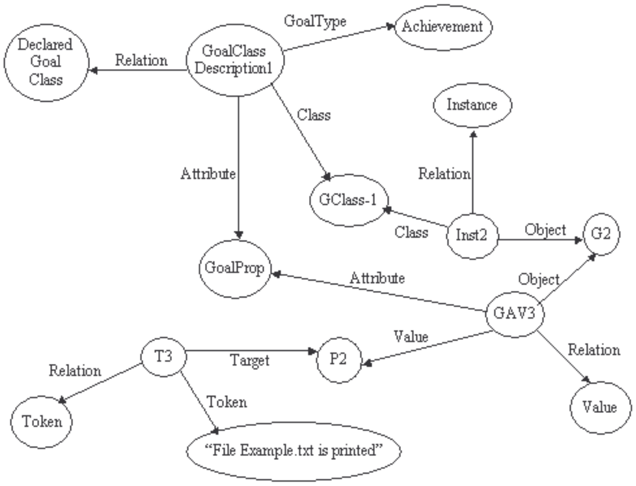

Figure 3. Brainstorm state: Goal definition

In the brainstorm state shown in Figure 3, the molecular node *GoalClassDescription1* describes the goal class *GClass-1* of type *Achievement* with an attribute named *GoalProp* (goal proposition); the molecular node *Inst2* represents the fact that the goal *G2* is an instance of *GClass-1*; molecular node *GAV3* represents the fact that the value of the attribute *GoalProp* of goal *G2* is the proposition *P2*; and finally, *T3* attaches the annotation *"File Example.txt is printed"* to proposition *P2*. All other nodes in Figure 3 are atomic nodes representing atomic concepts. In summary, the brainstorm state of Figure 3 represents the problem to be solved, namely to achieve a state of the world in which *P2* is true (i.e., the file *Example.txt* is printed).

In the brainstorm state of Figure 4, molecular node *Contr23* states that the action of printing file *Example.txt* in printer *LPT1* (molecular node *Act5*) contributes to achieve goal *G2*, which as explained earlier (see Figure 3) is the goal of having the file printed. This statement can now be accepted or rejected by individual participants or collectively by group, according to its decision structure. Acceptance or rejection of this statement would be represented by a propositional molecular node taking another proposition as an argument (*Contr23*). That is, the acceptance / rejection of this proposal would be a proposition taking another proposition as an argument. Since first order logic cannot represent such kinds of knowledge, we have decided to use the SNePS representation language.

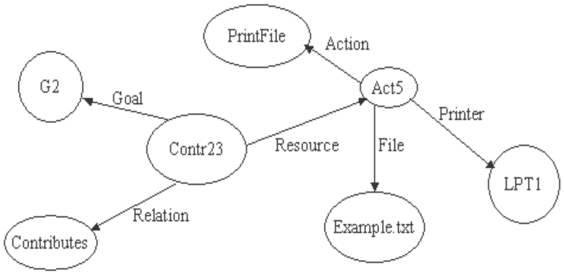

Figure 4. Brainstorm state: solution element

The graph of Figure 4 is textually represented by the following expression:

```
Contr23 # Contributes(resource : Act5 #
PrintFile(printer : LPT1, file : Example.txt),
goal : G2)
```

In the textual representation, the nodes representing relationships (*Contr23*) and actions (*Act5*) are attached to the expressions representing those relationships and action.

2.3 Brainstorm assistance tool (BAT)

The presented formal model allowed us to define and implement a computational system supporting distributed brainstorms with a user interface (BAT – Brainstorm Assistance Tool) [8]. Using the BAT tool, organizational actors participating in the distributed brainstorm only have to choose menu options. Those options are general brainstorm actions like: "propose a discussion problem", "argue", "propose a (partial) solution for a chosen problem", remove previously proposed solutions, and approve and reject previously proposed solutions.

The computational system supporting distributed brainstorms has a distributed peer to peer architecture through which each peer can be used by a brainstorm participant to interact with the others. This system, besides supporting distributed brainstorms, records the interactions of each participant in a case base. The generated case bases are used in the simulation stage, in which artificial agents representing the real organizational actors interact with each other using the BAT system.

For brainstorms related with particular domains it is possible to create plug-ins capable of converting domain specific actions into domain independent actions used by the computational system (e.g. cell phone selection plug-in).

3. BRAINSTORM SIMULATION AGENT MODEL

The agent model (see Figure 5) comprises two sub models: domain specific knowledge used to control the agent's actions; and the motivational model used to select the agent current motives, to compute their strengths, and to influence the operation of the domain specific action control knowledge. Despite being developed for the brainstorm specific application, the agent model is absolutely independent of the domain.

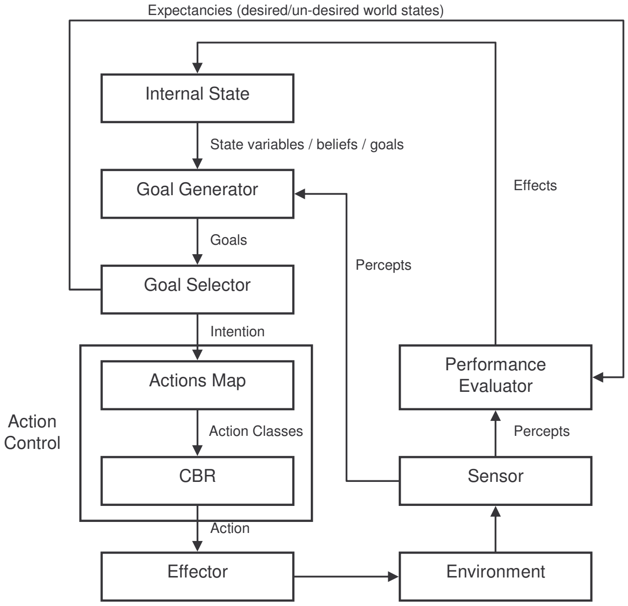

Figure 5. Brainstorm agent model

The domain action control knowledge of the agents of the simulation consists of the case bases extracted and stored during the organizational knowledge acquisition stage. A case based reasoner is used to solve the new problems faced by the agent during the simulated brainstorm. The CBR is influenced by the agent motivation, as determined by the agent motivational model.

According to the proposed motivational model, the agent motivation depends on the agent mental state (e.g., beliefs) and percepts, as suggested by Sloman [14], and also on a set of nuclear internal state variables (e.g., fatigue) [1][3].

We have adopted a BDI (Belief-Desire-Intention) [2][4] approach for representing the agent mental states, mainly for compatibility with a strong agent research community.

The agent acts to maintain the values of its state variables within satisfactory limits. The distance between the current value and the optimal value of a state variable influences the level of motivation to act (e.g. the motivation to sleep depends positively on the fatigue level). We assume that, in brainstorms, the agent's motives (i.e. states that led the agent to act) have two main natures: an individual nature, and a social nature. For this specific activity (brainstorms), we have identified the following nuclear state variables:

Table 1. Brainstorm agent state variables

Name	Type	Description
Intellectual performance	Individual	Perceived level of mental task performance (interventions quality)
Activity level	individual	Level of brainstorm activity (interventions quantity)
Fatigue	individual	Level of psychological fatigue
Empathy with agent X	social	Level of empathy with agent X
Trust in agent X	social	Level of trust on agent X interventions

The agent model implements three main processes: goal generation and selection, action control, and performance evaluation. The process repeats itself indefinitely while the agent is alive. First, one or more goals are generated with corresponding strengths. In the second step, the agent selects the strongest goal that it believes to be achievable. The selected goal and its strength become the agent current intention. The action control mechanism is guided by the generated intention, leading the agent to act. The performance evaluation process assesses the extent to which the agent's motives have been achieved by the enacted behaviour, and updates the agent mental state and state variables accordingly. These processes are described in detail in the next sub-sections.

3.1 Goal generation and selection

The goal generation and selection process described in this section is mainly inspired by the ideas put forth by Sloman [15].

At the goal generation stage, internal states (state variables, beliefs and goals) and external states (percepts) are evaluated, and goals are generated if certain conditions hold.

Goals consist of a proposition representing a desired state of affairs, a numeric value representing its importance, a numeric value representing its urgency, and a numeric value called strength, representing a heuristic combination of the urgency and importance of the goal.

`goal(accepted_proposal(proposal),0.4,0.6,0.5)`

The goal strength is calculated by the following formula:

$$strength = importance \times icf + urgency \times (1 - icf),$$

with $0 \leq strength \leq 1;\ 0 \leq importance \leq 1;\ 0 \leq urgency \leq 1;\ 0 \leq icf \leq 1.$

The *icf* value represents the importance contribution factor for strength.

Goal generation is implemented through a rule-based system (see Figure 6) whose rules are supplied by the agent designer. The rules structure comprises: the generation conditions (beliefs, goals and percepts); a list of alternative mental state and percept conditions (msc/2) used to influence the goal importance; a list of fuzzy rules used to compute the goal importance by evaluating the state variable values; a list of alternative mental state and percept conditions (msc/2) used to influence the goal urgency; a list of fuzzy rules used to influence the goal urgency by evaluating the state variable values; and the goal information structure, which comprises: the goal proposition, the goal failure conditions, the list of goal achievement effects, and the list of goal failure effects.

```
goal_generation_rule(
    (
        percept(brainstorm_problem(Problem)),
        belief(competent_towards(Problem))
    ),
    [   msc(bel(importance(Problem,high)),0.5),
        msc(bel(importance(Problem,medium)),0.25)
    ],
    [   fuzzy_rule_1,
        fuzzy_rule_2
    ],
    [   msc(bel(brainstorm_available_time(0)),0.5)
    ],
    [   fuzzy_rule_3
    ],
    goal(
        accepted_proposal(Proposal, Problem),
        (
            rejected_proposal(Proposal, Problem)
        ),
        [  increase(intellectual_performance)
        ],
        [  decrease(intellectual_performance),
           remove(belief(competent_towards(Problem
           )))
        ]
    )
)
```

Figure 6 – Goal generation rule example

Goal importance and urgency are determined by a set of mental state conditions, and a set of fuzzy rules whose conditions refer to the agent state variables, for example:

- If activity_level is low and intellectual_performance is high then importance is high;
- If fatigue is high then urgency is high;

Only goals whose strength would exceed a certain threshold (i.e. generation threshold) are generated. This prevents the agent from having irrelevant goals.

The agent continuously evaluates its goals in order to generate intentions. At each moment the agent intention is its strongest goal. The importance and urgency of the generated intention is the same as those of the corresponding goal. Intentions are states of affairs that the agent is committed to achieve.

`intention(accepted_proposal(proposal),0.4,0.6,0.5)`

The importance and urgency of goals and intentions are not static. They may change as the result of new evaluations of the goal generation rules. Goals may be removed when their strength decreases beneath the generation threshold. Intentions may be removed when their strength decreases beneath the generation threshold and their associated action control processes abandoned. When an intention is replaced by other, the action control process serving that intention is suspended, possibly being resumed afterwards if the originating goal becomes again the strongest.

The described interruption mechanism is completely based on the ideas put forth by Simon [14].

In order to achieve intended states, the agent uses its action control component, described in the next section.

3.2 Action Control

The first step of this process, implemented by the Action Map component (see Figure 5), is to identify the action classes that are likely to contribute to the achievement of the current intention.

The second step is the selection of the action to be performed. This action selection process is provided by the motivated CBR. Traditional CBR systems [5][11] select the case whose problem is the most similar to the current state of affairs. In the motivated CBR model, there are two criteria for case selection: the similarity of the stored case with the current state of affairs, and the extent to which the action class of the action prescribed by the case (i.e., the case solution) may contribute to the achievement of the agent intention, as determined by the first step. Often the solution prescribed in the most similar case is not adequate to the current problem therefore it must be adapted to fit the current problem.

Intention importance and urgency are used to determine the amount of effort to be employed for seeking the case to be used for the current problem, for the comparison between the current problem and each case that is retrieved from the case base, and for the adaptation to the current problem of the solution prescribed by the selected case. In the presence of an important intention, it is reasonable to expect that the agent performs an exhaustive search process, performs a systematic comparison process, and spends the best of its efforts to adapt the prescribed solution to the current problem. However if the urgency of the intention demands a quick response, it is reasonable that the agent assigns less effort to systematic and thorough processes.

These processes may influence state variables. For instance, exhaustive search processes can increase fatigue; and the discovery of a suitable case can increase the perceived level of intellectual performance.

The intention strength influences the way some actions are expressed in the agent world [1]. For example if the agent has the intention “injured(Creature)” and the CBR chooses the action “attach(Creature)”, it is expectable that the intention strength influences positively the motor strength used in the attach as well as the number of strikes. The influence of intention strength on the intensity of a given behaviour depends on the type of action to be performed therefore the agent designer must supply the strength influence model for each type of action.

For this specific application (brainstorms) we have chosen the CBR approach because it is suitable to generate new behaviours from the record of past cases. Besides, it is easy to capture specific interactions between organizational actors, in specific brainstorm instances as cases in which the case problem corresponds to the brainstorm state and the case solution corresponds to the action performed by the organizational actor in the specified state. For other types of applications, the agent designer has the possibility of extending the motivational model with the most suitable action control component (e. g. a planning algorithm or a rule based system).

3.3 Performance evaluation

The performance evaluator (see Figure 6) checks the extent to which the current/observed state of affairs satisfies the agent current goals. When a goal proposition is observed in the agent world, the agent performance is positively evaluated and the achievement effects specified in the goal generation rule are computed. When the goal failure conditions are observed in the agent world the performance is negatively evaluated and the failure effects are computed. The effects can be to increase or decrease the value of state variables, and the creation and deletion of beliefs. The magnitude of the effects in the state variables depends on the evaluated goal strength, and on the effort already spent by the agent trying to achieve the goal. The effort is calculated dividing the total time used by the action control processes serving this goal by the total time that the goal persisted. The effort is maximum when all the available time was used trying to achieve the goal, and minimum when no time was used.

3.4 Domain dependent components

Sensors and effectors are domain dependent components supporting the agent connection with its environment. The sensor role is to obtain the world sensorial information (percepts) and to deliver that information to the model components that require it. The effector converts the action descriptions (which include the originating intention strength) in effective actions in the agent environment.

3.5 Process description

In the presented model the goal generation and selection, the action control, and the performance evaluation processes run asynchronously. This option ensures that the agent is capable of identifying sudden internal and environmental changes and readjusting its course of action accordingly, at any time. Namely, it is capable of interrupting or suspending its current motives and attending new high priority motives.

4. SIMULATION AND EVALUATION

During the simulation stage, different teams of agents simulate the cooperative behaviour of the represented organizational actors with the purpose of selecting the best teams. Each team performance is then evaluated according to two sets of criteria: quality of the solution generated as the result of the brainstorm (domain dependent), and quality of the brainstorm problem solving process (domain independent). This section addresses these two steps.

4.1 Multi-agent simulation

The simulation process is carried out by a multi-agent platform populated with cognitive agents, each one representing a different organizational actor. The simulation agents will interact within a brainstorm process according to the underlying models of the organizational actors they represent. The main idea is for them to fully interact with each other in the same way as the organizational actors would.

This simulation system will be built on top of the agent platform JADE [18], which was built based on the FIPA specifications [19]. The simulation agents were built using component-based software developed within our research lab: XSP [20].

The simulation agents incorporate a brainstorm engine component built for the BAT system. This engine comprises the interaction capabilities needed for each agent to communicate with the remaining agents in the brainstorm.

The remaining capabilities of the simulation agents will be developed as separated components that will be assembled together with the brainstorm engine. These include the motivational and the domain action control (CBR) components. The emergence behaviour component will be developed as an external sub-system that, under certain circumstances, can influence all agents' activity.

4.2 Simulated Team Evaluation

The evaluation of the brainstorm problem solving process involves the following domain independent criteria:

- Brainstorms taking less time are better;
- Shorter brainstorms (with less states) are better;
- Brainstorms revealing more cooperation among participants are better;
- Brainstorms revealing low levels of indecision and divergence are better.

The first two criteria are simply evaluated and therefore do not need any further explanation.

It is clearly undesirable that a single person proposes most of the solution elements in a cooperative context. Since state transitions are labelled with the identifications of the participants, it is possible to assess if actual cooperation is taking place.

The degree of indecision can be determined considering the number of times that a given solution element is introduced and subsequently removed. And the degree of divergence by evaluating the number of ramifications and reunifications in the brainstorm graph.

5. SIMULATION SCENARIO

We have defined and implemented a demonstration scenario in which to evaluate our proposals. The brainstorm task for the demonstration scenario was that of designing a cell phone specification (set of characteristics) for a specified customer profile. This task is an instance of brainstorm problems in which the solution is incrementally and cooperatively generated through a constellation of components. The same kind of decision process can be applied for instance in house design.

With this purpose, we have identified a set of critical cell phone characteristics, a set of customer's critical characteristics (defining customer's profiles), and also the type of arguments that are likely to be used in the mentioned decision process. The cell phone critical characteristics are those that are more likely to support the customer choice, for example: talk time and video capability. The customer profile is the set of observable user characteristics that can give clues about the customer expectations regarding a cell phone (e.g., the user occupation and the user age). The arguments are used to support or contest solution elements (mobile phone characteristics), based on the customer profile characteristics. For example, a brainstorm participant may argue that, for a customer with eighteen years old, it is necessary that the cell phone has a photo camera. In this case, the argument type is "necessary".

We have implemented a plug-in for Brainstorm Assistance Tool (BAT) that provides a domain dependent graphical user interface for the described cell phone design task.

Besides the domain independent evaluation criteria (see Section 4.2), the brainstorm evaluation process will be based on the user feedback regarding the reached cell phone design decision.

6. RELATED WORK

Agent based simulation is being used in a large variety of problems of different scientific areas, namely it has been used in modelling social processes with the purpose of structuring and exploring theoretical hypothesis. In the mainstream multi-agent based simulation approach, the knowledge used by the agents of the simulation is endowed to the agent by the agent designer [16] [17]. In our approach, the domain knowledge used by the agent is previously extracted from the actual human organizational actor being simulated. This knowledge extraction process is done through a computational system that monitors the actual human actors performing cooperative tasks of the same class of those used in the simulation.

To conveniently simulate the expected behaviour of organizational actors, it was essential to adopt a cognitive approach, namely an agent that has explicit and symbolic representation about itself, about the other agents and its environment, and reasoning capabilities. For those representations, the BDI architecture [2][4] has been chosen since it is a successful and respected theory of human behaviour.

BDI architectures do not formalize the way goals are generated, and they also do not formalize the way goals are selected to become intentions. Some authors [10] have implemented a motivational architecture that works like an extension to the BDI principles, bridging the previously mentioned gaps.

According to Savage [12], motivation is a process where the individual decides what is important to its continued existence and determines the means of achieving the relevant goals. This process is responsible for determining the deliberative human behaviour in all activities, including brainstorm activities.

Simon [14] and Sloman [15] have presented complex and fairly complete conceptual models of motivation, but none are described with the necessary detail for being expressed computationally. Our approach is inspired in the ideas put forth by these two authors but providing the necessary detail for implementation.

Avila-Garcia and Canãmero [1] and Breazeal [3] have presented computational motivation models. These models are far from implementing the properties identified by Simon and Sloman as optimal. No attention is paid to the influence of mental states (beliefs, goals and intentions) in the behaviour choice.

Motivation, besides influencing action control, also influences cognitive processes. This influence is determined by epistemic motivators [6], [7]. This kind of motivators influences the idea formulation, planning and any other kind of reasoning, including CBR. In our motivated CBR approach the amount of effort to be used in the various CBR processes (search, comparison and adaptation), is determined by the importance and urgency of the motivation (intention) being handled, and the selection of the case to be used depends on the extent to which the prescribed action contributes to satisfy the agent motivation.

7. CONCLUSIONS AND FUTURE DEVELOPMENTS

This paper describes a decision making framework regarding human team formation in which a combination of cognitive multi-agent based simulation and an emergence model is used to simulate and evaluate the performance of all possible team compositions in order to identify a small set of the most promising ones.

In the described framework, the domain specific action control knowledge of each available organizational actor is acquired through a computational system designed to support his or her interaction with the other organizational actors in a selected collaborative task. The acquired models are used by cognitive agents in the simulation stage. This process considerably reduces the human decision-making workload.

The project selected the brainstorm as the test bed collaborative task, created a general model of domain independent brainstorms, implemented the computer system that supports distributed brainstorms, defined the models of the agents of the simulation, implemented the CBR system used for the agents model, and defined the initial set of domain independent criteria used in the performance evaluation of each simulated team composition.

An important contribution of this project is the proposed agent model. This model is an extension of BDI models, since its motivational model bridges two important gaps in that class of models: goal generation, and goal selection (i.e., intention formation).

Recently we have realised that the brainstorm state representation for realistic examples becomes impracticably huge. So we will move towards a sentence based representation. The case comparison is also very resource consuming. Nevertheless, we expect the model of motivation will judiciously constrain the amount of computation assigned to each of the agent’s tasks.

The project next steps are to implement the sentence based representation for the brainstorm states, to design and implement the emergence model, and to run the simulation process.

8. ACKNOWLEDGEMENTS

This work is supported by the FCT (Foundation for Science and Technology) National Project KnowledgeAndCulture.org contract nº POSI/SRI/46582/20027.

9. REFERENCES

[1] Avila-Garcia, O., Cañamero, L. Comparing a Voting-Based Policy with Winner-Takes-All to Perform Action Selection in Motivational Agents. In *Proc 8th Iberoamerican Conference on Artificial Intelligence (IBERAMIA'02)*, 2002.

[2] Bratman, M. E. *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.

[3] Breazeal, C., Scassellati, B. Infant-like social interactions between a robot and a human caretaker. *Adaptive Behavior*, 8, 1, (2000), 49–74.

[4] Georgeff, M. P., Barney P., Pollack, M. E., Tambe, M., Wooldridge, M. The Belief-Desire-Intention Model of Agency. *ATAL 1998*, (1998), 1-10.

[5] Kolodner, J. *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann, 1993.

[6] Kruglanski, A. W., Klar, Y. Knowing what to do: on the epistemology of actions. In Kuhl, J. and Beckmann, J. (eds): *Action Control: from cognition to behaviour*. Springer, Heidelberg, 1985, 101-128.

[7] Kunda, Z. The case for motivated reasoning. *Psychological Bulletin*, 108, 3 (1990), 480-498.

[8] Lopes, A., Esteves, S., Gonçalves, B., Botelho, L. M., BAT:Brainstorm Assistance Tool. *Mobile Computing Meets Knowledge Management Workshop*, Prague, Czech Republic, 15-17 September, 2005.

[9] Louçã, J. *Cartographie Cognitive, Réflexion Stratégique et Interaction Distribuée: une Approche Multi-Agent*. PhD Thesis, Paris IX Dauphine University and Sciences Institute of Lisbon University, 2000.

[10] Norman, T. J., Long, D. Alarms: An implementation of motivated agency. In M. Wooldridge, J.P. Muller, and M. Tambe, editors: *Intelligent Agents: Theories, Architectures, and Languages*, LNAI 1037, Springer, 1996, 219-234.

[11] Pal, S., Shiu, S. Foundations of Soft-Case-Based Reasoning. John Wiley & Sons, 2004.

[12] Savage, T. The grounding of motivation in artificial animals: Indices of motivational behaviour. *Cognitive Science Research*, 4 (2003), 23-55.

[13] Shapiro S. C. An Introduction to SNePS 3. In Bernhard Ganter & Guy W. Mineau, Eds. *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Lecture Notes in Artificial Intelligence 1867, Springer-Verlag, Berlin, 2000, 510-524.

[14] Simon, H. A. Motivational and emotional controls of cognition. *Psychological Review*, 74, 1 (1997), 29-39.

[15] Sloman, A. Motives, mechanisms, and emotions. *Cognition and Emotion*, 1 (1987), 217-234.

[16] Sycara, K.: MultiAgent Systems. *AI Magazine*, 19, 2 (1998).

[17] Wooldridge, M. An Introduction to Multi Agent Systems. John Wiley & Sons, 2002.

[18] JADE – Java Agent Development framework – “JADE, a White Paper” http://exp.telecomitalialab.com/articolo.asp?id=2&idart=1.

[19] FIPA – Foundation for Intelligent Physical Agents specifications - http://fipa.org/repository/index.html

[20] XSP – eXtended Service Platform - http://we-b-mind.org/EN/software.html#XSP