COGNITIVE AGENT BASED SIMULATION PLATFORM FOR MODELING LARGE-SCALE MULTI-LEVEL SOCIAL INTERACTIONS WITH EXPERIMENTAL GAMES

Maurice Grinberg, Emiliyan Todorov

Abstract*: The paper introduces a new multi-agent architecture for simulations of complex social interactions with cognitive agents of different sophistication using experimental games. Such games are the Prisoner's dilemma, Chicken, Battle of the Sexes, Ultimatum, Trust, and Dictator games, Public goods game, and other important games that have been used to investigate social dilemmas. The main principles and innovative components are presented together with the design and first implemented components of a computer architecture aimed at large scale distributed calculations allowing for simulations involving millions of interacting agent living in environments built on games. The main goal of the ABM architecture is to build and explore societies of artificial cognitive agents based on rich social interactions modeled by games with agent implementing various computational and cognitive models. The platforms introduces several innovative tools like agent and environment spaces of features and information, interfaces for cognitive models incorporation, possibility to build social environments on experimental game-theoretic games.*

Keywords*: agent-based modeling, cognitive modeling, social interactions, experimental game theory, parallel, distributed systems*

ACM Classification Keywords*: I.6 Modeling and Simulations*

Introduction

Social interactions have been subject of intense interest in many scientific fields like biology, psychology, social psychology, and philosophy. The main focus of this research was on cooperation among various agents like bacteria, animals, humans, etc. The importance of the understanding of the mechanisms behind cooperative behavior, i.e. going far beyond self-interest, is related to the understanding of the main driving force of evolution and human society and culture.

Any exploration of the mechanisms of cooperation in strategic interactions requires appropriate approaches and models. One of the most influential approaches is Game Theory (GT) proposed by Morgenstern and von Neumann [Morgenstern and von Neumann, 1947]. Although GT is a normative and not a descriptive theory, it allowed building a formal theory based on preference for outcomes and expected utility maximization. GT makes several strong assumptions which make possible the derivation of the expected utility maximization principle. For instance GT assumes that the definition of the game by its payoff matrix (strategies and outcomes) and everything that can be deduced from it is common knowledge, i.e. players know it, and they know that the other players know it, and that they know that the other players know that they know, etc. Additionally, players are assumed to be instrumentally rational which means that they always chose strategies which maximize their own payoff, based on their knowledge and beliefs, the latter assumed to be also common knowledge.

Although many discrepancies between extensive empirical data from experimental and behavioral game theory [Camerer, 2003; Colman, 1995; Fehr & Fischbacher, 2004; Van Lange et al. 2013] and GT have been established, the main idea of GT to model social interactions and more specifically social dilemmas with games has been widely accepted.

Some recent theoretical approaches assume people maximize some utility function as in GT but propose new decision making mechanisms like Cognitive hierarchy theory [Camerer, Ho, & Chong, 2004], Stackelberg reasoning, and team reasoning [Colman, Pulford, & Lawrence, 2014] which involve taking into account a model of the opponent and her payoffs. Related models are Social projection theory, e.g. [Acevedo & Krueger, 2005] and other-regarding social values and preferences [Fehr & Schmidt, 2006; Van Lange & Rusbult, 2011].

Some more minimalistic models of decision making are based on reinforcement learning [Camerer, Ho, & Chong, 2002; Erev & Roth, 1999; Grinberg, Hristova, & Lalev, 2010; Macy & Flache, 2002] or on heuristics (e.g. the ‘avoid the worst’ heuristics) [Gigerenzer & Goldstein, 1996; Krueger, 2014]. An interesting perspective in experimental games is the concept of psychological games [Geanakoplos, Pearce, & Stacchetti, 1989] in which the choices of the agents depend not only on the payoffs of the game and the choices made but also on the beliefs about the other agents, and their beliefs and intentions. This approach allows accounting for the emotions of other players and their influence on game outcomes and equilibria.

All these approaches have their strengths and weaknesses and are complementary in the sense that none of them can account fully for the data and there are many evidences showing that one and the same player can use more than one of them depending on the characteristics of the games [Camerer, 2003; Colman et al., 2014; Fehr & Schmidt, 2006] The brief discussion of the various approaches to investigate the mechanisms of decision making in experimental games shows the complexity of the factors that have been considered. In most models of decision making, the influence of GT is quite strong and they try to modify the way players reason about the game in order to account for experimental results (e.g. by using other-regarding strategies). Some of the models start by defining a utility function or a quantity which reflects the attractiveness of a strategy which then allow to determine the best move or the probability of making such a move [Flache & Macy, 2002].

One possible development [Chater, 2015; Sun, 2006] is to apply the progress made in the field of cognitive modeling for agents in simulation of decision making in experimental games. Such attempts are not very common in the literature so far (e.g. [Grinberg & Lalev, 2008; Taiji & Ikegami, 1999; West, Lebiere, & Bothell, 2006]). In general, players are not explicitly considered as cognitive agents with specified perceptual capabilities, attention mechanisms, memory, etc.

Another relatively distinct, but very influential approach for exploring social interactions and their evolution is Evolutionary game theory [Gintis, 2009; Nowak, 2006; Maynard Smith, 1982]. One of the most influential example of this approach is the study of cooperation in iterated Prisoner's dilemma game tournaments [Axelrod, 1984; Axelrod, 1997; Axelrod & Hamilton, 1981] and the demonstrated advantages and power of agent-based model simulations.

The experience of this line of research showed that, in some situations, simulations provide solutions and evolutionary stable strategies [Adami et al. 2015] which are beyond the ones expected based on GT. These and many other results demonstrate the power of multi-agent simulations, based on rigorous mathematical treatments [Shoham & Leyton-Brown, 2008; Epstein, 2006, 2014].

The goal of the present paper is to introduce the idea and the first implementation steps of a novel approach which aims at building a flexible large scale distributed multi-agent platform entirely based on experimental game-theoretic interactions, agents implementing various computational and cognitive models of decision making, and various learning and evolutionary approaches. The platform focuses on the multiple roles an agent can have depending on the group, environment, and the specific interactions (family, company, country, etc.) they imply. A central question of interest is the interactions among these social roles related to cooperation and competition. Part of the social theoretical background of the platform was broadly inspired by the social relational model theory of Alan Fiske [Fiske, 1992; Fiske & Haslam, 1996; Fiske, 2012]. It posits four main relationships which according to this theory underlie any more complex relations: communal sharing, authority ranking, equality matching, and market pricing. According to Fiske [Fiske, 1992], these four relationships can be regarded as psychological models that underlie human sociality. This theory has a considerable empirical support but what is more important here is its amenability to game theoretic terms one version of which has been explored in [Grinberg, Hristova, & Borisova, 2012; Hristova, Grinberg, Georgieva, & Borisova, 2013].

In our agent-based platform, we want to provide the possibility to build social environments consisting only of games that model various social interactions (for a formal multi-agent approach in game theory see [Shoham & Leyton-Brown, 2008]). For instance, some of these games or combinations of them can model the four relationships in Fiske's relational model theory but could also stand for interactions among artificial agents or artificial agent-human interactions [Grinberg, 2011; West et al., 2006]. This essential capability of the platform will allow for the construction of complex environments in which the dynamics of cooperation among agents of various sophistication can be explored.

The agents in our simulation environment will be modelled after the general model of a cognitive architecture (see e.g. [Grinberg, 2011; West et al., 2006] providing an interface appropriate for the implementation of agent models from the main modern approaches in cognitive modelling (computational models, connectionist models, dynamic system models, Bayesian models, etc.).

A third innovative aspect of the multi-agent platform is the conceptualization of agents as multi-dimensional vectors in a space spanned by the various characteristics of the agents which can include not only spatial and temporal localization but also specific parameters (with a predefined distribution or learned), history of strategy choices, accumulated payoffs in various games, etc. Such multi-dimensional agent space would allow finding closeness and similarity among agents and providing information for in-depth analysis of the structure of the artificial agent societies. This information also would allow for the analysis of the complex network structure that emerges out of the multi-level interactions and apply social network analysis to understand better the results of the simulations.

A fourth important feature of the environment is the possibility to run simulations involving large number of agents (up to billions) in order to be able to explore large scale phenomena of multi-agent interaction in a multidimensional space.

While the theoretical background behind the multi-agent platform with characteristic examples will be presented elsewhere, in this paper, we want to report the progress made so far in its implementation.

Agent Based Modeling

Agent Based Modeling (ABM) is an approach for modeling and simulating complex systems composed of autonomous agents interacting with one another [Macal & North, 2010; Epstein, 2006]. The basic idea is to represent inhabitants and artifacts of the real world as mock-up agents inside an artificial environment. Then let them communicate and act with one another inside it, according to specific rules and observe their evolution. This way agents can influence each other, learn from their experience and modify their behavior to survive in an ever changing environment. The environment modifies the behavior of the agents and the agents modify the environment, which leads to a system which is "…therefore, emergent on the interaction of the individual parts." [Barnes & Chu, 2015]. The application of ABM spreads through a wide range of areas and domains – social sciences [Axelrod, 1997], bioinformatics [Barnes & Chu, 2015], epidemics [Parker & Epstein, 2011], etc.

The typical structure of an ABM platform consists of three main elements [Macal & North, 2010]:

1. Autonomous agents with attributes and behavior.
2. Relationships and methods of interaction and communication.
3. Environment to interact with and within in addition to other agents.

Agents are supposed to be heterogeneous and active entities in pursuit of their internal goals, rather than returning passive responses (Figure 1). Relationships between agents are generally defined by distances and connectedness in an underlying space where agents are situated. The most used topologies are grids, Euclidean space, graph, Geographic Information System (GIS) and a spatial “Soup” model. Many ABMs include agents interacting in multiple topologies. Additionally the environment may contain constraints for the evolution of the agents like restricted amount of resources available to them, or restrains over the exact form and implementation of the topology (e.g. infrastructure, capacities of nodes, number of links allowed).

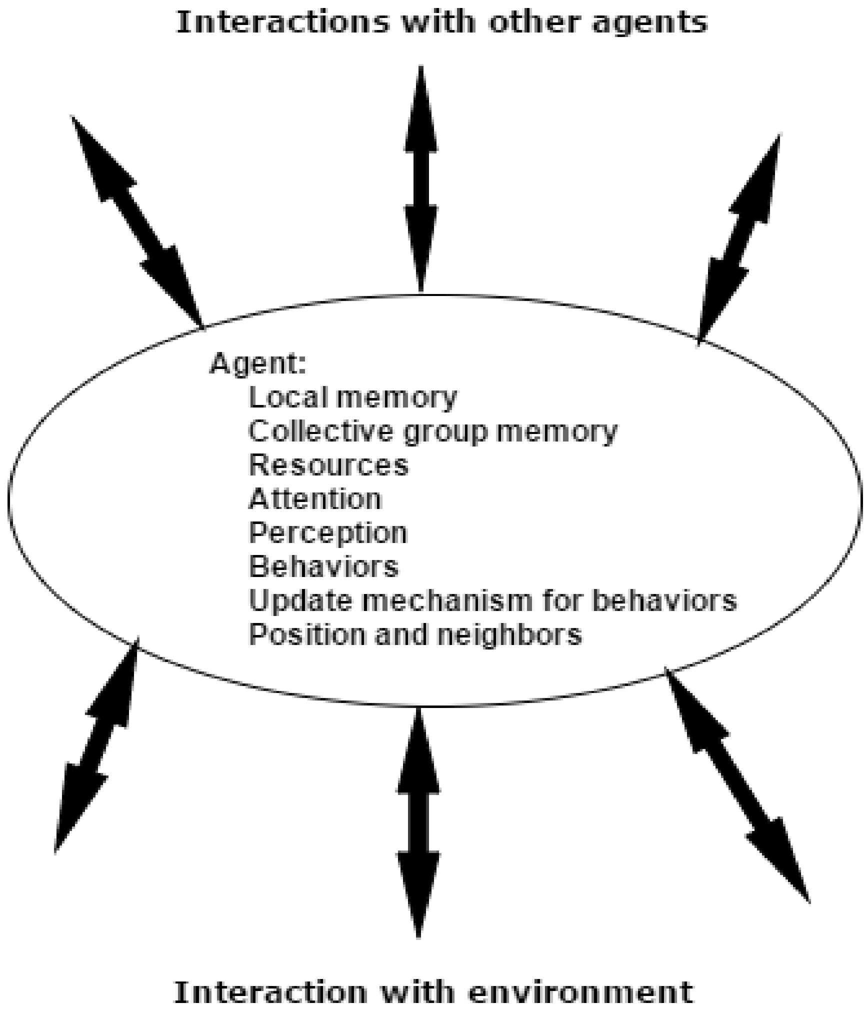

Figure 1: A general representation of an agent.

Optionally, agents can evolve within the environment via the standard evolutionary mechanisms like cross-over and mutation. This leads to additional complexity of the ABM and augment it with Evolutionary Computations (EC). This requires the need to additionally define rules for parent selection, mating, and mutation. These rules are not part of the internal agent model, but are imposed from outside. Therefore the system dynamics will not be the result only of the interaction rules defined by through the agents, which makes the models go outside the pure ABM.

To perform a simulation with ABM means to let agents repeatedly interact with their neighbors and behave within the environment. For more details and brief introduction see [Macal & North, 2010].

ABM Drawbacks

ABM simulations are a valuable research tool and add value and insights to the purely formal theoretical models of the real world. Several software toolkits have made the usage ABM relatively easy to be attractive for the scientific researchers. The usage of these toolkits allows researchers to concentrate on the modeling, as the infrastructure and reliability of the tools are created and optimized by specialists in the respective fields, mainly software developers. However they come with their drawbacks and limitations.

On one hand, the demands and formalism for creating such toolkits are not well established and can vary in their demands and complexity. Because of this, designers usually target specific use cases and problems, although trying to keep their approach as general and conceptual as possible, which leads to some limitations. The creation of new types of models that require different agent communication and interaction topologies, together with more sophisticated techniques for behavior modeling (e.g. artificial intelligence, deep learning), proved it to be not so simple and often require better programming skills and deep knowledge of the framework in order to be done. Often it turns out that a simulation can be done more easily from scratch rather than using an existing ABM platform. Another main issue is the computational cost in terms of time and memory needed to run large scale simulations. Moreover, if the behavior of the agents or/and the environment are probabilistic, several runs have to be carried out in order to have sufficient statistical confidence in the results [Barnes & Chu, 2015].

Additionally, more complicated and intricate agents and scenarios require more complex simulation models. The presence of multiple factors that need to be taken into account and the complexity of the resulting behavior to be analyzed afterwards increases the computational resource needed. Similar problems arise when the number of agents are very large, i.e. millions and billions [Parker & Epstein, 2011].

ABM Frameworks and Technologies

In terms of implementation the ABM can be regarded as a concept similar to the object-oriented programming (OOP). Similar to an object an agent can have internal state or memory and behavior. An agent, however, can undertake actions or learn and change its own state or the state of the environment while an object is more constraint.

Although related the Multi-Agent Simulations or Systems (MAS) and ABM are not the same. ABM tries to understand the emerging phenomena as a result of the behaviors of individual agents. Taking the examples with GT, given in the preceding sections, most of the game-theoretic models consider decision making at or near equilibrium, while ABM can consider complex and interesting behavior far from equilibrium. MAS focuses much more on solving practical and engineering problems [Parker & Epstein, 2011]. Another subset of objects that exhibits similarities to ABM is the actor-based modeling that is on top of the idea of messaging systems where interactions are based on immutable messages sent between actors. However this paradigm is motivated by multi-threading engineering problems and not by ABM although it seems very close to and useful for MAS [Suereth, 2012].

Some of the most used platforms are JADE[1], Repast[2] and Mason[3]. JADE is a Java-based framework designed to simplify the implementation of multi-agent systems. Although very powerful and well-constructed, it is designed according to the FIPA standards and does not allow too much freedom when designing a model. Because of this, it can be regarded not so much of an ABM framework but more like MAS software. However, if the model is in accord with FIPA the framework gives a stable and distributable environment that can handle use cases requiring excessive needs of computational power.

In terms of model definition Repast and Mason provide almost overlapping functionalities. Both allow the definition of multi-dimensional space where agents can live and interact. Cases where agents can be positioned into a subset of dimensions or topologies within an environment, and where agents can have different types of communications according to concrete dimension, may be tricky or even impossible to implement. Moreover, complicated scenarios require specific knowledge of the frameworks and software development skills. Evolutionary computations are supported mainly on intra-agent level through JGAP or ECJ respectively.

1:JADE: http://jade.tilab.com

2:Repast: http://repast.sourceforge.net

3:Mason: http://cs.gmu.edu/~eclab/projects/mason

In terms of large-scale simulation support there are two main solutions: parallelization and distribution of processes. Repast supports the first through another framework - Repast HPC, but this requires rewriting the model from Java to C++, and a good understanding of parallel operations' implementation. The distribution is offered through additional frameworks for distributed data like GridGain [1]that have a certain level of integration within Repast [Repast Symphony, 2006]. However their usage requires a good knowledge of them. Mason on its term provides multi-threading, but it is defined for advanced users [Luke, 2015] and require additional knowledge for multithreading and concurrency within the Java framework and more specifically the MASON implementation. MASON states [Luke, 2015] states that it can support millions of agents when no user interface is defined, but there are no marks on the memory consumption. The distribution of the simulation on cluster of servers is achievable through additional framework D-MASON [2]that may still require an adaptation from the user.

All three of JADE, MASON and Repast, as well as many other ABM frameworks, show a great level of advancement and provide a lot of functionalities, often overlapping, and allowing researchers to develop sophisticated simulations. Although they are relatively easy to use and therefore attractive for the scientific researchers, more complicated cases executed with parallelized runs on distributed machines, are difficult if not impossible to implement. They can require both integrated usage of several toolkits and additional software development knowledge and skills, shifting time and attention of researchers away from their domain problems.

CASPer Framework Overview

The above mentioned uses cases are not supported out-of-the-box from the currently most used platforms. Future research may also lead to the need to define new uses case that may also not be supported. Therefore a new architecture for ABM is proposed – Cognitive Agent-based Simulation Platform, briefly “CASPer”. In what follows an overview of functionalities is given. Then a general design and implementation details are presented. An example of two agents that are positioned in three types of environments is presented on Figure 2.

---

[1] GridGain: http://www.gridgain.com

[2] D-Mason: https://sites.google.com/site/distributedmason

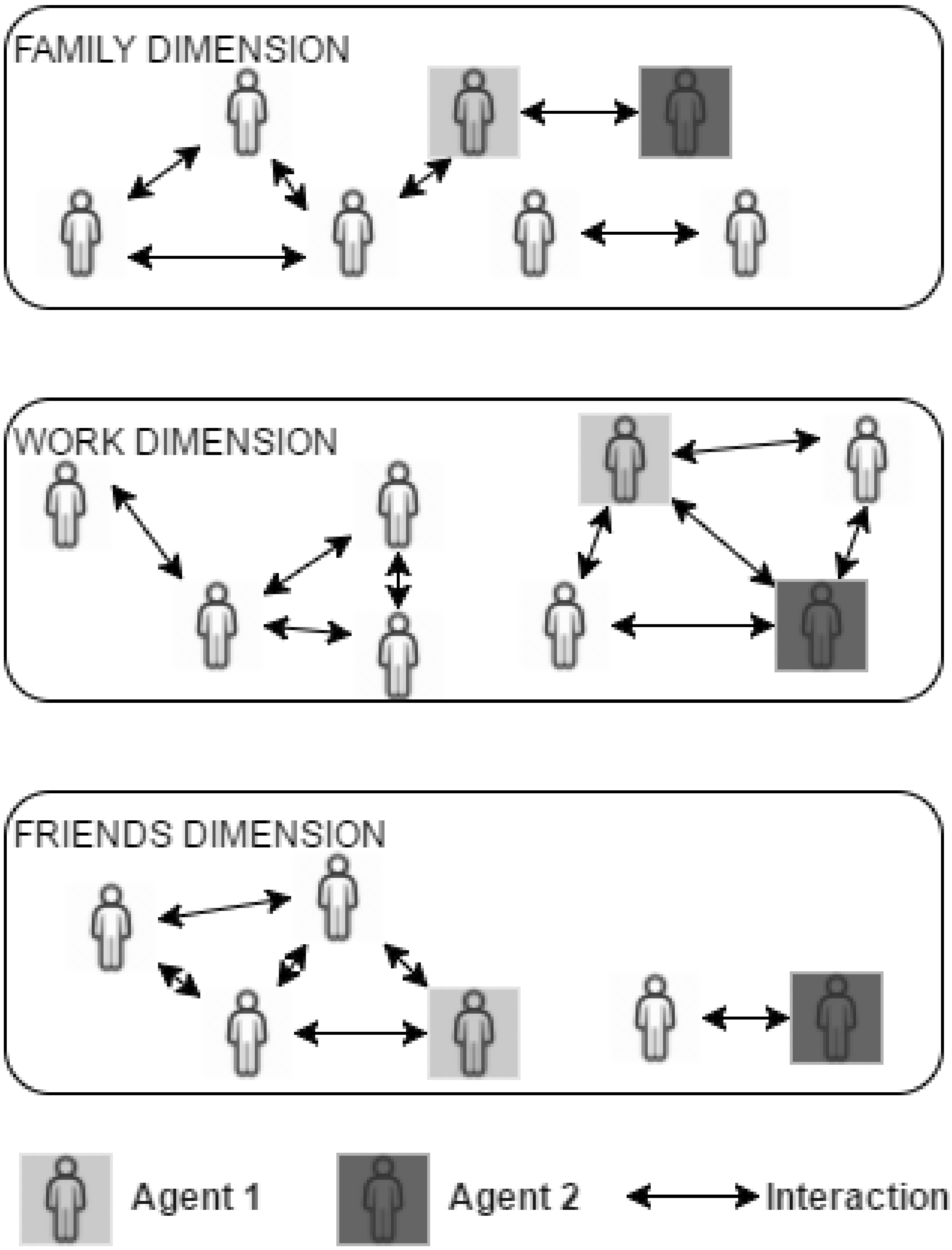

Figure 2: In the given example 2 agents that are positioned in 3 types of environments. In the FAMILY both agents are part of the same group. In the WORK they also make part of the same group, but in the FRIENDS dimension they are in separated groups. Each interaction represents a game played between agents, and in each environment there can be different type of games played.

On top of the basic ABM functionality presented above the CASPer framework aims to address the following use cases:

1. Agents can have access for read/write operations over a collective memory that is shared with other agents according to a grouping criteria.
2. Each agent is allowed to inhabit a subset or all of the dimensions available within an environment. Different agents may inhabit different subsets of dimensions.
3. Agents can be in multiple groups that are formed according to different criteria in one or more dimension. A simple example is to have an agent that is a member of a family (family group). In the same time he/she is an employee (work group), and keep close contacts with childhood friend (friend group). This agent can work with his sister, therefore in some of the groups he belongs to, he/she may see the already familiar agents.
4. Agents are supposed to have different types of communication within different groups. This means that the agent can play Prisoners' Dilemma at work group and Battle of the sexes in the family group.
5. Agents can have complex decision making models motivated by deep learning, neural nets, machine learning or artificial intelligence.
6. Also the framework is meant to support large-scale simulations. The aim here is to support millions and even billions of interacting agents in order to simulate the whole Earth human population or even more.

CASPer Implementation

In terms of usability recent trends in software development [Tulach, 2008; Martin, 2009] are oriented towards hiding implementation details from users and letting them add only their domain specific customization. This means hiding the multi-threading parallel execution on distributed machines from users and letting them implement only the logic of their domain specific model.

Following software design best practices [Reinhold, 2016] separation and encapsulation of different functionalities into modules, so that they can be easily removed or replaced, is favored instead of having one major component that rules everything.

The basic idea behind the framework consist of making a few interfaces available to users, where they can customize the simulation steps according to their specific needs. All the rest – parallel execution, distribution over machines and search algorithms for spaces is provided on the back side. Users should have no need to perform specific installations. To realize this functionalities are hidden behind interfaces. Moreover they are separated into different modules allowing an easy mechanism for replacement or updating these functionalities, as only a new implementation of the corresponding interface is required.

The basic outline of the framework consist of an engine of type ISimualtionEngine that takes care of the algorithm's execution steps. The other parts of the public API are:

An IStateSpace instance take care of the environment dimension, placing the agents inside, and perform the search and group actions. There are two implementation - DefaultStateSpace and HazelcatsStateSpace, but the interface is extensible and allows implementations with other dimensions and search engines (e.g. ElasticSearch, graph data bases etc.).

IEnvironmetDimension describes a particular dimension where agents can be positioned. The default implementations are for discrete, continuous and string representation. The interface implementation requires also a meaningful way to search within it, and a may require new IStateSpace and IRepositioning interfaces.

ICollectiveMemory is a data holder object that can be positioned inside one or more IEnvironmetDimensions. It allows read and write by IAgent that are in his range (i.e. belongs to the same group, society or island).

An IGame instance take care about the communication between agents. A default implementation gives the possible moves and the payoff matrix for them. The interface can be implemented to allow other types of communication not based on game theory.

IAgentMemory instance represents the memory of the agent. The instance can hold other IAgentMemory instances dedicated to concrete tasks. A subtype is IGameStrategyMemory that can hold the memory for specific type of IGame. It can be used in ICognitiveModel. IAgent instances represents an agent in the environment. It has an IAgentMemory, ICognitiveModel and access to ICollectiveMemory.

ICognitiveModel implementations provide the IAgent's behavior logic. It has access to IAgentMemory, ICollectiveMemorys that are visible to the IAgent. The implementation allows to have different behavior depending on who started the communication game. The implementation of ICognitiveModel can be easily made to access 3rd party libraries for machine learning or AI (Deep learning, neural networks, recurrent learning, etc.).

IFitnessFunction implementations gives the possibility to make an evaluation of the IAgent according to criteria. The function will be called when the IAgent has played all games within a particular cycle or epoch. The interface has access to IAgentMemory, ICollectiveMemorys and IEnvironemntalArtifact that are visible to the IAgent. The result is written in the memory of the agent and in the IStateSpaceContext and can be used for repositioning or the agents within the space later on.

IRepositioning implementation allows the optional repositioning of an IAgent instance. It has access to the IStateSpaceContext and the IAgent's memory. ITerminationCondition implementation states when the simulation can terminate. Examples are number of cycles or fitness gained by an IAgent. It has access to the IStateSpaceContext. Asynchrounous events can be sent on each step in order to monitor and save the simulation data for further analysis. Advanced users are allowed to modify easily all hidden parts of the implementation.

The parallel run is supported by an ISimulatoinEngine. Whenever a different approach is requested (e.g. using sequential single-thread implementation), the only things that need to provided is the respective implementation of the interface.

The distribution over a cluster of machines is provided by the Hazelcast [1]framework. It provides the support for the clustering, distributed data structures and locking mechanism for accessing and modifying objects. Similar to the parallelization there is an interface to provide access and locking to the data structure that can be re-implemented with other frameworks.

The different types of dimensions or topologies currently follow custom implementation, but can be extended to use sophisticated engines that provide better support for specific topologies like ElasticSearch[2], Neo4j[3], OrientDB[4], etc. The only thing that needs to be done is implementing the IEnvironmentDimension.

Conclusion

In the paper a novel framework for modeling social interaction has been presented with emphasis on the general approach and the first step of the implementation of a multi-agent environment. The general approach consists in combining artificial agents that instantiate cognitive models and a simplified social environment of experimental games like Prisoner's dilemma, Stag Hunt, Chicken, Ultimatum game, and other games used to model interesting social interactions. This simplification is regarded as a reasonable trade-off between the complexity of the expected emergent phenomena and the possibility to perform analyses using game theoretic approaches and social network theory.

[1] Hazelcast: http://hazelcast.com

[2] ElasticSearch: https://www.elastic.co

[3] Neo4j: https://neo4j.com

[4] OrientDB: http://orientdb.com/orientdb

The presented platform – CASPer – combines the complexity of the cognitive agents with the concept of a multi-dimensional agent space specified by the characteristics and history of the agents.

The incorporated constructs and mechanisms allow the agents to be efficiently monitored as part of more than one group with specific behavior and communication for each the interaction within each of the groups they may belong to permanently or temporarily. This is achieved by implementing an agent related and an environment related memories that can be used to model dynamically the state of the agents and the environment.

The framework takes care of the usage of multi-core machines and clusters of servers when available and needed. The cluster support will make possible large scale simulations with possibly billions of agents. The software design of the framework allows and promotes the usage of third party tools and modules for enhancement and customization of different parts of the simulations or adding new functionalities.

Acknowledgments

The paper is published with partial support by the ITHEA ISS (www.ithea.org), the ADUIS (www.aduis.com.ua), and the Central Fund for Strategic Development of the New Bulgarian University.

References

[Acevedo & Krueger, 2005] Acevedo, M., & Krueger, J. I. Evidential reasoning in the prisoner's dilemma. The American Journal of Psychology, 431–457., 2005.

[Adami et al., 2015] Adami, C., Schossau, J., & Hintze, A. Evolutionary Game Theory using Agent-based Methods. arXiv Preprint arXiv:1404.0994., 2015.

[Axelrod, 1984] Axelrod, R. The Evolution of Cooperation. New York: Basic Books, 1984.

[Axelrod & Hamilton, 1981] Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. Science, 211(4489), 1390–1396., 1981.

[Axelrod, 1997] Axelrod, R. The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration. Princeton University Press, 1997.

[Barnes, 2015] Barnes, Chu. Guide to simulation and modeling for biosciences, 2nd edition, Springer-Verlag London 2010, 2015.

[Binmore, 2007] Binmore, K. G. Playing for real: a text on game theory. Oxford ; New York: Oxford University Press, 2007.

[Camerer, 2003] Camerer, C. Behavioral game theory: experiments on strategic interaction, 2003.

[Camerer, Ho, & Chong, 2002] Camerer, C. F., Ho, T.-H., & Chong, J.-K. Sophisticated Experience-Weighted Attraction Learning and Strategic Teaching in Repeated Games. Journal of Economic Theory, 104(1), 137–188, 2002.

[Camerer, Ho & Chong, 2004] Camerer, C. F., Ho, T.-H., & Chong, J.-K. A cognitive hierarchy model of games. The Quarterly Journal of Economics, 861–898., 2004.

[Chater, 2015] Chater, N. Can cognitive science create a cognitive economics? Cognition, 135, 52–55., 2015.

[Colman, 1995] Colman, A. M. Game theory and its applications in the social and biological sciences (2nd edition). Oxford: Butterworth-Heinemann Ltd., 1995.

[Colman et al., 2014] Colman, A. M., Pulford, B. D., & Lawrence, C. L. Explaining strategic coordination: Cognitive hierarchy theory, strong Stackelberg reasoning, and team reasoning. Decision, 1(1), 35–58., 2014.

[Erev & Roth, 1999] Erev, I., & Roth, A. On the role of reinforcement learning in experimental games: the cognitive game-theoretic approach. In D. Budescue, I. I. Erev, & R. Zwick (Eds.), Games and human behavior: Essays in honor of Amnon Rapoport. Mahwah, NJ: Lawrence Erlbaum., 1999.

[Epstein, 2006] Epstein, J. M. Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton University Press, 2006.

[Epstein, 2014] Epstein, J. M. Agent_Zero: Toward Neurocognitive Foundations for Generative Social Science. Princeton University Press, 2014.

[Fehr & Fischbacher, 2004] Fehr, E., & Fischbacher, U. Social norms and human cooperation. Trends in Cognitive Sciences, 8(4), 185–190., 2004.

[Fehr & Schmidt, 2006] Fehr, E., & Schmidt, K. M. The economics of fairness, reciprocity and altruism–experimental evidence and new theories. Handbook of the Economics of Giving, Altruism and Reciprocity, 1, 615–691, 2006.

[Fiske, 1992] Fiske, A. The Four Elementary Forms of Sociality: Framework for a Unified Theory of Social Relations. Psychological Review, 99(4), 689–723., 1992.

[Fiske & Haslam, 1996] Fiske, A., & Haslam, N. Social Cognition is Thinking about Relationships. Current Directions in Psychological Science, 5(5), 143–148, 1996.

[Fiske, 2012] Fiske, A. P. Metarelational models: Configurations of social relationships. European Journal of Social Psychology, 42(1), 2–18., 2012.

[Geanakoplos, Pearce & Stacchetti, 1989] Geanakoplos, J., Pearce, D., & Stacchetti, E. Psychological games and sequential rationality. Games and Economic Behavior, 1(1), 60–79., 1989.

[Gigerenzer & Goldstein, 1996] Gigerenzer, G., & Goldstein, D. G. Reasoning the fast and frugal way: models of bounded rationality. Psychological Review, 103(4), 650, 1996.

[Gintis, 2009] Gintis, H. Game theory evolving: a problem-centered introduction to modeling strategic interaction (2nd ed.). Princeton: Princeton University Press, 2009.

[Grinberg, 2011] Grinberg, M. A cognitive approach to web-based intelligent agents: The TRIPLE architecture. Web Intelligence and Agent Systems, 9(1), 69–79, 2011.

[Grinberg, Hristova & Borisova, 2012] Grinberg, M., Hristova, E., & Borisova, M. Cooperation in Prisoner’s Dilemma Game: Influence of Social Relations. Proceedings of CogSci, 2012.

[Grinberg, Hristova & Lalev, 2010] Grinberg, M., Hristova, E., & Lalev, E. Models for cooperative decisions in prisoner’s dilemma. Advances in Cognitive Systems. IET: London, 2010.

[Grinberg & Lalev, 2008] Grinberg, M., & Lalev, E. Anticipation in Coordination. In The Challenge of Anticipation (pp. 215–236). Springer, 2008.

[Hristova et al., (2013)] Hristova, E., Grinberg, M., Georgieva, I., & Borisova, M. Cooperation in Prisoner’s Dilemma Game: Influence of Players’ Social Roles. In Proceedings of the 35th Annual Meeting of the Cognitive Science Society (pp. 2578–2583), 2013.

[Krueger, 2014] Krueger, J. I. Heuristic game theory. Decision, 1(1), 59–61, 2014.

[Luke, 2015] Luke, S. Multiagent Simulation and the MASON Library, 2015.

[Macal & North, 2010] Macal, C.M., North, M.J. Tutorial on agent-based modelling and simulation. Journal of Simulation 4, 151–162, 2010.

[Macy & Flache, 2002] Macy, M. W., & Flache, A. Learning Dynamics in Social Dilemmas. PNAS, (99), 7229–7236, 2002.

[Nowak, 2006] Nowak, M.A. Evolutionary dynamics _ exploring the equations of life. Belknap Press of Harvard University Press, 2006.

[Martin, 2009] Martin, R. Clean Code. Pearson Editions, Boston, 2009.

[Maynard Smith, 1982] Maynard Smith, J. Evolution and the theory of games. Cambridge ; New York: Cambridge University Press, 1982.

[Niazi, & Hussain, 2011] Niazi, M., & Hussain, A. Agent-based Computing from Multi-agent Systems to Agent-Based Models: A Visual Survey. Scientometrics (Springer) 89 (2): 479–499, 2011.

[Parker & Epstein, 2011] Parker, J., & Epstein. J.M. A Distributed Platform for Global-Scale Agent-Based Models of Disease Transmission. ACM Transactions on Modeling and Computer Simulation. Vol. 22, Issue 1, 2011.

[Reinhold, 2016] Reinhold, M. The State of the Module System, 2016.

[Repast Symphony, 2006] Repast Symphony. Repast Reference, 2006. Retrieved from: http://repast.sourceforge.net/docs/RepastReference.pdf.

[Shoham & Leyton-Brown, 2008] Shoham, Y., & Leyton-Brown, K. Multiagent systems: Algorithmic, game-theoretic, and logical foundations. Cambridge University Press, 2008.

[Suereth, 2012] Suereth, J. Scala in Depth. Mining publications, New York, 2012.

[Sun, 2006] Sun, R. (Ed.). Cognition and Multi-agent Interaction: From Cognitive Modelling to Social Simulation. Cambridge University Press, 2006.

[Taiji & Ikegami, 1999] Taiji, M., & Ikegami, T. Dynamics of internal models in game players. Physica D: Nonlinear Phenomena, 134(2), 253–266, 1999.

[Tulach, 2008] Tulach, J. Practical API Design. Apress, New York, 2008.

[Van Lange & Rusbult, 2011] Van Lange, P. A., & Rusbult, C. E. Interdependence theory. Handbook of Theories of Social Psychology, 2, 251–272, 2011.

[Van Lange et al., 2013] Van Lange, P., Balliet, D. P., Parks, C. D., & Van Vugt, M. Social dilemmas: Understanding human cooperation. Oxford University Press, 2013.

[Von Neumann & Morgenstern, 2007] Von Neumann, J., & Morgenstern, O. Theory of Games and Economic Behaviour (60th Anniversary Commemorative edition). Princeton, NJ: Princeton University Press, 2007.

[West, Lebiere & Bothell, 2006] West, R. L., Lebiere, C., & Bothell, D. J. Cognitive architectures, game playing, and human evolution. Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation, 103–123, 2006.

Authors' Information

Maurice Grinberg *– Research Center for Cognitive Science, Department of Cognitive Science and Psychology, New Bulgarian University, 1618 Sofia, Bulgaria; e-mail: mgrinberg@nbu.bg*

Major Fields of Scientific Research*: decision making, cognitive modeling, experimental game theory.*

Emiliyan Todorov, e-mail: *emo_todorov@mail.bg*

Major Fields of Scientific Research*: data analysis, decision making.*