ETHOS: A MAS FRAMEWORK FOR MODELLING HUMAN SOCIAL BEHAVIOR AND CULTURE

Jorge Simão[1] and Luís Moniz Pereira[2]
Centro de Inteligência Artificial – CENTRIA
Faculdade de Ciências e Tecnologia – Universidade Nova de Lisboa
2829 - 516 Caparica, Portugal
E-mail:[1] **jsimao@di.fct.unl.pt**; [2] **lmp@di.fct.unl.pt**

KEYWORDS

Human social behavior, culture, agent-based modelling, MAS simulation framework.

ABSTRACT

We present a framework for a Multi-Agent System (MAS) devised to support the modelling and simulation of agent-based models of human social behavior and culture change. We set forth its main abstractions, and test the usefulness and generality of the framework by describing how two previously published models from the literature have been re-implemented in it. We argue that our framework provides features that simplify the modelling process of a wide range of models of human social behavior, beyond what current MAS accomplish.

INTRODUCTION

Human social behavior and culture change are amongst the most complex phenomena studied by science. This results from the large number of interacting entities within a society, the different neuropsychological mechanisms underlying human social behavior, the multiple channels of information inheritance, amongst other factors. All of these contribute to create non-trivial dynamics in inter-personal relationships, social structure, and collective beliefs and values, which require careful analysis. This has challenged many researchers to propose unified theoretical frameworks, towards a social science based on a naturalistic interpretation of human behavior and culture (Durham, 1990; Boyd & Richerson, 1985; Laland, Odling-Smee, & Feldman, 2000; Barkow, Cosmides, & Tooby, 1992; Sperber, 1996). Although an agreed language of social phenomena is not yet available, the emerged consensus is that to understand human societies special treatments are required (Gilbert, 2000; Deacon, 1998; Donald, 1993; Mithen, 1996). (This, non withstanding the commonality to studies of nonhumans species.)

In spite of this widespread awareness, computational tools employed to support the modelling and simulation of systems with many interacting elements often lack features that permit capturing human specifics in a simple manner. For example, frameworks like SWARM, REPAST, and ASCAPE (Minar, Burkhart, Langton, & Askenazi, 1996; Collier, 2002; Brookings, 2000), while largely convergent on the set of tools provided, do not have any specific flexible mechanisms to simplify the modelling of human social learning (e.g. observational learning (Bandura, 1985; Boyd & Richerson, 1985)), social relationships dynamics (Nowak & Vallacher, 1998), dissemination of values (Durham, 1990), emotional contagion (Doran, 2000), or non trivial behavioral control (Bryson, 2000). This has the effect of making many interesting models hard to program, thus wasting away many advantages of using computational models as a complement to analytic mathematical ones. Namely, fast prototyping and analysis, and relaxation of assumptions made mostly for mathematical tractability (e.g. focusing on equilibrium states, and the use of infinite populations sizes).

To address these unique characteristics, we have developed a new conceptual framework and an object-oriented implementation of a Multi-Agent System (MAS). This framework, ETHOS, extends the traditional features in MAS for agent-based modelling, with new abstractions specially designed to model human behavior and culture. These include the transmission of information between agents through observation of performed behavior and direct generation of social stimulus, management of agents' social relationships, and support for flexible behavior selection mechanisms. This allows social science researchers to start model design with higher-level building-blocks than possible with current MAS, thus simplifying their work and fasten the development cycle.

This article presents the ETHOS framework, and is organized as follows: First we describe its main abstractions in terms of a class structure implemented in the JAVA language. Next, are presented two application case studies, intended to demonstrate the usefulness and generality of the framework features. The last sections compare ETHOS with other MAS, describe ongoing and future work, and present our conclusions.

FRAMEWORK OVERVIEW

The ETHOS framework provides as basic building blocks the kind of entities a modeler is likely to consider when thinking intuitively about human social behavior and culture. This includes objects describing the structure and topology of physical spaces, physical entities placed in this space (such as resources and agents with varying attributes and genetic makeup), different kinds of social relationship amongst agents, behavior selection mechanisms, and mechanisms for the social influencing of agents' mental states. In figure 1, we depict some of these, and in subsequent sections we describe in some detail ETHOS's main abstractions and how they fit together to create social simulation models. The object labels in fig. 1 correspond to object classes in the presentation below. The relationships between the main classes is shown in fig. 2.

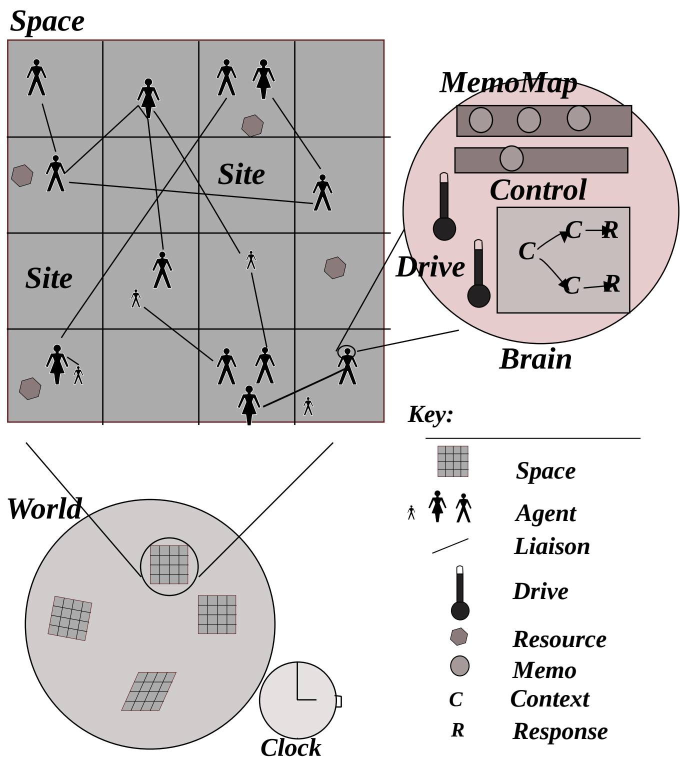

Figure 1: ETHOS's Framework Main Abstractions

Modelling the Physical and Social Environment

The top level abstraction of ETHOS is a World object containing a set of one or more physical spaces (Space object), whose events are timed by a universal clock. Each such Space object consists of a topological arrangement of Site objects, each of which is a place-holder for a set of physical bodies (Body object). Sites define a neighborhood relationship along with other sites in the *same* space, the details of which depend on selectable aspects, such as neighbor type (e.g. Moore or von Neumann) and space topology (e.g. 2D torus, frame, etc.). Sites can be asked for the list of neighboring sites (in a specified distance range) thus allowing simple navigation between sites.

The bodies in sites may be agents (Agent object), or other physical entities such as growing/consumable resources (Resource object). Agents can move from site to site, while other physical bodies have a fixed site location unless acted on by an agent. Agents live usually only in one space throughout their life-span, although they can migrate on demand between different spaces in the world. In addition to physical (site) neighbor relationships, agents may establish social relationships with other agents. Specifically, each agent maintains a list of social networks (SocialNet object), each corresponding to a different relationship type (e.g. parent-offspring, acquaintance, sexual, etc.). Each specific relationship is coded as a Liaison object, that holds information about a relationship, such as the agents involved, directionality, and its duration. Parent-offspring relationships are created and managed automatically by Ethos during agent creation, while other types of relationship are defined and managed by the simulation model. Different criteria can be used to specify which agents are added/removed from a particular social network of some agent. Agent selection in relationship management (and elsewhere) is made using Selector objects that specify particular criteria and modes of selection (e.g. non-competitive, where agents are selected by a local criteria, or competitive, where agents are selected by rank). If an agent migrates to a different space in the world, all its current relationships are deleted. This simplifies the distribution of a simulation world, as discussed below.

Agents' behavioral responses are triggered by stimulus generated by the physical or social environment, by having a Stimulus object associated with each environment element. For example, the Stimulus object associated with an Agent describes its previous behavioral responses and the agent's physical attributes. Agents can interact with other agents by observing each other's behavior (or attributes), by active transmission of information, or indirectly by inspecting and manipulating some part of the shared environment. For the purpose of behavior observation, we arrange for behavioral responses, and the set of stimulus that have triggered the responses to be made visible for observation by other agents. This is conducive to implement a simple abstraction of human psychological mechanisms used in imitation and shared attention (Tomasello, 1999/2001). The set of agents that a particular agent chooses to observe is usually a subset of the agents in its physical or social neighborhood, but it can be arbitrarily defined by the modeler. For example, if an Agent wishes to observe its physical neighbors it requests to the Space object where it lives for the set of agents living in neighboring sites. Active social transmission of information is obtained by having agents perform a behavioral response that sends Stimulus objects to target agents. This is useful to model social feedback of another's actions (Bandura, 1985), transfer of knowledge such as in an (active) teacher-learner type of roles (Boyd & Richerson, 1985), or to induce emotional states in other agents (Doran, 2000).

Population objects are used to aggregate agents and other bodies into collective units, whose event dispatching is coordinated (see below). Population objects can also have sub-populations has members, to allow hierarchical control and functional aggregation of many elements. A derived class AgentPopulation also provides miscellaneous operations specific to agents, such as mating of agents and generation of offspring according to specified assortment criteria (e.g. random or based on some agent's attribute), sexual recombination type (e.g. one sex or two sex based reproduction), and mating system (e.g. promiscuous, or monogamous (Alcock, 1997)).

Agents and Agents' Brains

Agents have a finite life-span, and may be dynamically created and eliminated during a simulation run, potentially leaving lasting effects in the space environment (e.g. born offspring, transmitted information, or modification of other agents' niche (Laland et al., 2000)). Agents have a genetic makeup (Genome object) which is randomly generated (if created without parents), or inherited from one or two parents. Each agent's genome contains a list of genes, whose number, data type, and initial value distribu-

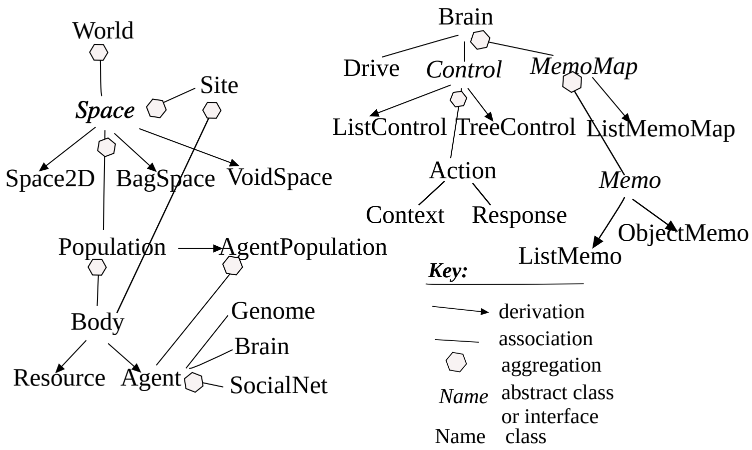

Figure 2: ETHOS's Class Hierarchy

tions (when not inherited) are selected by the modeler. The details of the genetic system, such as type of crossover and mutation intensities can be selected from the ones available or defined by the model designer (e.g. the top level Genome class implements a multi-point crossover, while the sub-class Genome1PCX implement one-point crossover (Goldberg, 1989)). The interpretation of gene's values is left to the modeler, but it is conceivable that in the future we include genes interpreted by ETHOS's runtime system (e.g. properties that modulate agents' behavior).

Agents have brains (Brain object) which process incoming stimuli and generate responses according to a behavior selection mechanism. Each brain may contain a set of different types of memory maps (MemoMap object), which specify the way in which incoming stimuli are integrated in the internal state of the Brain. (For simple models usually a single MemoMap is needed.) MemoMap objects contain sets of Memo objects which are conceptual assemble of memory state values competing for storage, activation, and valuation (preference) in the agent's brain. Thus, they abstract preference dimensions similarly to Durham's concept of *allomeme* (Durham, 1990). (We have opted for a different class name to avoid connotations with the word *meme*, which does not have yet a consensual interpretation (Aunger, 2000)). Derived subclass of MemoMap and Memo implement desired semantics. For example, the sub-class ObjectMemo can be used to represent an agent's favorite value for a certain cultural trait (e.g. clothe style, or political allegiance). The more complex sub-class ListMemo, on the other hand, stores the relative preferences of an agent's for a cultural trait and uses similarity effects in setting up preferences (e.g. an agent might prefer mostly a car model/brand such as "Ferrari", but like more a"Lotus" than a "Mini" because the former is more perceptually similar to the "Ferrari" than the later). In this case, a list of MemoItems objects is used, each of which stores information related to the associated stimulus.

The aggregation of Memo objects in a MemoMap, also allow for the preferences in one dimension two be affected by other due to (spatiotemporal) correlation of stimulus attributes (e.g. an arbitrary object can become a culture specific status symbol due to its recurrent association with people possessing status (Boyd & Richerson, 1985)). Such holistic valuation mechanism is implemented in ETHOS by having Stimulus object which have internal structure and having the MemoMap considering that structure in updating preferences. In general, this is useful to setup preferences by a processes of classic conditioning and assignment of emotional charged response to previously neutral stimuli (LeDoux, 1998). The intended behavior of MemoMap and Memo objects can be redefined by sub-classing any of the provided classes. Brains also contain drives (Drive object) which model agent's internal variables that change in time and that can influence behavior selection. This might be used in combination with memory elements with some activation latency to create complex temporal dynamics in agent's behavior.

A brain's Control object is used to enact agent's respostes trough a behavior/response selection mechanism. A Control manages a set of Action objects which specify a pair of a Context object, and a Response object. (An Action is very much like a production rule or a classifier in a classifier system (Goldberg, 1989)). The Context object represent a condition, usually upon the internal state and other attributes of the agent, and returns an activation level when requested. The default behavior is to compute the overall activation level from the individual activation levels of a set of internal drives and memory elements, but this can be overrided. The Response object specifies the procedure that should be performed case the action is selected for execution. The way actions are selected for execution depends of the particular sub-class of Control used. ListControl mantain a simple linear list of actions, and the actions selected for execution are the ones with maximum activity. The more complex sub-class TreeControl mantains a tree of actions to allow for the hierarchical "contextual parsing" of an agent's environment (Bryson, 2000). This is done by making some actions not having a response, but instead pointing to further actions, creating a tree of actions. Actions that are leaves of the tree are selected when all intermidiate actions up to the top one are have the maximum activation at their level. Thus, a context without associated responses is used as a condition for more fine-grained contexts. If a response is defined as requiring only a fraction of the available number of time units, several responses with high activation contexts can be jointly produced by the Control (e.g. social response feedback in the form of punishments and rewards can usually be assumed time thrifty). Responses have either internal or external effects. Internal effects are used to change only the agent's mental/brain state, while external effects are used to perform actions on the environment. Once the set of responses an agent's brain produces in a specific time step is computed, they are registered for execution and to be made visible to other agents.

Event Management and Behavioral Processes

ETHOS uses a simple discrete time step scheme to trigger events. The main framework objects define a well-specified order for event invocation in their step() method, which is executed in each simulation step (of a particular simulation run). The World object of a model invokes the step() method of each of its constitutive Space objects, which in turn do the same for each of its Population objects. Population objects relay control to its members to perform a *behavioral act*. The relay of control to objects down the structural hierarchy, is done at each level according to a set scheduling policy. For example, it can be specified whether the iterative sequence of objects lower in the hierarchy should be made random at each time step or fixed. (Scheduling of events at an arbitrary time step in the future is also provided in ETHOS using Scheduler objects.)

Each agent *behavior act* involves a sequence of four events or phases: inspect, feedback, respond, perform. The inspect phase is used to process the stimulus associated with elements in the physical and social environment an agent is attending to. The definition of which elements are to be attended is done in the special method inspectSetup(), that is invoked at the begining of the inspect phase. This method is overrided by the modeler and uses calls to toInspect(X) method, where X is the element to be inspected. Alternatively, calls to toInspect(X) can be performed in responses triggered by the agent's brain Control object. In this later case, the processing of the stimulus object is delayed until the inspect phase in the next time step. The stimulus are processed by the agent's Brain by relaying them to its MemoMap objects, which process them as described above. The feedback phase is similar to the inspect phase, and it is used to process stimulus previously generated by other agents and directed to the agent using the feedback(X) method, where X is the stimulus. For example, the feedback can be used to implement social rewards and punishments that might change the action selection mechanism of the agent's brain if learning by reinforcement is implemented in a subclass of Control (Goldberg, 1989).

In the respond phase, agents' brains are activated to select the responses to be performed based on stimuli received in the inspect and feedback phase, and the consequent brain state. At the end of this phase, generated internal responses perform changes in the brain's state (if any), and other automatic brain internal processes are also triggered (e.g. time passing effects on drives, and decay of memory activation levels). As a complement, the special method responseSetup() can be defined to setup additional responses to be performed. Finally, in the perform phase, the effects on the environment of the selected external responses are triggered. At the end of this phase, the external response(s) performed by an agent, and context stimuli is made visible to other agents. By default, the context stimuli include all those received by the agent in the current time step.

In aditional to other scheduling policies, an AgentPopulation object can be set to impose synchronous or asynchronous scheduling. In synchronous execution, the inspect and feedback phases are executed in all agents before moving on to the respond and perform phases. In asynchronous execution, each agent executes all phases sequentially after which control is moved to the next agent.

Other Features

Similar to most other MAS frameworks for agent-based modelling, ETHOS provides miscellaneous features for visualization of the simulation state, the gathering of statistics with simulation results using several types of objects such as binners and time-series, a "ready to use" Graphic User Interface (GUI) for parameter setting and simulation control, and the tailorable creation of the simulation objects. For most core classes requiring dynamic object creation, we use a corresponding object creator class to make user derived classes easy to integrate with ETHOS core classes. (Thus implementing an instance of the well-known *factory* object design pattern (Gamma, Helm, Johnson, & Vlissides, 1995).) Since these features do not differ significantly from the ones available in other MAS, we do not discuss them further. Worthy of mention, though, is the parallelization/distribution scheme of ETHOS which allows different simulation runs and/or parameter setting to be run concurrently in different remote server systems. This can be made either in batch mode or in full control by the GUI. The partition of a World object into a set of (quasi) independent spaces, also allows for distributed execution of events in distinct spaces. Because agent migration between spaces deletes all relationships, this makes the simulation semantics in the local and distributed case identical, without the complications of distributed object pointer semantics.

DEMO APPLICATION CASE STUDIES

To test the usefulness and generality of our framework, we have reimplemented several agent-based models from the literature with some form of non-trivial social interaction. Of particular interest are models of gene-culture dual inheritance, and inter-personal social dynamics, since they constitute the main application target of ETHOS. In the following sections, we describe how we implemented two such models, summarizing first the structure of the models (for the sake of completeness), followed by a short description of how the abstractions of our MAS framework were employed. Due to the wide varity and expressiveness of Ethos abstractions, specific models will typically use only a sub-set of the ones provided. In particular, the two models discussed here do not require complex action selection, and use worlds with a single space (no migration or distribution used).

Higgs's Mimetic Transition

Paul G. Higgs's mimetic transition model is an agent-based model of gene-meme co-evolution, and was designed to study the conditions under which the capacity for learning memes can evolve, even when there is no mechanism to make the distinction between memes which increase biological fitness and those that decrease it (Higgs, 2000). The model uses non-overlapping generations of agents which have two biological parents and a set of cultural parents from which they learn memes, with a probability proportional to their learning ability. Agents can also, with some small probability, invent new memes. Higgs's results show that, for a wide range of parameter values, the capacity for learning evolves in a phase-transition, where the capacity for learning evolves initially very slowly, but, after a critical point, increases very fast.

We model this by using agent brains with a single sub-class of the ListMemoMap object, whose elements are the individual memes coded in a class Meme derived from Memo. Since this model is very simple, we did not use a Control object for action selection. In stead, the responseSetup() method executes a single response representing the "performance" of all the memes an agent knows, and occasional generation of new memes. Agents are aggregated in a AgentPopulation being mated to generate an offspring population, according to a competitive Selector object. This is done in the step() method of the AgentPopulation object which is overwritten. The agents in the offspring population select as teachers a set of agents from their parents' generation, set in the inspectSetup() method, and observe their behavior as incoming stimulus. The sub-class of ListMemoMap detects the incoming stimuli coding for the memes, and, with a probability proportional to the agent's learning ability, store them in memory (if unknown beforehand). This is the default behavior of ListMemoMap, except that the defined sub-class overwrites the detect() method to make storage probabilistic. Once the AgentPopulation for parents as competed its job, it is replace by the offspring population.

Although Higgs's model does not require agents to discriminate and selected between competing memes, it would certainly be interesting to see how such ability influence model results. Given the above model setup, this extension could be easy done in ETHOS by replacing Meme objects with more complex Memo sub-classes, and having agents performing the associated responses more often than for less preferred memes. This is the kind of easy experimentation in model design that we look forward to test in Ethos.

Modelling Human Mate Choice

We also re-implemented a set of human mate choice models, developed by us in the past to study how human mating demographic patterns could be "grown" from the bottom-up, by using plausible behavior rules inspired by an evolutionary functional analysis of the task domain (Simão & Todd, 2002a, 2002b). Our models assume agents with a finite life-span, and a (normally distributed) one-dimensional quality attributed. An agent's goal is to try to mate with a high quality partner as early as possible in their life. Mating requires mutual acceptance, and involves a minimal courtship period during which agents may switch partner. Comparing several behavioral strategies, we found that the best types of strategies involve a combination of partner switching during courtship, and the setting of a minimal aspiration level threshold, to avoid mating with low quality partners (Simão & Todd, 2002a). Using these strategies provided an account of several population level patterns, such as reasonably high levels of assortative mating for most parameter settings, and the right-skewed bell distribution of age at marriage/mating time found in a wide range of cultures.

To model this with ETHOS, we defined three social networks for agents coded as SocialNet objects. One SocialNet represents the set of opposite sex agents a given agent is socially acquainted with. Another includes only one Liaison object used to represent the relationship with the agent being currently courted (this model imposes dating exclusivity). And a third SocialNet is used as temporary subset of the acquaintances an agent will interact with in a specific time step. Agents are added to the acquaintances SocialNet according to a Selector object that requires meet agents to be of the opposite sex. After the interaction SocialNet is initialized at each time step, the agent selects which members of the social interaction network the agent proposes to date, employing the specified strategy and depending on the quality of the agent's current dating partner (if any), and on the quality of the alternative agent acquaintances. The responseSetup() method is used to produce response specifying the set of members to propose to. In the step() method of the AgentPopulation that aggregates all agents, a procedure is executed to update the dating SocialNet objects of all agents, based on the set of dating proposals from all agents.

DISCUSSION

ETHOS currently adds three main modelling constructs to the ones provided by widely used MAS frameworks for agent-based modelling, such as SWARM (Minar et al., 1996), REPAST (Collier, 2002), and ASCAPE (Brookings, 2000). Namely, social transmission, relationship management, and behavior selection. The social transmission mechanism provides a principal manner for modelling inter-agent communication, without requiring ad hoc implementations by model developers. As different mechanisms with well documented semantics are provided, this will allow modelers to experiment with them and see in what scientifically relevant ways that affects simulations' results. For example, MemoMap and Memo object with new semantics, can be used to specify how social stimulus are recorded and valuated in memory. Social relationships are partly supported in these other MAS through generic graph like data structures, but Ethos takes this a step further. Giving a semantic interpretation to agents' social relationships, permits the automation of some processes, such as parent-offspring relationships management or the dynamics of social network growth. ETHOS is unique compared with these other MAS in providing integrated behavior selection mechanisms. While all such features can potentially be implemented on top of these other MAS, providing them "off the shelf" simplifies modelers' work. Finally, with respect to the event management scheme, we provide a structure than SWARM and REPAST, which is based on scheduling of event for arbitrary points in time, while not being so restricted as ASCAPE that imposes iteration of behavior rules at the grain of agent aggregates. In any case, we plan to bridge ETHOS's core features with other MAS because some users might prefer (to continue) to use them.

To keep testing the usefulness of ETHOS main abstractions (possibly extending and refining them), we plan to implement additional models of human social behavior, either from the literature or designed specifically for our future studies. Our design philosophy is that a feature should be incorporated in ETHOS only if it used by a wide range of models, and if its availability considerably simplifies model development and testing. Although scientifically useful models should be kept simple, it is our conviction that some types of models benefit from additional abstractions than that provided by current MAS. Still, further work is required to see which of ETHOS features are most useful. ETHOS is currently in prototyping stage, and we plan to deliver a publicly available version on the Web during the second trimester of 2003. ,

CONCLUSIONS

In this article, we described ETHOS, a MAS framework devised to support the development of agent-based models of human social and culture change. We described its main abstractions and how they fit together. To test the usefulness of the framework we described briefly how two models published in the literature have been re-implemented in ETHOS. We have argued that ETHOS's features simplify the modelling process of a wide range of agent-based models intended to study human social behavior.

ACKNOWLEDGMENTS

This work was partially supported by a PRAXIS XXI Ph.D. scholarship, and project FLUX, funded by FCT/MCES, Portugal. We would like to acknowledge the scientific and intellectual support and/or comments to earlier draft versions of this paper to Peter Todd, Paulo Gama Mota, Nuno Preguiça, and João Sousa.

REFERENCES

Alcock, J. (1997). *An introduction to behavioral ecology (sixth edition).* Sunderland, MA: Sinauer Associates, Inc.

Aunger, R. (Ed.). (2000). *Darwinizing culture: the status of memetics as a science.* Oxford University Press.

Bandura, A. (1985). *Social foundations of thought and action: A social cognitive theory.* N.J.: Prentice-Hall, Inc.

Barkow, J. H., Cosmides, L., & Tooby, J. (Eds.). (1992). *The adapted mind: Evolutionary psychology and the generation of culture.* New York: Oxford University Press.

Boyd, R., & Richerson, P. J. (1985). *Culture and the evolutionary process.* Chicago: University of Chicago.

Brookings, R. S. (2000). Ascape: An agent based modeling framework in java. (http://www.brook.edu/es/dynamics/models/ascape/)

Bryson, J. (2000). Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, *12*(2), 161–184.

Collier, N. (2002). Repast: An extensible framework for agent simulation. (http://repast.sourceforge.net/projects.html)

Deacon, T. W. (1998). *The symbolic species: The co-evolution of language and the brain.* New York: W.W. Norton & Company.

Donald, M. (1993). *Origins of the modern mind: Three stages in the evolution of culture and cognition.* New York: Harvard Univ Press.

Doran, J. E. (2000). Trajectories to complexity in artificial societies: Rationality, belief, and emotions. In T. A. Kohler & G. J. Gumermman (Eds.), *Dynamics in human and primate societies — agent-based modeling of social and spatial processes* (pp. 89–105). New York: Oxford University Press.

Durham, W. H. (1990). *Coevolution: Genes, culture, and human diversity.* Standford: Standford University Press.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object oriented software.* Addison-Wesley.

Gilbert, N. (2000). Modeling sociality: The view from europe. In T. A. Kohler & G. J. Gumermman (Eds.), *Dynamics in human and primate societies — agent-based modeling of social and spatial processes* (pp. 355–371). New York: Oxford University Press.

Goldberg, D. E. (1989). *Genetic algorithms: in search, optimization & machine learning.* Addison Wesley.

Higgs, P. G. (2000). The mimetic transition: a simulation study of the evolution of learning by imitation. *Proceedings of The Royal Society of London*(267), 1355–1361.

Laland, K. N., Odling-Smee, J., & Feldman, M. W. (2000). Niche construction, biological evolution, and cultural change. *Behavioral and Brain Sciences*, *23*, 131–175.

LeDoux, J. (1998). *The emotional brain: The mysterious underpinnings of emotional life.* Touchstone Books.

Minar, N., Burkhart, R., Langton, C., & Askenazi, M. (1996). The swarm simulation system: A toolkit for building multi-agent simulations. (http://www.swarm.org/index.html)

Mithen, S. (1996). *The prehistory of the mind: The cognitive origins of art, religion and science.* London: Thames and Hudson.

Nowak, A., & Vallacher, R. R. (1998). *Dynamical social psychology.* Guilford.

Simão, J. P., & Todd, P. M. (2002a). Modeling mate choice in monogamous mating systems with courtship. *Journal of Adaptive Behaviour*, *10*(2).

Simão, J. P., & Todd, P. M. (2002b). The self-organizing nature of mating systems. In *Preproceeding of the workshop on self-organization, evolution and social behavior.*

Sperber, D. (1996). *Explaining culture: A naturalistic approach.* Oxford: Blackwell.

Tomasello, M. (1999/2001). *The cultural origin of human cognition.* Harvard University Press.

AUTHOR BIOGRAPHY

JORGE SIMÃO completed his graduate and masters degree in computer science and engineering at the Faculty of Sciences and Technology, of the New University of Lisbon (FCT/UNL). He is currently completing is Ph.D. work on the topic of computer modelling and simulation of human social behavior at the Department of Computer Science. This research was carried out at the Center for Artificial Intelligence (CENTRIA), in association with the computer science department of FCT/UNL, and at the Center for Adaptive Behavior and Cognition of the Max Planck institute for Human Development, Berlin, and at the Anthropology Department of the University of Coimbra, Portugal. His research interests include studying aspects of human (social) behavior at the neuropsychological, behavior, and societal level, and connecting the three levels within a bio-computational and evolutionary framework. E-mail: **jsimao@di.fct.unl.pt**. HomePage: http://centria.di.fct.unl.pt/∼jsimao/.

LUÍS MONIZ PEREIRA is full professor since 1985 at the Department of Computer Science, Faculty of Sciences and Technology, of the New University of Lisbon (FCT/UNL), the Director of its Center for Artificial Intelligence (CENTRIA), and the Coordinator of its M.Sc. in Applied AI. His research interests include Knowledge Representation and Reasoning, Logic Programming, and Cognitive Science. E-mail: **lmp@di.fct.unl.pt**. HomePage: http://centria.di.fct.unl.pt/∼lmp/.