AAAI Technical Report SS-12-01
AI, The Fundamental Social Aggregation Challenge, and the Autonomy of Hybrid Agent Groups

Modeling the Effects of International Interventions with Nexus Network Learner

Deborah V. Duong
Agent Based Learning Systems
dduong@agentBasedLearningSystems.com

Abstract

Nexus Network Learner is an intelligent agent based simulation used to study Irregular Warfare (IW) in several major studies at the Department of Defense (DoD). Heterogeneous autonomous agents, each with their own separated inductive learning mechanism, have initial attributes and behaviors in proportion to demographic groups in the simulated population, and learn new behaviors as they serve culturally based goals. Nexus agents create a dynamic role-based network, and learn how to choose partners as well as what behaviors they should have with their network partners. As Nexus agents coevolve, nexus models the emergence of social institutions from individual behaviors, the fundamental social aggregation challenge. Nexus models the formation of learned vicious and virtuous cycles of behavior, some of which have higher average utility for the agents than others, and can be used to test the effects of interventions on the natural motivation-based system. An experiment is presented that uses Nexus to model the vicious cycle of corruption in an African country, from the first Irregular Warfare Analytical baseline at the Office of the Secretary of Defense (Messer 2009).

Nexus Network Learner:

A Symbolic Interactionist Simulation

Nexus Network Learner is part of the author's research program on Symbolic Interactionist Simulation, that started with the first Cognitive Agent Social Simulation, the Sociological Dynamical System Simulation (SDSS) (Duong 1991, Duong and Reilly 1995). Symbolic Interactionist Simulation addresses the fundamental social aggregation challenge with coevolving autonomous agents that maximize the utility of their actions based on their individual perceptions, acting solely in their own self-interest. From these individual actions emerge institutions, or expected behaviors that are more than the sum of their parts. Institutions in Symbolic Interactionist Simulations are dissipative (dynamic) structures: vicious or virtuous cycles of corresponding behaviors that symbolic interactionist agents develop as they adjust to each other. SDSS concentrates on vicious cycles such as racism, social class, and status symbols. The Symbolic Interactionist Simulation of Trade and Emergent Roles (SISTER) demonstrates Adam Smith's invisible hand in virtuous cycles such as price, a standard of trade (money), and a role-based division of labor (Smith 1994, Duong 1996, Duong and Grefenstette 2005). Symbolic Interactionist Simulations, as adaptive systems, model not only the emergence of social institutions from individual motivation, but the effect of institutions on individual motivation, a process known as immergence (Andrighetto et al. 2007) (See Figure 1).

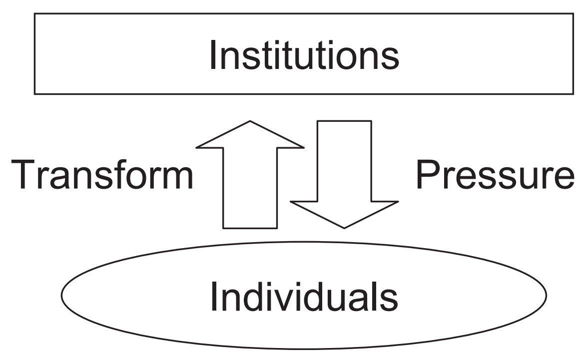

Figure 1. Emergence with Cross-scale Dynamics in Symbolic Interactionist Simulation

However, both of these Symbolic Interactionist Simulations are more theory based than simulations of real world scenarios. SDSS, SISTER, and most agent based simulation are general representations of processes, that don't exist in the same way in the real world. For example, SISTER agents can form a grid of stores that have some things in common with the real world, such as more expensive local convenience stores vice less expensive centrally located markets (Duong 2010). However, they are not the markets that actually exist on any street. Nexus was made to bridge the gap between theory based simulation and simulations that can anticipate real world outcomes, for the purpose of Irregular Warfare analysis at the Department of Defense.

Getting phenomena to emerge accurately from a simulation that is initialized to real world data is a difficult task. The social structures that form from individual agent motives are dissipative (dynamic) structures, and to anticipate what will happen to them with any accuracy, we need to recreate the cycles that derived the data in the first place. We need to reverse engineer the data by taking correlations in the data and explaining them with agent motivation causes. This is what Nexus was designed to do: every agent in Nexus has its own Bayesian Optimization Algorithm (BOA), an estimated distribution algorithm (EDA) that is very similar to a Genetic Algorithm, except that it can be initialized from and existing scenario and change from that initialization point. Nexus agents use the BOA to learn strategies to behave within roles and to choose role partners in a way that benefits their cultural goals. Through coevolution, agents initialized to behave in the expected distribution of behaviors in accord with their attributes and role come to behave in that distribution through motivated strategies, performing the desired "reverse engineering" from statistical correlation to cause. This capability is the most important contribution of Nexus Network Learner.

Nexus Network Learner Corruption Scenario

The Nexus Network Learner corruption scenario possesses a theoretical basis in interpretive social science, which is expressed in economics in the "New Institutional Economics" (NIE) theory (North 2005). NIE is the theory that institutions, which are social and legal norms and rules, underlie economic activity and constitute economic incentive structures. These institutions come from the efforts of individuals to understand their environment, so as to reduce their uncertainty, given their limited perception. However, when some uncertainties are reduced, others arise, causing economic change. To find the levers of social change, NIE would look at the actor's definition of their environment, and how this changes incentives and thus institutions. Interpretive social science is expressed in sociology in Symbolic Interactionism (SI), in which roles and role relations are learned, and created through the display and interpretation of signs. In Nexus Network Learner examples of roles and role relations are "Consumer" and "Vendor," and examples of signs are social markers such as gender and ethnicity. Symbolic Interaction provides the nuts and bolts for individual interaction in the model.

Nexus Network Learner is a model of corruption based on the theory that corruption is a result of globalization. Many social scientists assert that corruption is the result of conflict between the roles and role relations of the kin network and the bureaucratic network, two separate social structures with their own institutions forced together because of globalization (Smith 2007). Nexus Network Learner models the kin network and the bureaucratic network (in the context of the trade network or network of economic activity), as well as the role behaviors which result in corruption, and the capacity of individuals to learn new behaviors resulting in new institutions, based on their cultural motivations (Duong et al. 2010) (See Figure 2).

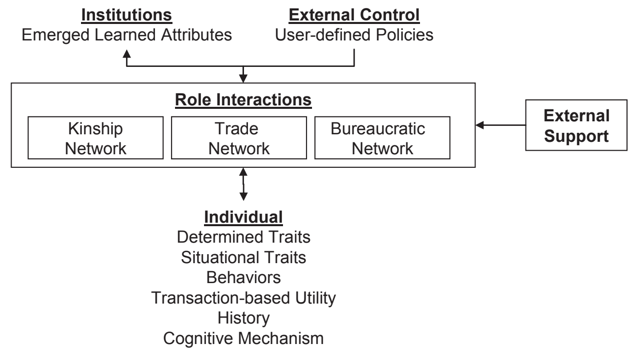

Figure 2. Nexus Network Learner Conceptual Model

In order to model institutions from first principles, a model of 'cognition' and 'agency' is needed, including 'cognitive agents' (or individuals) that perceive their environment and act based on their motivations. Institutions 'stabilize' when the individuals converge upon an agreement on how to act with each other, at a point when no agent can improve its situation by changing its behavior (Nash Equillibrium). Changes to institutions, such as the reform of corruption, may be studied by applying policy changes which change the motivations of the actors. New institutions form, perhaps repairing the problem, as individuals react not only to the policy changes but to each other as well.

Nexus Network Learner individuals learn to navigate their environment according to their individual incentives. As they learn they affect each other's incentives. As they change each other's incentives, the choices they make become new social structures (institutions). The choices include both "actions" and "perceptions". For example, individuals learn the 'type of persons' to include in their social network, including their kinship, ethnicity, and bribing behavior. Individuals learn whether to divert funds across networks through bribing and stealing. With incentives modeled, the effects of different government actions such as increased penalties for behaviors, foreign aid, or actions which affect the price of natural resources can be objectively evaluated. Changes in the habits of individuals change the prevalence and types of corruption as they evolve. These changes are driven by evolving incentives from government actions, and individuals' reactions to those actions and to each other.

Computational Model

Nexus Network Learner was created with the REPAST Simphony agent based simulation (North and Macal 2007). It was used in the first large study of Irregular Warfare at the Office of the Secretary of Defense (OSD), the Africa Study. In this study, the effects of international interventions on corruption were examined.

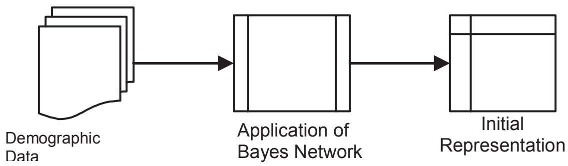

Figure 3. Model Initialization Process

Model Initialization

The Nexus Network Learner uses a Bayesian Network to characterize the demographic data of a country, which generates the initial agents of the scenario. The Bayesian Network describes characteristics that agents cannot change, for example, social markers such as ethnicity or gender. It also describes other characteristics that agents can change on an individual basis during the simulation, for example, behavioral characteristics, such as bribing or stealing, or preferences for choices of others in social networks (based on social markers or behavioral characteristics). Finally, the Bayesian Network describes demographic characteristics which individual agents do not learn, but are rather the output of the computations made during the simulation, such as unemployment statistics.(See Figure 3).

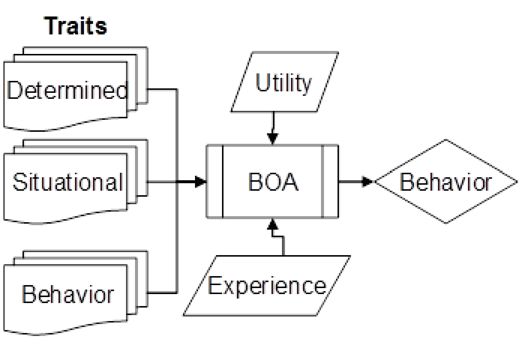

Figure 4. Agent Components

Agents

In Nexus Network Learner, agents are adaptive: that is, they are able to learn and adapt to new role behaviors through the use of evolutionary computation techniques of artificial intelligence, also known as genetic algorithms. Agents start out with different propensities to behave that are generated by a Bayesian Network, but they learn other behaviors based on utility, that is, what they care about. Utility is in the trade interactions (transaction-based utility) of the kin that an agent cares about. As the simulation runs, agents learn how to navigate their environment according to their individual traits and experience through a Bayesian optimization algorithm (BOA). These choices include both actions and perceptions. Agents learn the type of persons to include in their social network, including kinship, ethnicity, and bribing behavior. They also learn whether to divert funds across networks through bribing and stealing. With incentives modeled, the effects of different whole of government actions such as increased penalties for behaviors, foreign aid, or actions which affect the price of natural resources may be studied. Corruption behavior changes through synchronous alterations in the habits of individuals, driven by new incentive structures that come both from whole of government actions, in agent's reactions to those actions, and to each other. (See Figure 4).

Role and Role Relations

There are sixty four different roles in the three networks. Typical Kin roles include Father, paternal grandmother, maternal cousin. Typical Trade roles are retailer and vendor. Typical Bureaucratic roles include government employee and government employer. There are eight types of corruption relations possible:

1. Stealing/Trade Network (Scam)
2. Bribing/ Trade Network (Gratuity)
3. Hiring Kin/ Trade Network (Nepotism)
4. Bribing to be hired/ Trade Network (Misappropriation)
5. Stealing/Government Network (Levy, Toll, Sidelining)
6. Bribing/Government Network (Unwarrented Payment)
7. Hiring Kin/Government Network (Nepotism)
8. Bribing to be hired/Government Network (Misappropriation)

Nexus Network Learner Algorithm Summary

Calibration/Baselining

The demographic and statistical data of a country is expressed in a Bayesian network. Some of those qualities are input, and some of the relations are present values for what it is that we are trying to explain. The variables we are trying to explain are used to calibrate the simulation in the beginning, but these variables will also be the computed output of the simulation. Of these variables that we are trying to explain, there are direct behaviors which are learned, that the agent has control over choosing, and there are demographic results, such as resultant employment or penalty, that may be higher order effects. So, of the Bayesian variables, some are designated as fixed, some are designated as learned, and some are designated as derived, in the input file.

Initialization and Attribute Access

At initialization, Nexus reads the fixed attributes from a Weka "arff" file (Witten 2000), pregenerated by a Bayesian Network. These are set per agent for the length of the run. There are also two kinds of are learned behaviors in the Bayesian Network: learned network choice behaviors and simple learned behaviors. During initialization, Nexus uses the same Bayesian network to generate the chromosomes for the initial BOA, that represent two-valued attributes (Booleans) using the initial Bayesian Network Conditional Probability Tables, in each agent. At a later stage, multiple value attributes and floating point for the learned network choice behaviors will be supported. Nexus uses the conditional probabilities of the learned network choice behaviors in the Bayesian net to make a preference function to rank network members, implementing the tendency to behave through ranking. The preference function is created from the current cycles BOA "chromosome." The network choice behaviors are learned through deviations from the conditional probabilities of the behaviors encoded on the chromosome. For now, the learned behavior increments the ranking by a fixed amount, but when floating point numbers are implemented that may be tuned. For learned behaviors that do not involve network choices, the chromosome encodes the behavior choice, from a population initialized to the original conditional probability table values of the Bayesian network. Which attributes are the fixed attributes, the learned network choice behaviors and the learned behaviors are designated in the input file. The derived (computed) attributes, accessed at calibration but not at initialization, are also listed in the input file.

The learned behaviors are :

Network Choice Behaviors: Employee.bribeEmployer, Employer.stealFromWorkplace, Employee.isKin, Employee.ethnicity, wife.ethnicity, vendor.ethnicity.

Simple Behaviors: bribeEmployer, acceptBribeEmployer, bribeForServices, acceptBribeForServices, stealFromCustomer

Many attributes in the Bayesian network have equivalents designated in the properties file, for example, Employee is the equivalent of GovernmentEmployee in the Bayesian Net. This is so not to have wasteful repetition in the Bayesian net, and so preexisting Bayesian nets may be combined with social role network descriptions.

Some examples of derived (computed) attributes, that are not kept track of but computed, given the agent and the point of view, on the fly, are: isKin, isProductive, stealFromWorkplace, and penalty are in the Bayesian net, isHead is in the properties and isCategory which finds out if an agent is in a derived role. stealFromWorkplace differs from a behavior in that we are measuring whether it happened, and someone in the perceivers social network saw it, rather than whether there is a tendency for it to happen. They have default values. There is a method from which all attributes of an agent are read, whether fixed, learned, or derived.

There is a method that finds the value of an attribute, that takes an observer. In some calculated methods, for hidden kinds of attributes, an observer is required so that social networks of the observer may be checked for knowledge of the action. Knowledge of behaviors is saved in every agent. For statistical purposes, there is a ground truth view that gives ground truth of the actual tendency of the observed (which is what we are looking for when we look for behaviors!) that is, his genotype, as well as the expression of the behavior in the world when given opportunity, the phenotype, that is, the ground truth statistical view of how many times a behavior actually occurred.

Cycle Initialization

At the beginning of a cycle, the next BOA chromosome is used. The BOA chromosome contains learned changes to conditional probabilities which are initially random, as well as binary behaviors, which were initially generated from the Bayesian network. Individual agents create the initial network from the learned network choice preference function changes. Then, initial funds (in the properties file) are pumped into the system. Funds are conserved, even if the utilities that depend on them are not. Exogenous funds (from the properties file, in this case resource funds and foreign aid.) are then pumped in on a regular basis, if there are any. As a future direction, funds are to be recycled until they reach equilibrium, with money in accounts pretty steadily in the absence of the exogenous input. After a few runs and equilibrium is reached, utility begins to be kept track of.

Network Creation

The properties file determines the properties of the active roles that choose a relationship. All those with active roles seek relationships. The number of the relationship is determined by normal random variate draws given the ranges in the properties file, and the agent tries to choose that many members. The Bayesian network, modified by the BOA results, is used to rank the population members for the networked relationship. The BOA allows each probability in the Conditional Probability Table of the network choice attributes to be incremented or not by an increment specified in the properties file. After the change, the CPT is normalized. There are distance thresholds associated with the role choice, that make them have to match by a certain amount or no partner will be chosen. Both active and passive agents have thresholds for initiating or accepting a partnership. Some of the attributes are calculated, like Employer.stealsFromWorkplace, and are calculated in a method for hidden attributes based on network knowledge, containing hashmaps of what agents know what.

Fund Cycles

Funds move through networks and are conserved. Once the funds reach the threshold level of the account they are moved into, they are distributed to network relations. The agent must already have the recipient relation. For example, in this case, the Head of Corporation receives corporate income, and when it reaches threshold, he becomes a corporate taxpayer and gives the money to the taxman, to put in the taxes account. However, he must already have found a taxman to give it to, which is just a government sector employee of a certain paygrade, who is his taxman. Then when threshold is reached on the taxes account, the taxman becomes a government receiver and gives government income to the head of government. The money in one account is usually distributed to several accounts.

Each account lists allocation percentages to other roles. There may be different allocation schemes conditional upon the traits of the agent as marked in the properties file. For example, a patrilocal home provider will distribute his support to different relatives than a matrilocal home provider. Or, a stealing or bribing Employer will distribute his employees paycheck account to his home account whereas an honest one would not. A penalized employer (which is decided) is not able to distribute any money to family, and for now sits out until a new chromosome (a new strategy of behaving) is being tested. Penalty is decided based on a random variate, but is more likely the more people know about it. In the future, the gods-eye-view of the penalized method will know how many turns the penalization occurs as well. Additionally, individual transactions are marked in the properties file with utility values that are also conditional on traits. For example, if a transaction occurs between two bribing, or includes a stealing agent, utility is reduced as compared to honest agents. Many of the attributes on which allocations are based are learned binary behavior attributes.

Learning

Every n clock ticks, every agent tries a new strategy, that is, it makes another chromosome from its evolutionary computation methodology active. Before it pulls in a new one, it judges the success of the first by adding up the utilities of all the dependant relations of the agent, the agent's utility. After all of the strategies have been tried (after all of the BOA chromosome preference/attribute lists have been used), selection occurs based on the comparison of utility. The n attribute lists with the best utility fitness values are used to learn a new Bayesian net using the Bayesian Optimization Algorithm, with naive learning (the k2 algorithm). The Bayesian net is then used to form m more BOA chromosome preference/attribute lists.

Output

For output, the Bayesian network may be learned with the original structure, and an arff file is created, that is readable in the Weka machine learning program (Witten 2000). It is learned from all fixed behaviors, all of the actual behaviors, and the actual penalties, using the gods-eye calculation method for these (See Figure 5).

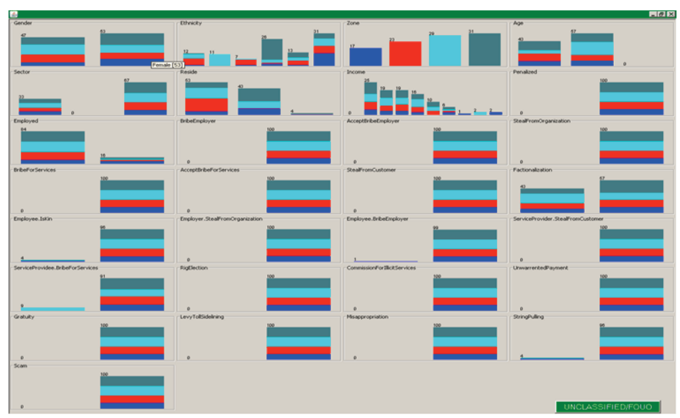

Figure 5. Sample output "arff" file in Weka, showing percentage of persons of each zone having corrupt and other attributes

Example Experiment

Run two different initialization scenarios, one in which the agents all engage in stealing, and another at the normal amounts of bribing and stealing for a particular African country. Observe the flexibility each has to improve, and any unexpected side effects from the stealing vs. the normal scenario. This experiment is taken from the Nexus Scenario Development Guide written for US Army Training and Doctrine Analysis Center (Duong and Pearman 2012).

Results:

- In the excessive stealing scenario, agents never learned to stop stealing. We also see evidence of agents in the excessive stealing scenario never trying to find service providers that did not steal from them. In contrast, agents in the 'normal' stealing scenario learned to find service providers that did not steal from them within two years.
- In general, agents in the excessive stealing scenario never used bribing to accomplish goals, rather electing to implement and acknowledge stealing as the preferred strategy. In contrast, agents in the 'normal' stealing scenario applied bribing techniques in the first two years, but after 15 years agents quit bribing role relations, specifically employers and service providers.
- In the excessive stealing scenario, employers did not accept bribes. In contrast, employers in the 'normal' stealing scenario accepted bribes during the first two years, but after 15 years employers generally stopped accepting bribes.
- After 15 years in both the excessive stealing scenario and the 'normal' stealing scenario, we see homogeneous responses across strategies. For instance, agents generally provided the same response for a particular parameter (say 'BribeForServices') after 15 years as opposed to more heterogeneous responses for the same parameter after two years.

General findings:

- Agents in the excessive stealing scenario find themselves in a vicious stealing cycle, unable to evolve into less corrupt patterns. This finding highlights the importance of initial conditions when developing the Nexus scenario.
- Agents in the 'normal' stealing scenario become less corrupt over time.
- Agents in both the excessive stealing scenario and the 'normal' stealing scenario become more consistent and predicable over time, as evidenced by homogeneous responses after 15 years.

Nexus as Adaptive Simulation

At present, the Nexus Network Learner prototype only adapts to initialization data from country demographic studies. However, it has the potential to adapt to online data and other simulations as the simulation progresses in time. Adapting to data is a future direction of Nexus Network Learner.

The Nexus Network learner uses coevolution to have individual agents adapt to the data, through their autonomous perceptions and motivations, so that the data that enters Nexus comes to be explained dynamically by Nexus. The input to Nexus are the demographic characteristics for an entire population, both physical characteristics and social categories, as well as behavioral traits that are based on these physical characteristics and social categories. This input is in the form of a Bayesian Network, which actually generates the agents based on the data. However, some of the variables of this data are designated by the user as "stuck" in place, such as the gender or ethnicity of the generated agent. Other of the simulation data is changeable, and is learned in a way that increases the agent's utility. In starting out with tendencies toward behaviors, and then changing them based on utility, a differentiation in the agents occurs, and the motivation structure come to explain the data on an individual agent basis, displaying immergence. Each agent varies its behavior according to the distribution of what is expected of the population of agents that have its attributes, but tends to niche itself onto part of that initialization distribution, applying selective pressure to other agents of its class to niche themselves to another part of that initialization distribution, depending on what is locally best for the agent. Because the niche the agents find is motivated by agent goals, these goals explain the distribution. For example, if 30% of the female Mongos offer bribes in the real world, then each of 100 female Mongo agents in Nexus Network Learner offer bribes 30% of the time in the beginning of the simulation run. Towards the end of the run, 30 of the female Mongo agents offer bribes 100 % of the time and 70% do not offer bribes 100% of the time.
Co-evolutionary selective pressure promotes this niching process, a process of coevolutionary data adaptation that is modeled after nature and explored in different ways in several of the author's works. To illustrate we present examples from nature and from other symbolic interactionist programs.

1. Coevolution in nature case:

Species have a relationship to each other in an ecosystem. Seed the ecosystem with a foreign species. The species directly affected adapt and then the other species adapt to their adaptations. An example of an adaptation would be a fast running predator would cause the prey to run fast, and then the predators of the real system again. So we get one ecosystem to mimic another by seeding it with the species of another.

2. SISTER case:

In the author's SISTER simulation, agents receive dispersed selective pressure from the environment to perform the behaviors that result in composite goods (like succotash) from the displaying and interpretation of the signs (like succotash-seller), with other agents who they are in a mutually beneficial relationship with. SISTER has shown that the succotash-selling agent develops a multidimensional role in the social space that other agents are pressured to have in its absence, a dynamic structure which is a role in the social system. A new agent introduced into the social system, without the recipe for succotash, would learn to make succotash if it displayed the sign "succotash seller" as lima bean sellers and corn sellers attempt to transact with the agent. The new agent receives selective pressure to adapt to the role of succotash seller, the role duties of which were decided through a cultural history of self serving transactions over time. In this case, a new "blank" agent is seeded with the society around it.

3. Nexus case:

In the Nexus simulation, we start out with a total seeding to the environment that agents are to adapt to, and expect the agents to develop the strategies that give them an advantage given the signs they display and their current mix of behaviors with the environment. It is the equivalent of the SISTER programs effect on the new agent that happens to display the sign, except that all agents are subject to the same data adaptation process simultaneously. In the SISTER program, a new agent that displays a sign receives offers from all the other agents to engage in transactions. The environment of behaviors prompts them to develop corresponding behaviors that serve its utility. We can think of the interface of signs from the old established agents to the new agent as the "outer data" – something that is measurable, but doesn't include the inner agent strategies (intentional reasons – the "inner data") for its behavior. The agents together find a way to serve their utility through their behavior, strategies which, when they interact together, make the vicious and virtuous cycles that caused the outer data in the first place.

In Nexus, the "outer data" is introduced to simulation during intialization, in the form of making agents behave certain ways, in proportions that are expected for a class. They have behaviors that they are forced to make without motivation, and behaviors that they are slated to start out with. They develop "explanations" of the behavior through adaptation: We start with what is, and it comes to be necessary, just as in the newly introduced agents of SISTER – they start with what is and adapt to what they have.

In SISTER the role, the set of behaviors that define the role, reproduces itself in expectations through the display of signs. It is the same in Nexus for agents of a demographic class. In Nexus, agents start out with the distributions of behavior that occur in a real world society according to a demographic class. Nexus agents gradually develop strategies to deal with each other based on the initial conditions in the demographic class data. If the right factors are captured, Nexus agents reproduce the data through their intentional strategies ("inner data").

Summary

Nexus Network Learner is a Symbolic Interactionist Program designed to anticipate real world outcomes in social simulations. It models basic principles of interpretive social science: in sociology, roles and role relationships that develop in the process of symbolic interaction and in economics, individual self interested transactions that form corresponding trade plans. Institutions emerge and at the same time affect the motivations of individual agents. Coevolution fuels this engine of society, as well as enabling Nexus Network Learner to adapt to and explain data, "reverse engineering" the correlative data relations into the causes and motivations that underlie it. Nexus Network Learner has been used in several major DoD studies of Irregular Warfare, including the modeling of the institution of corruption.

References

Andrighetto, G., Campennì, M., Conte, R., Paolucci, M. 2007. On the Immergence of Norms: a
Normative Agent Architecture. In Proceedings of AAAI Symposium, Social and Organizational
Aspects of Intelligence, Washington DC.

Duong D. and Pearman, G. 2012. Nexus Scenario Development Guide, US Army Training and Doctrine Analysis Center. http://www.scs.gmu.edu/~dduong/NexusTechnicalReport.pdf .

Duong, D.V, Turner R. and Selke K. 2010. "Crime and Corruption," ESTIMATING IMPACT: A Handbook of Computational Methods and Models for Anticipating Economic, Social, Political and Security Effects in International Intervention, Kott and Citrenbaum, eds. New York: Springer Verlag. http://www.scs.gmu.edu/~dduong/Estimating_Impact_Corruption .pdf

Duong, D. V. 2010. “Autonomous Tags: Language as Generative of Culture” Second World Conference on Social Simulation, Takadama, Keiki; Revilla, Claudio Cioffi; Deffuant, Guillaume, eds. New York: Springer Verlag. http://www.scs.gmu.edu/~dduong/WCSS08.pdf

Duong, D.V. and Grefenstette, J. 2005. “SISTER: A Symbolic Interactionist Simulation of Trade and Emergent Roles”. Journal of Artificial Societies and Social Simulation, January. http://jasss.soc.surrey.ac.uk/8/1/1.html.

Duong, D. V. 1996.“Symbolic Interactionist Modeling: The Coevolution of Symbols and Institutions”. Reprinted in Gilbert, ed. Computational Social Science, London: Sage Publications, 2010. http://www.scs.gmu.edu/~dduong/symbolicInteractionistModeling.pdf.

Duong, D.V. and Kevin D. Reilly. 1995. “A System of IAC Neural Networks as the Basis for Self-Organization in a Sociological Dynamical System Simulation”. Behavioral Science,40,4,275-303.
http://www.scs.gmu.edu/~dduong/behavioralScience.pdf.

Duong, D. V. 1991. A System of IAC Neural Networks as the Basis for Self-Organization in a Sociological Dynamical System Simulation. Master’s Thesis, The University of Alabama at Birmingham.
http://www.scs.gmu.edu/~dduong/behavioralScience.pdf.

Messer, K. 2009. The Africa Study. HSCB Focus 2010 Conference.

North, D. 2005. Understanding the Process of Economic Change. Princeton: Princeton University Press.

North, M. and Macal, C. 2007. Managing Business Complexity: Discovering Strategic Solutions with Agent Based Modeling and Simulation. New York: Oxford University Press.

Smith, A. 1994. The Wealth of Nations. New York: The Modern Library.

Smith, D. 2007. A Culture of Corruption. Princeton: Princeton University Press.

Witten, I., Frank, E. 2000. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. NewYork: Morgan Kaufman.