AN INFERENTIAL APPROACH FOR VALIDATING AGENT SIMULATIONS

A. YAHJA*, National Center for Supercomputing Center, Urbana, IL
K. M. CARLEY, Carnegie Mellon University, Pittsburgh, PA

ABSTRACT

As the size and complexity of the agent-based simulation models increases so does the time and resources needed to validate the model. Validation is critical for replication of simulation results, which is a basis for scientific advance. Automated and semi-automated tools are needed to support validation activities and so reduce the time and number of personnel needed.

A tool called WIZER (What-If Analyzer) which embodies our inferential approach is implemented. WIZER consists of four parts: an Alert WIZER, an Inference Engine, a Simulation Knowledge Space module, and an Empirical/Domain Knowledge Space module. The Alert WIZER characterizes simulation data with assistance from the statistical tools it semantically controls, compares simulation data to empirical data, and produces semantic descriptions of both the data and the comparison. The Inference Engine performs both causal and "if-then" rule inferences.

WIZER is run on a simulator called BioWar which models disease spread in a demographically-representative city population. The results show that WIZER validates in a clear and automated manner the simulation models for the relative timing of peaks of influenza incidence and school absenteeism. They indicate that the inferential approach underlying WIZER can increase the transparency and reduce the time for model validation.

Keywords: simulation validation, semantics, knowledge systems, causality, virtual experiments

---

* *Corresponding author address*: Alex Yahja, 4038 NCSA Building, National Center for Supercomputing Applications, 1205 West Clark St., Urbana, IL 61801; e-mail: alexy@uiuc.edu

INTRODUCTION

Computer modeling and simulation provides a means of understanding and predicting the behavior of real-world systems based on knowledge of basic laws, empirical findings, and assumptions. It complements theory and experimentation/observation as the third pillar of science. Simulations can be viewed as virtual experiments. Computing advances mean better simulation models can be built. Typically, however, simulation results are designed solely for human analysis and validation is provided by subject matter experts judging that the model "feels right," possibly after preprocessing the results using statistical tools. This process is time-consuming and deficient in clarity, transparency and objectivity. The remedy is usually prescribed in the form of a methodological approach (Yilmaz 2006), of which Verification, Validation and Accreditation (VV&A) process is one, but validation remains a cumbersome process.

NASA lost the Mars Climate Orbiter spacecraft on September 23, 1999. Mission specifications called for using metric units, but the Lockheed Martin group sent navigation information in English units. The mix-up meant that Lockheed Martin engineers modeled navigation with pounds force (the English unit for measuring thruster impulse) while JPL did their calculations in newtons (the metric measurement). One pound force is equivalent to 4.45 newtons. The software for the spacecraft thrusters uses the wrong unit. While management failure played a role, this would never have happened if an automated validation process existed – one as simple as tying-up each number with its semantics. The error would have been caught early if a continuously validated spacecraft and orbital simulations existed. Similar problems and misunderstandings happen in the modeling and simulation world where researchers rarely are able to replicate others' simulations quickly, precisely and reliably.

Computational modeling and analysis focuses on employing computers to build model specifications, verify code, and execute simulation. Indeed, the notion of computational modeling and analysis usually means a quantitative run completed by computers and inference/analysis on the results of the computer run completed by human experts. Computers are not employed to help automate inference, validation, model improvement, or experimental design. Figure 1 depicts this imbalance in automation. Recent advances in data mining made automated analysis more common, but data mining deals only with empirical data, not with automatic building, validating, and improving models. Machine learning approaches can be applied to learning logical, mathematical, and statistical models from data, but they have not been extended to automatic construction, validation, and improvement of simulation models.

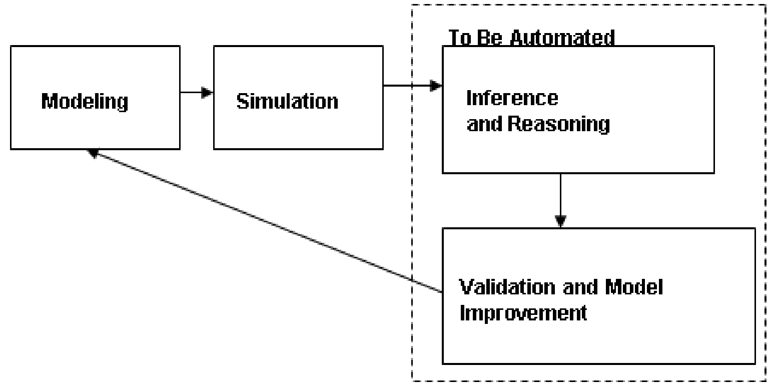

FIGURE 1 Automation of Inference, Validation and Model Improvement

We take the road less traveled to automate validation: an inferential approach uniting simulation with ontological, causal, and knowledge-based reasoning. A tool is implemented based on the approach. This tool is applied to a simulation testbed called BioWar (Carley et al. 2003) which emulates how a city's population reacts to influenza outbreaks.

OUR INFERENTIAL APPROACH

Our inferential approach for validation consists of causal reasoning, knowledge-based reasoning, ontological reasoning, and the scientific method. We call the tool WIZER for What-If Analyzer. WIZER is a knowledge-based tool; the importance of knowledge – and the reasoning based on that knowledge – is emphasized. While WIZER uses statistical tools, they are used in the context of knowledge bases and inferences. Simulation and its outputs are described based on knowledge. Inference rules and descriptions of statistical tools are encoded semantically. WIZER consists of an Alert module, an Inference module, and two knowledge space modules. Figure 2 below shows the diagram of WIZER.

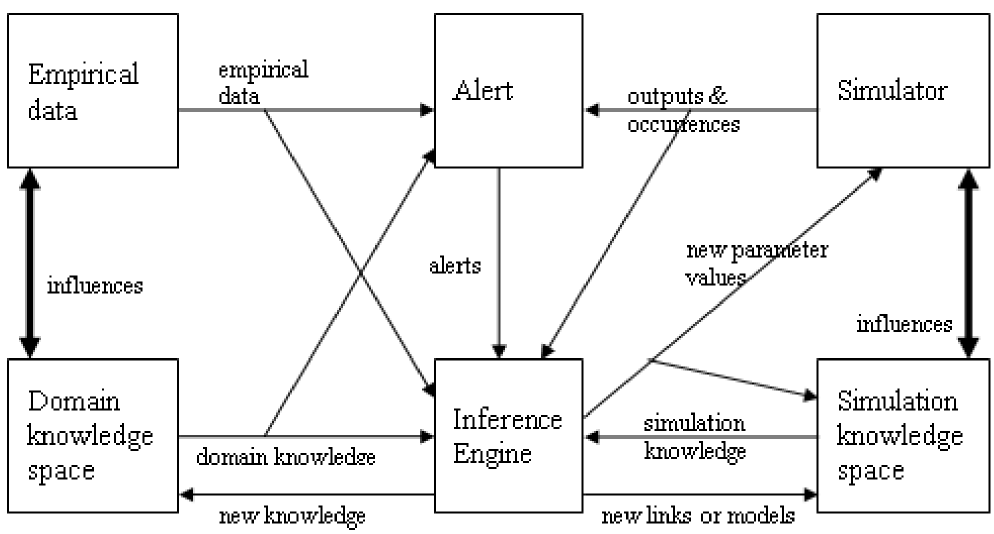

Figure 2 WIZER Diagram

The Alert module does two tasks: (1) describing data using statistical and pattern classification tools, (2) matching a data description with empirical data, producing semantic alerts. Alerts here are defined as semantic characterizations of numerical data (not just alerts in the sense of imminent danger). For example, the Alert module can semantically describe the ups-and-downs of a school absenteeism curve taking into account other semantic or contextual information such as holidays and weather-incurred closings. While not depicted in the figure, the Alert module can also semantically categorize input data and empirical data.

The Inference Engine takes in the outputs from the Alert module, the simulator's causal diagram, a meta-model of the simulation's knowledge space, combined with empirical data, domain knowledge, and parameter constraints (of the domain knowledge space), to determine which parameters, causal links (Pearl 2000, Pearl 2003), and model elements to change – or not to change – and how. The Inference Engine calculates the minimal number of perturbations to the simulation model to best fit the outputs. The model (including the causal diagram) and any potential alternate models are coded into ontologies and rules. Perturbations are implemented as the effects of ontological and rule-based reasoning. An inference produces new parameters for the next simulation. This cycle repeats until a user-defined validity level (which can be defined semantically) is achieved. The user interface module is not shown in the figure for clarity.

The Domain knowledge space module provides domain knowledge to the Inference Engine. Empirical data can change domain knowledge and domain knowledge can ascertain what empirical data are relevant. This depends on the strength of evidence supporting the knowledge and the data. The Simulation knowledge space module provides the simulator with knowledge such as the causal network of the simulation model. The Inference Engine produces new parameter values and possibly new links for the Simulation knowledge space module. The simulator influences and is influenced by the Simulation knowledge space module. The parameter data is empirical, but this empirical data is used in the simulator. Because the empirical data used in the simulator is not the same as the data used for validation, the delineation is clear. Both domain and simulator knowledge spaces are represented by a graph. We use an RDF-based semantic representation. This semantic representation describes and facilitates control of simulation models, knowledge spaces, results, inferences, and statistical tools. In the N3 notation, the basic syntax for RDF is a simple one: *<variable1> <relationship> <variable2>*, where *variable1* represents a subject, *relationship* a verb, and *variable2* an object part of an English sentence. In our implementation, the verb "causes" can specify empirical relationships. In a conventional ontology and semantics, an ontological and semantic relationship is defined conceptually and logically – based on description logics – and not empirically. Scientific method is employed to get empirical causal relations.

RESULTS OF WIZER RUNS ON A SIMULATION TESTBED

Here we present the results of WIZER runs on the BioWar simulation testbed. BioWar (Carley et al. 2003) is a city-wide simulation model of weaponized biological and chemical attacks on a demographically-realistic population with a background of naturally-occurring diseases.

We describe below the results for one validation scenario that examines the relative timing of the peaks of the children's absenteeism curve and the incidence curve. The empirical data for this scenario is gathered from the National Institute of Allergy and Infectious Disease (NIAID).

Validation Scenario: Absenteeism Curves

The variables and output values for this scenario are as follows.

- Outputs for empirical matching: we choose the simulated actual incidence and school absenteeism drug purchase curves.
- Variables: because the onset of absenteeism is influenced by symptom onset and symptom severity, these two factors are important model variables.

The knowledge base consists of causal rules and "IF-THEN" rules related to the causal ones. The causal conceptual diagram is as follows:

(causes symptom-onset absenteeism-onset)
(causes symptom-severity absenteeism-onset)
(convertible infection-rate incidence-rate); *computable from each other*.

Onsets are computed with reference to the time of infection. The rules related to the causal relations are as follows:

(if-then (toosoon absenteeism-onset) (op-lengthen symptom-onset))
(if-then (toolate absenteeism-onset) (op-shorten symptom-onset))
(if-then (toosoon absenteeism-onset) (op-lower symptom-severity))
(if-then (toolate absenteeism-onset) (op-higher symptom-severity))
(if-then (tooshort absenteeism-vs-actual-incidence)
    (op-toosoon absenteeism-onset))
(if-then (toolong absenteeism-vs-actual-incidence)
    (op-toolate absenteeism-onset))

The simulation instantiations of variables are as follows:

(setvalue symptom-onset 2)
(setpriority symptom-onset 3); *priority for conflict resolution*
(setvalue symptom-severity 3)
(setpriority symptom-severity 1)

The simulation instantiations of outputs are as follows. One BioWar simulation of Hampton city (population 142,561 persons) with 100% scale is run. The Alert WIZER computes the peaks of the actual-incidence and school absenteeism curves. It outputs the relative timing of the peaks. The following figure shows the actual-incidence curve.

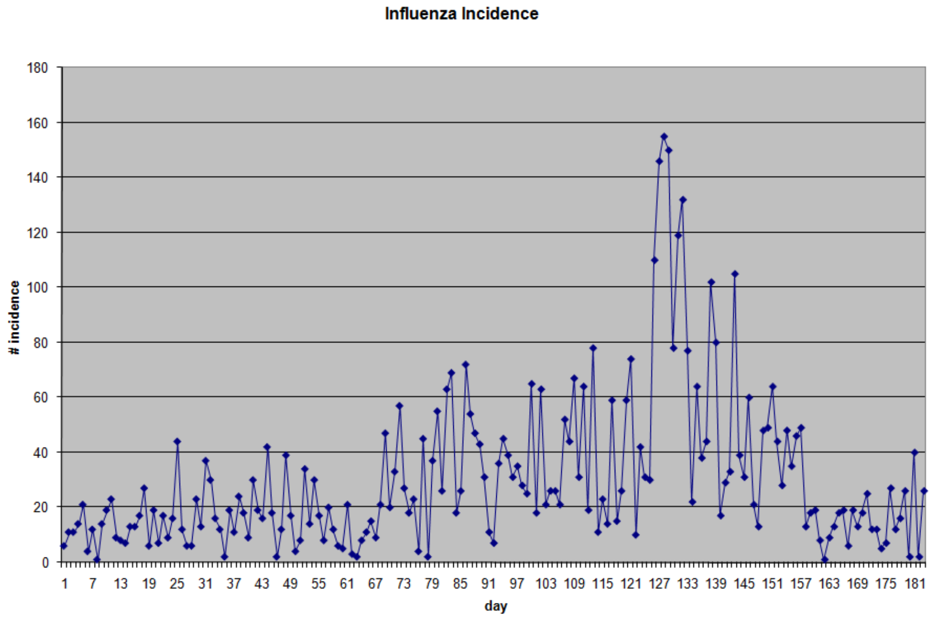

Figure 3 The Peak of Incidence Occurs on Day 128

As shown, the peak of incidence occurs on Day 128. Day 1 is the start of the simulation, corresponding to September 1, 2002. In the simulation trial, the relative time difference between simulated absenteeism and simulated actual-incidence peaks is 10 days.

(setvalue absenteeism-vs-actual-incidence 10)

The empirical data gives 1-4 days as the incubation period for influenza. Absenteeism occurs a day after the end of incubation. Thus, the empirical data are as follows:

(setvalue emp-absenteeism-vs-actual-incidence-lowval 2)

(setvalue emp-absenteeism-vs-actual-incidence-highval 5)

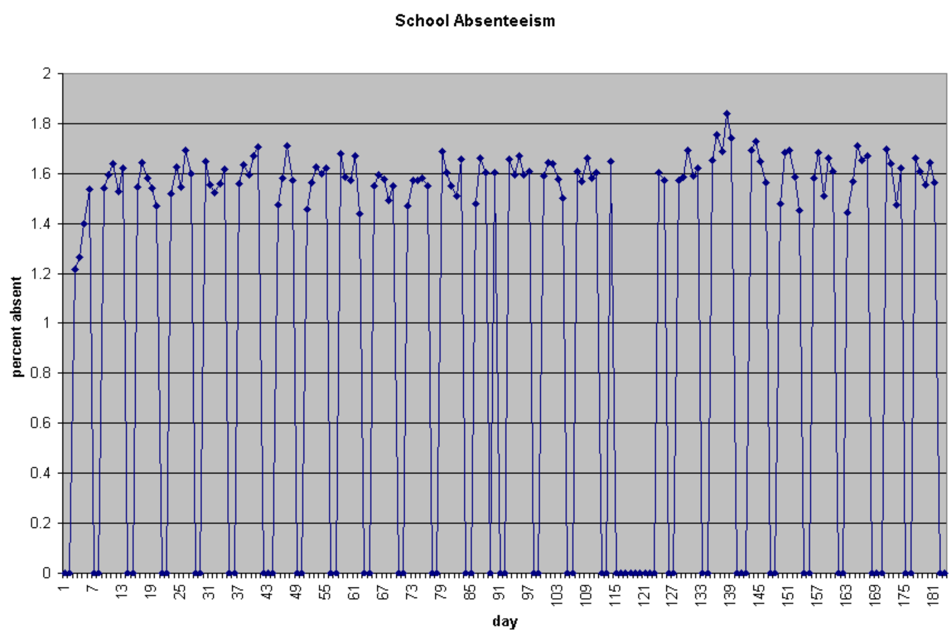

Figure 4 The Peak of School Absenteeism Occurs on D

As shown, the peak of school absenteeism occurs on Day 138. The curve is discontinuous on Saturdays and Sundays because schools are closed. Days 115-121 are holidays.

The Inference Engine compares the relative timing of absenteeism and incidence peaks with the empirical relative timing. After conflict resolution based on the priority value (here other weighting factors are not considered), it produces the following inference

(toolong absenteeism-vs-actual-incidence)

(op-higher symptom-severity)

because the absenteeism peak lags 10 days behind the incidence peak; twice as long as the empirical maximum of 5 days.

The inference is that the symptom-severity (the relative magnitude of manifested symptoms) should be increased. For the next cycle of the simulation, symptom severity is increased by 100% using an encoded rule about critical point heuristics. BioWar is re-run and then WIZER is re-run. The following figure shows the resulting school absenteeism curve.

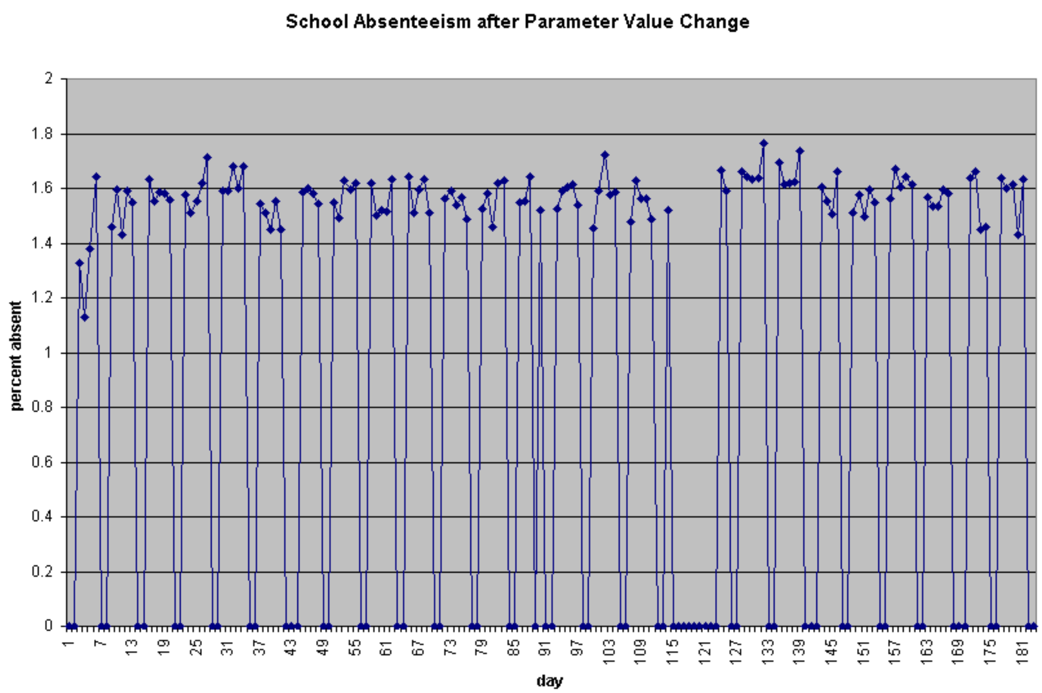

Figure 5 The Peak of School Absenteeism after Change Occurs on Day 132

As shown, the peak of school absenteeism now occurs on Day 132. The Inference Engine compares the relative timing of absenteeism and incidence peaks with the maximum empirical relative timing. After conflict resolutions are performed, it now produces the inference of:

(within-range absenteeism-vs-actual-incidence)

(op-valid)

The relative time difference between absenteeism and actual-incidence peaks is now 4 days, less than the previous cycle's relative time difference of 10 days, and now one day shorter than the maximum empirical time difference. Thus, the peak of school absenteeism has moved into the valid, empirically bounded range of 2-5 days. The Inference Engine announces that the simulated absenteeism curve peak is now valid.

The following figure shows the school absenteeism curves before and after the parameter value change.

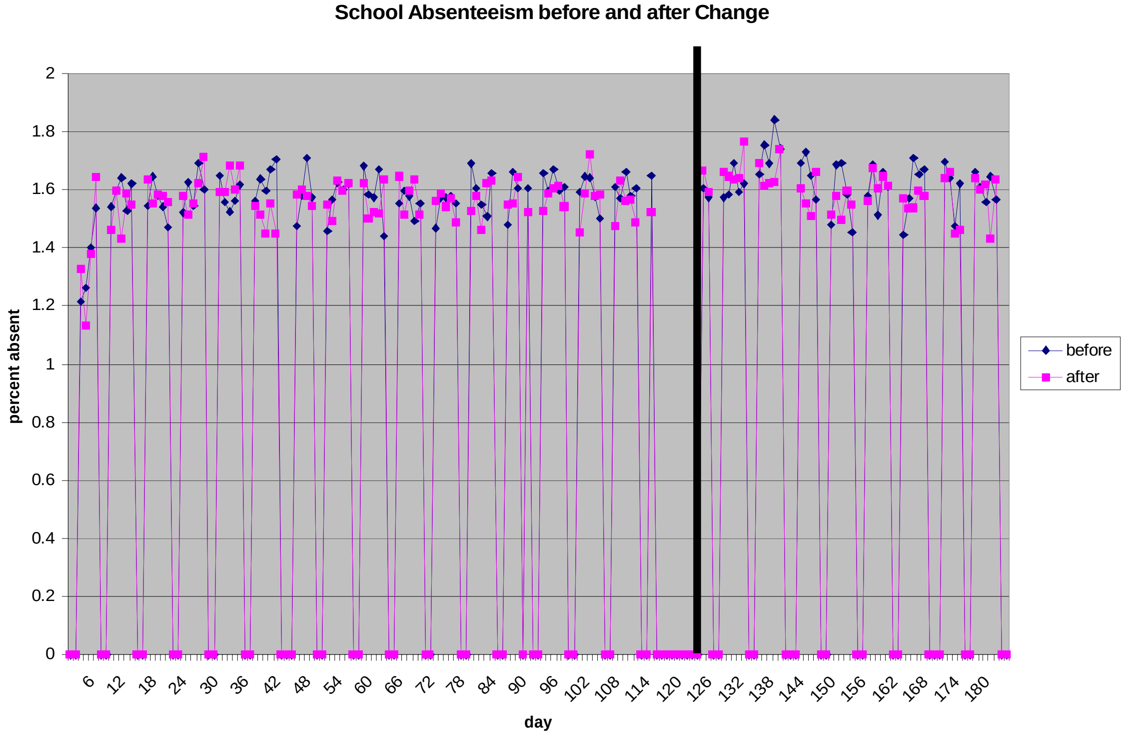

Figure 6 School Absenteeism Curves before and after Parameter Value Change

As shown, after changing the parameter value, the absenteeism peak moves closer to the incidence peak (as shown by the black vertical line).

Validation Measures

Validation is measured based on a piece of knowledge that corresponds to a data stream. For the results on school absenteeism above: initially, the simulated school absenteeism peak occurs later than it should be. Thus this data stream has zero validity, strictly speaking. After parameter values were changed by WIZER, the simulated absenteeism peak moved to within the empirical range, achieving validity.

WIZER versus Response Surface Methodology

BioWar has hundreds of parameters. The resulting parameter space is gigantic. Suppose that the Response Surface Methodology or RSM (Myers and Montgomery 2002, Carley et al. 2004) is used to completely characterize BioWar for validation. Let us assume that that BioWar has 200 parameters (a conservative number) and that each parameter can have 3 different values (3 levels), the parameter space is *3^200* cells, unmanageable with any current technology. As BioWar is stochastic, each cell requires multiple virtual experiments (assume here 40 virtual experiments) to achieve statistically significant results.

Experimenters, of course, can divide the system into modules and validate module by module, assuming all other modules have reasonable parameter values and the existence of some modularity in the system. If this is done for BioWar, experimenters can probe the relationships between incidence rate and infection factors such as *ailment effective radius* (initial infection radius), *ailment exchange proximity threshold* (person-to-person transmission of bioagents for contagious diseases) and *base rate* (initial infection rate). Assuming each of these factors has 3 levels (3 possible values) the following table shows the number of cells required.

TABLE 1 Cell number for incidence factors validation

Parameter	Categories	Size
Ailment effective radius	500, 1000, 1500 meters	3
Ailment exchange proximity threshold	500, 1000, 1500 meters	3
Base rate	10%, 30%, 50%, 70%	4

As shown, the total number of cells required is *3 x 3 x 4 = 36* for the non-stochastic case. Being stochastic, BioWar requires at least *36 x 40 = 1,440* virtual experiments.

WIZER enhances the way experimenters decide which parameters and what parameter levels to choose by codifying the knowledge in a form that is clear, explicit, and operable by computers. With its inference engine, WIZER can reason about parameters and simulation results, producing new inferences. Furthermore, utilizing its knowledge inference, WIZER can reduce the number of virtual experiments needed. The above number of virtual experiments for RSM of *1,440* is the upper limit of what WIZER needs. Typically, WIZER needs fewer due to its inferences about simulation results after each simulation cycle. The better the inferences and the knowledge bases, the fewer the number of required simulation runs.

The following table shows what WIZER gains when used for BioWar. The gain is compared against what normally transpires when humans do the validation. The numbers are first-order estimates. The time it takes for WIZER depends on computer speed, memory, and storage capacity. In addition to BioWar, we have validate a simulation model of socio-cognitive co-evolution called CONSTRUCT (Carley 1991) using WIZER with comparable performance gains.

TABLE 2 Advantages of WIZER versus manual validation for BioWar

Aspect of validation	Manual validation	WIZER
Time to generate input data	Days if not weeks, due to the data access rights, usage policy, non-disclosure rules, privacy concerns, data ownership rights, and other problems.	Days if not weeks, and longer than what it takes if done by human, as the data needs to be formatted and prepared for computer processing
Number of points in response surface that can be estimated	1 per 10 minutes	20 per 10 minutes
Number of data streams	One data stream examination per 15 minute	Many more data stream examinations (>15) per 15 minutes, limited only by computer speed
Knowledge management	Difficult	Facilitated
Number of rules processed	One per 5 minutes	300 per 5 minutes
Number of causal relations considered	One per 5 minutes	300 per 5 minutes
Selection of experimental variables	Implicit but good, depending on experience	Explicit and computer operable
Use of statistical tools	Depends on experience	Encoded in the inference
Documentation of inference and experiment steps	Need extract work	Included in the inference trace
Ability to explain simulation results	Depending on experience	Part of inference trace
Enforced precision	No	Yes
Enforced clarity	No	Yes
Man-hours	Large	Medium-to-Large
Retention of knowledge	Depends on personnel	Facilitated
Large problem solving	Possible, e.g., by careful analysis	Facilitated

ON THE CONSTRUCTION OF KNOWLEDGE SYSTEMS

While the automation of simulation validation brings efficiency, there is an upfront investment in the construction of knowledge bases and inference rules. There is also a research question of how one validates the knowledge bases and inference rules.

Our perspective on the above issues is multi-faceted. The knowledge codification in a form that is clear, explicit and operable by computers facilitates replication of simulations and their results. Replication of results is critical for scientific progress. Current practices of simulation validation (which is usually done by people who construct the simulation) leave this critical issue of replication as an afterthought. As a result, it slows down the scientific advance of modeling and simulation. The codification of knowledge is necessary to do sound engineering and science. Current codification focuses on model specifications and usually has a form of formalized English language. (Codification for code verification can be done using formal methods.) It is straightforward to require that the codification be done not just for model specifications but also for validation specifications and that it be done in a form that is operable by computers. It is a simple extension of existing activities to cover broader scope. The fact that now the codification allows computer automation will recoup some of the time and resources investment spent on codification. Needless to say, this is similar to building houses by designing detailed engineering models beforehand. People can build houses without detailed designs but this often results in a quality-problem and delays in construction. Indeed, quality does matter, not just quantity. Codification also facilitates collaboration. To ameliorate the startup investment cost, we are implementing simulation infrastructure to help modeling and simulation practitioners encode their model and validation specifications in a form operable by computers.

An investment in clear, explicit and computer-operable representation of knowledge for specification and validation is also useful because this higher-level of representation can help reduce errors in the specification and validation process. This is analogous to the fact that high-level languages such as Java helps reduce programming errors and increase programmer productivity as contrasted to the low-level assembly language or machine-level machine-code. An upfront cost here is the compiler and, in our case, the knowledge bases and inference rules. In fact, modeling and simulation itself is an investment vis-à-vis construction without models. Boeing is successful in using computer modeling and simulation in lieu of physical prototyping in the construction of its latest airplanes.

Avoiding a conflict of interests, it is a good practice to separate people or institutions who build a simulation system with those who validate it. Thus, the validation people will build their own knowledge bases for validation. Current validation process has already put validation specifications on paper. It is an extension to current activities to put those specifications in a form that is clear, explicit and operable by computers. We will have knowledge bases from validation stakeholders and from model builders. Comparing these is one way to validate the knowledge bases. The fact that we structure our knowledge bases according to causality simplifies the validation of the knowledge bases. The issue of knowledge bases validation as a whole, however, is a subject for another paper.

Another aspect of validation of large and complex simulations is the need to facilitate collaborations among diverse experts located at various locations. For this, an explicit and clear specification of simulation models and results facilitates collaborations. WIZER and cyberinfrastructure components can help researchers setup, run and replicate simulations with precision and speed.

The conventional knowledge systems have a weakness of being brittle, which means that the inferences will go awry if they are employed outside the specified application domain. It is also hard to ensure the correctness of knowledge and inferences when new rules are added. We address these issues by restructuring rules in knowledge bases using causality and by using knowledge systems strictly within their application domains. For large simulations such as BioWar, we have multiple knowledge modules representing different domains. This is similar to what happens in human problem solving: epidemiologists deal with diseases and symptoms, city health officials deal with quarantine and other response policies, police deal with how to maintain security and order, first-responders deal with how to give first-aid quickly, etc. As causality is empirical, the inferences are grounded on empirical knowledge and data. Structuring knowledge bases along causality is one way to partition the knowledge bases into smaller, more coherent and more manageable knowledge bases. A related work on scaling up knowledge bases is structure-based partition (Amir and McIlraith 2005, Ramachandran and Amir 2005). As WIZER is a knowledge-based causal system, it can scale well given appropriate knowledge including statistical knowledge. BioWar itself is a sufficiently complex model to test validation approaches: it can represent a demographically-realistic, spatiotemporally-realistic, and features-rich city with millions of people. In the real-world, statistics is used to scale economic models and market indicators are used to scale the model measurements. If we would like to have precise world-scale validated economic models/simulations, we will encounter the challenge of getting the proper data before the challenge of inference. As more and better economic data become available, WIZER can scale with the data and help build better economic models.

We have argued above that it is critical and necessary to invest in a clear, explicit and automated representation and validation of simulations because we need to have replications to do good science. It is also a good engineering practice to have design schematics covering the entirety of the system (not just model specifications but also model behaviors and results) that can be automatically checked and executed by computers. While it takes time and resources to construct knowledge bases and inference rules, Table 2 indicates that once this is done we can recoup the investment and get the dividends. The needed investment is also reduced as the construction of knowledge bases and inference rules can piggy-back the model specification activities.

DISCUSSION

WIZER is unique in that it pioneers ontological and knowledge-based inference for simulation validation and model-improvement. WIZER is a causal and logical reasoning, hypothesis building & testing, and simulation control engine with statistical and pattern recognition capabilities. It strives to employ deep and structural knowledge by employing causal and ontological reasoning. WIZER seeks to emulate scientists doing experiments and analyses via the scientific method, rather than providing another methodological approach or programming environment. The causal reasoning component provides link to empirical data and knowledge of scientific experiments.

While other toolkits such as Swarm (http://wiki.swarm.org), TAEMS (O'Hare and Jennings 1995, Lesser et al. 2004), and Repast (http://repast.sourceforge.net) provide programming environments for agent-based simulation systems, WIZER is designed to help with scientific experimentation, validation, scenario analysis, and model improvement. WIZER is able to run on top of any simulation system, including those constructed using Swarm and Repast toolkits.

The following table compares WIZER and other tools:

TABLE 3 Feature comparisons between WIZER and other techniques

	WIZER	Swarm/TAEMS/Repast	Evolutionary Strategies	Data Farming
Programming environment?	No	Yes	No	No
Unit of inference	Rule and causation	None	Evolutionary and genetic operators	Data growing heuristics
Object of operation	Simulation, data and knowledge	Code	Simulation and data	Data
Experimentation?	Yes, automated	Yes, human operated	Yes, automated (fitness)	No
Knowledge operation?	Yes	No	No	No

WIZER differs from evolutionary programming (Fogel 1999), evolutionary strategies, and genetic algorithms in that it does not need a population of mutation/crossover candidates nor does it need mutation, crossover, or other evolutionary and genetic constructs. Instead, WIZER applies knowledge inference to simulations to determine the next simulation run. If the result of inferences mandates a radical change, a revolution will occur. From the historical point of view, evolution took millions of years to affect change, while the application of the scientific method after the Renaissance advanced science and affected changes on the Earth's surface in only a few hundred years.

Our approach facilitates the integration of simulation and knowledge inference. As a simulation runs, producing perhaps emergent behaviors, simulation-based knowledge is automatically captured and analyzed. As knowledge changes, the simulation can be changed.

For social sciences, the inferential approach allows investigation of the foundations of social networks, first by the validation of agent-based systems and in future by the validation of more realistic systems (e.g., physical models). Unlike the Exponential Random Graph Model (ERGM) or p* (Robins et al. 2006) which attempts to characterize the probability of social network structures in a top-down manner using pure statistics, WIZER can be used to characterize a range of agent behaviors and the resulting emergent social behaviors from agent interactions in a bottom-up fashion and within a proper context of the application and semantics. Our inferential approach indicates a path toward more profound theories for social interactions and group behaviors.

In the scientific community, the explosion of data and the need for collaboration paved the way for cyberinfrastructure, which is an infrastructure for data acquisition, data management, knowledge sharing, visualization, and collaboration over the Internet for scientists and engineers. The linkage between simulations within cyberinfrastructures and knowledge inferences is not yet automated. Our approach suggests one way to automate the linkage and thus provide a simulation infrastructure for scientists and engineers. This may speed up the analysis of massive data sets. Social scientists, artists and humanists in particular need a simulation infrastructure to play out and investigate phenomena that cannot be described by math, logic and statistics alone.

The inferential approach underlying WIZER for simulation validation facilitates more precise research in organization and management sciences. As data become more available, aided by high performance computers, the simulations become more precise enabling more detailed theories to be built and tested.

References

Amir, E. and S. McIlraith, 2005, "Partition-Based Logical Reasoning for First-Order and Propositional Theories," *Artificial Intelligence*, 162 (1-2), pp. 49-88.

Carley, K.M., 1991, A Theory of Group Stability. American Sociological Review, 56(3): 331-354.

Carley, K. M., D. Fridsma, E. Casman, N. Altman, J. Chang, B. Kaminsky, D. Nave, and A. Yahja, 2003, *"BioWar: Scalable Multi-Agent Social and Epidemiological Simulation of Bioterrorism Events,"* Proceedings of North American Association for Computational Social and Organizational Science (NAACSOS) Conference 2004, Pittsburgh, PA, http://www.casos.ece.cmu.edu/casos_working_paper/carley_2003_biowar.pdf.

Carley, K.M, N.Y. Kamneva and J. Reminga, 2004, *Response Surface Methodology*, CASOS Technical Report CMU-ISRI-04-136, Pittsburgh, PA

Fogel, L.J., 1999, *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*, Wiley Series on Intelligent Systems, New York, NY.

Lesser V, K. Decker, T. Wagner, N. Carver, A. Garvey, B. Horling, D. Neiman, R. Podorozhny, M. NagendraPrasad, A. Raja, R. Vincent, P. Xuan, and X.Q. Zhang, 2004, *Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. Autonomous Agents and Multi-Agent Systems*, Volume 9, Number 1, Kluwer Academic Publishers, pp. 87-143.

Myers, R.H. and D.C. Montgomery, 2002, *Response Surface Methodology: Process and Product Optimization using Designed Experiments, 2nd ed.*, New York, NY: John Wiley.

O'Hare, G. and N. Jennings, 1995, *Foundations of Distributed Artificial Intelligence*, Wiley Inter-Science, pp. 429-448.

Pearl, J., 2000, *Causality: Models, Reasoning, and Inference*. Cambridge, UK: Cambridge University Press.

Pearl, J., 2003, "Statistics and Causal Inference: A Review," *Test Journal* 12 no. 2 (December): 281-345.

Ramachandran, D. and E. Amir, 2005, Compact Propositional Encodings of First-Order Theories, in 20th National Conference on Artificial Intelligence (AAAI'05).

Robins, G., T. Snijders, P. Wang, M. Handcock, and P. Pattison, 2006, "Recent Developments in Exponential Random Graph (p*) Models for Social Networks," *Social Networks, forthcoming*.

Yilmaz, L., 2006, "Validation and Verification of Social Processes with Agent-Based Computational Organizational Models," *Computational & Mathematical Organization Theory*, vol. 12, no. 4, pp. 283-312.