A DIALOG ON THE LIMITATIONS OF DIS

Thomas L. Clarke and Michael D. Myjak
Institute for Simulation and Training, University of Central Florida
3280 Progress Drive, Orlando, FL 32826

ABSTRACT

Distributed Interactive Simulation (DIS) is both a protocol for networking simulators and a style for implementing Department of Defense training simulator networks. Follows-on to DIS such as ADS (Advanced Distributed Simulation) or other architectures that adopt the style will have to overcome the same networking and communication problems now being faced by DIS. This paper discusses these problems in an abstract, mathematical context so that the conclusions will be applicable to DIS and its heirs.

DIS is defined as a formal system which can be extended to cover any conceivable simulator networking architecture. This formalism is used to prove several theorems about the computational capability of formal DIS networks. These theorems demonstrate that quite simple DIS networks can attain computational universality. Of significance for the DIS standards process is the result that the specification of protocols for DIS networks of Turing machines can only approach completion asymptotically. Limitations are also found in the ability to validate a simulation as a result of mismatches in simulation host speeds.

When DIS networks simulate physical processes, a relationship between the maximum simulated physical velocities of entities and the propagation delays is derived. Another relationship is found between the network propagation delays and the maximum rate of state change of network entities. These maxima are only a problem for networks with very fast movers or with very long propagation delays.

To keep the presentation suitable for a general audience, the formalism and the conclusions are presented in the form of a Socratic dialog. The talk will consist of the dialog as staged by the two authors. One will play the inventor of DIS, Thorpus, and the other his skeptical critic, Skepticus.

Thomas L. Clarke

Education

Doctor of Philosophy, Applied Mathematics, University of Miami, 1982

Master of Science, Applied Mathematics, The University of Virginia, 1975.

Bachelor of Science, Mathematics, Florida International University, 1973.

Experience

Institute for Simulation and Training, University of Central Florida, Orlando, Florida Principal Mathematician, June 1988 to Present

Atlantic Oceanographic and Meteorological Laboratory, Miami, Florida Mathematician, 1975 to 1987.

Michael D. Myjak

Education

Master of Science, Computer Science, University of North Texas, December 1988

Bachelor of Science, Computer Science, Clemson University, 1982.

Bachelor of Science, Electrical Engineering Technology, Clemson University, 1982.

Experience

Institute for Simulation and Training, University of Central Florida, Orlando, Florida Senior Research Scientist, June 1994 to Present

McDonnell Douglas Space Systems Company, Kennedy Space Center, Florida Principal Engineer, Integrated Network Services, July 1989 to November 1993

Texas Instruments, Corporate Research, Development, & Engineering Division, Dallas, Texas. Member of Technical Staff, Computer Science Laboratory , January 1983 to June 1989.

A DIALOG ON THE LIMITATIONS OF DIS

Thomas L. Clarke and Michael D. Myjak
Institute for Simulation and Training, University of Central Florida
3280 Progress Drive, Orlando, FL 32826

I. INTRODUCTION

Participants: Thorpus, Skepticus

Thorpus is a legionary centurion who is the chief advocate of training soldiers with distributed networks of interactive simulators. Under his guidance, a networking protocol known as DIS (Distributed Interactive Simulation) has been developed through a series of DIS workshops that bring together representatives of all major simulation manufacturers. At these workshops problems with the DIS protocol are identified and solutions are suggested. Periodically the workshop then issues a new standard version of DIS that codifies DIS protocol elements that have been accepted by consensus at the workshops.

Skepticus is a natural philosopher and computer scientist who is skeptical of the ability to reach the goal of a communications protocol, such as DIS, that can encompass the intercommunication needs of all simulators.

Thorpus: I have a vision for the future of simulation. Thousands, perhaps even millions of individual simulators networked together to implement simulations beyond the capability of individual simulators. These simulators will be distributed across both real and virtual spaces but will continue to interact; for short I call it DIS, Distributed Interactive Simulation.

Skepticus: I have serious doubts about this. The work of Turing shows that any universal computer can do anything any computing device whatsoever can. Your network of simulators, each of which I assume contains a universal computer, would be just a large universal parallel computer. As such it would fall under the limitations of Amdahl's law; communication inefficiencies would make the networked simulators less effective than a single simulation computer of equal cost.

Th: You are assuming a naive simple model of inter-simulator communication. I have developed schemes and protocols that greatly reduce the network traffic.

Sk: For the moment then let us concede that communication overhead can be reduced, but I would like to return to this point later. I foresee other problems with your network. I assume that you will want to have different makes and models of simulators on your network. How will you insure that they can all work together? How do you keep it from becoming a Network of Babel?

Th: That is easily taken care of by the establishment of network protocol standards.

Sk: We will have to discuss these standards later in the context of language theory, but more problems with DIS continue to occur to me. Assuming that both communications overhead and network standards issues can be dealt with, what will be the performance of the network? If your simulators are very far apart, there will be significant propagation delays that will slow your DIS network.

Th: I assure you there are means to deal with the issue of propagation delay. By properly assigning computation tasks to the networked simulators involved in the DIS exercise, the deleterious effects of propagation delays can be eliminated.

Sk: Again you will have to give me the details later, but now I remember that you mentioned virtual space. What does this mean, do you include simulators in space, near the moon? Will this not cause an accuracy problem for your simulators? What about continuous fields, like electromagnetism and sound? What about intelligence?

Th. (sputtering.) Hold on! Let me explain DIS from the beginning. I think I can meet all your objections. If I can't, perhaps you can recommend ways that I can improve DIS.

DIS is an outgrowth of an earlier project that I directed, SIMNET. Under my sponsorship, Bacchus, Barthalomew, and Nestor developed the first network of chariot training simulators. Certain communication problems became evident in developing SIMNET. Communication delays, limited network bandwidth, etc., were problems that had to be overcome.

BBN's brilliant solution was to use dead reckoning. Each simulator maintains a model of every other simulator on the network. In the absence of communications from the other simulators, this model simulator continues to move from the last known position at the last known velocity, so that network bandwidth is not used up constantly sending position information ...

Sk: (Interrupting) Yes! Very clever. Then you would send true position information only every so often to maintain accuracy. Are these true positions sent at fixed time intervals?

Th: In the absence of other reasons, true positions, or updates, are sent at a minimum rate that is set by the DIS protocol. Typically one update or status packet is sent every second so that other simulators are constantly reminded of the sender's existence and do not drop the simulator from consideration. The usual terminology in DIS is to refer to a packet of data a protocol data unit (PDU).

But to maintain highest accuracy, a given simulator not only runs models of other simulators, but also contains a model of itself. The dead-reckoned position of the self-model is constantly compared with the true position of the simulator. Whenever the self dead-reckoned position and the true position differ by more than a protocol determined threshold, a new update PDU is broadcast to all other simulators. Since the self dead-reckoning is in error, it is known that all other dead-reckonings will be in error as well and will require correction.

This scheme for using dead-reckoning models, sometimes referred to as ghosts, also alleviates many problems associated with propagation delays, transmission errors etc. No simulator can remain in error for very long since it is constantly receiving update PDUs from other simulators.

If a simulator joins the networked simulation exercise in progress, within a short time it will have picked up all the other simulators on the network via their minimal stay-alive PDU transmissions. If a PDU is missed due to transmission error, the dead-reckoning will maintain a good approximation until a good PDU is obtained.

Sk: All right, I can see how this DIS scheme with its dead reckoning or ghosts will support a network of training simulators, at least in the first approximation. However, I see many problems that present severe difficulties for implementing any such network.

Let me explain some of these objections in detail.

II. COMMUNICATION OVERHEAD

Sk: Let us begin with how you propose to reduce the network communication overhead and beat Amdahl's law (Quinn, 1987). (Writing on blackboard) If I recast Amdahl's law to apply to your DIS network, it would take the form:

$$S <= 1/(f + (1-f)/N)$$

where S is the speedup factor for the network as compared to carrying all of the simulations in a time-shared fashion on a single simulator. The quantity f is the fraction of operations that must be carried out sequentially, that is operations that require data be passed from one simulator to another and modified by each simulator in turn. N is the number of simulators.

Oh, I distinguish here between simulations and simulators. Simulation is the mathematical model of the real world, a simulator is the combination of hardware and software that implements the simulation. Do you find this a good distinction?

Th: Yes, I like that distinction. It is good to make clear the difference between the model and the implementation of the model.

This distinction opens the way to circumventing Amdahl's law. Consider that the simulation is of the physical world and that all communications in the physical world are point to point, from space-time event to space time event. There are no physical processes that require that data travel from one point to another and another before the outcome is determined. The problem with Amdahl's law which arises from the fraction f or irreducibly serial processes is a problem with the implementation, with the simulator, not with the simulation.

As I explained, DIS as currently implemented uses a process called dead reckoning to avoid the need for serial calculations. Each simulator simply broadcasts information about its state. Other simulators receive these broadcasts and use the information to update internal models of the other simulators; the process of maintaining these internal models between updates is called dead reckoning.

III. NETWORK STANDARDS

Sk: As I mentioned, I find it difficult to believe that it is possible to guarantee that a simulator manufactured by XYZ Computer Corp will in the DIS environment with a simulator manufactured by ABC Aerospace Corp.

Th: I'm glad you reminded me of that. It gives me an opportunity to discuss the process whereby standards are being developed for DIS operation. A DIS Standards workshop has been conducted twice a year in Thamaturgia for the past seven years. These meetings have been very energetic. Version 1.0 was generated very quickly. Version 2.0 was finalized last year, and version 3.0 is well underway.

Sk: Version 1.0, 2.0, 3.0? Isn't it a bit of an oxymoron for a standard to have so many versions. It seems you have replaced a problem of compatibility between manufacturers with a problem of compatibility between versions of the standard.

Th: Not at all. Let me tell you a bit of the history of DIS. When I was at the Agency, I sponsored the SIMNET project, to prove the concept of networking simulators. The people who worked with me at the Agency developed ad hoc protocols for the SIMNET project that incorporated many of the ideas we discussed earlier.

When the DIS Standards workshops started, this earlier SIMNET protocol, with a few rough edges smoothed over, became version 1.0. Version 2.0 incorporated features and information packets that were beyond the scope of the SIMNET project. Now that more experience has been gained with DIS, the limitations of 2.0 are becoming clear and will be removed in version 3.0

Sk: How can you be sure that this process will end, or at least approach an asymptotic optimum? What if some future version reveals a fundamental flaw in the D of DIS. I am still not at all convinced that DIS does not have fundamental limitations.

Th: Please bear with me. I think all doubts will be removed at the end of our discussion.

I think the ability to change and adapt to new requirements is one of the great strengths of the DIS standards process. It is not a flaw.

Sk: Here is a problem. Think of the DIS packets as elements of a formal language. The simulators on the DIS net are Turing machines, or else it would not be a very interesting simulation. Thus the language must be context sensitive if the internal state of the Turing machines are to be transmitted. This is a new result, that I just worked out last night; I am including the formal proof as an appendix to our dialog.

To put it another way since it takes a Turing machine to recognize a Type 0 or Recursive language, a type 0 or recursive language is needed to transmit the state of a Turing machine. However, most programming languages are Type 2 or deterministic context free.

It seems likely that the standards workshops will thus always generate a Type 2 language. Thus the DIS standard will always be inadequate for transmitting the Type 0 behavior of the Turing machine simulators.

Th: Possibly. You will have to give me time to examine your proof. But if it is true it will represent job security for generations of DIS Workshop attendees!

IV. GENERAL IMPOSSIBILITIES

Skepticus: Your DIS network of simulators has as its ultimate goal, does it not, the simulation of battle in all its fog and confusion? Is not warfare one of the most complex of human endeavors, involving psychology, physics, engineering, in short all of the sciences?

Thorpus: Yes. What's your point?

Sk: Science in the last hundred years has discovered a wide variety of things that it cannot effectively model. I believe these phenomena will present very great difficulty to your goal of simulating warfare.

Let me outline these discoveries.

Th: Very well, although I doubt if they will amount to more than ivory tower amusements. Real war is concrete and not subject to these airy fairy academic notions.

Sk: Let me begin then with something that is very concrete, the weather. Weather, and many other natural phenomena must be modeled by non-linear equations. Examples are turbulent fluid flow as in weather, celestial mechanics, most of solid state physics, population dynamics in biology and even the Lanchester type battle simulation equations.

It turns out that in general non-linear equations are very sensitive to initial conditions. A very small perturbation can produce a large change in the outcome. The classic example for weather is that the wind produced by a butterfly's wing beats in Africa can influence the weather in America weeks later. Since it is impossible to know the world's winds to this level of accuracy, and it would not be practical to calculate the winds to the level of detail in any finite sized computer, the weather after about a week is effectively unpredictable.

This phenomena of extreme sensitivity is called deterministic chaos. The equations are predictable in principle, but the effect is as if the phenomena governed by those equations were randomly chaotic. As a result it is in general impossible to predict the evolution of non-linear systems very far in advance, and most systems are non-linear.

Th: Weather, smeather. Battles only last a few days, there's no need to predict weather within a DIS simulation.

Sk: Sigh. But its not just the weather that is chaotic. Science discovers chaos nearly everywhere, whenever a system is non-linear chaos is nearly surely to be found. Recall Ben Franklin's ditty: "A little neglect may breed mischief,... for want of a nail, the shoe was lost, for want of a shoe the horse was lost, and for the want of a horse the rider was lost". Let me add: for the want of the rider the battle was lost, and for the want of the battle the war was lost. Chaos turns the loss of a nail into the loss of a war.

Another recent example was the movie Jurassic Park where chaotic effects let the dinosaurs lose ...

Th: I never waste time on movies.

Sk: Well then let me remind you about D-day. Eisenhower had information that the weather would clear, the German meteorologists said that it would not. Eisenhower bet correctly. The chaos of weather had a significant effect on the outcome of the war.

But enough with chaos, it just introduces randomness into the world. Let me tell you about some actual impossibilities.

Th: Very well.

Sk: Mathematicians have had the dream of formalizing the logical reasoning so that a machine, like a computer, could discover new mathematical theorems. In the nineteen thirties they discoveries that it is impossible to formalize reasoning.

Th: What does this have to do with training for war?

Sk: Isn't it obvious? The simulators in your DIS are machines. Mathematicians have discovered limitations on the kinds of reasoning they can perform. Hence DIS is limited in the same way.

Let me give you the simplest example. You know how hard it is to debug a computer program? Consider one simple bug; you know how programs sometimes get into infinite loops and never stop until you hit the break key?

Th: Yes.

Sk: Wouldn't it be wonderful to have a master program that would read your program and determine whether your program would get into an infinite loop? This would help you quickly eliminate programs that got caught in loops.

Th: Yes. But that's only a very simple bug, what about ...

Sk: Well that is impossible. No program can be written that when given another program as input can tell you whether the subject program will halt.

Th: How can they know that?

Sk: Its a theorem that Alan Turing (Dewdney, 1989) proved by the trick of giving the program itself as input. To make a long story short, the result was a program that would infinitely loop only if it stopped, and would stop only it infinitely looped. This is a logical contradiction, therefore no such program can exist.

Th: But that's a trick! Surely you could add to the program to avoid this problem.

Sk: But then you could do the same "trick" to get a contradiction from the augmented program. Mathematics is like war; Turing's goal was to "kill" the debugging program, to show that it could not exist. To kill it he just had to find a single chink in its "armor" and thrust home, he did not have to strip his enemy naked to find the most vulnerable anatomy. Any gap in the armor will do for dispatching the enemy or a math problem.

Th: I never thought of math as being like war, but I'm still not sure what this has to do with DIS networked training simulators.

Sk: Consider the dead reckoning model. What if the vehicle being modeled is more complicated a simple ballistic projectile, perhaps an unmanned aerial vehicle (UAV) with an internal computer? If you wanted the dead reckoning model to simulate the UAV's behavior fully, your simulator's computer would have to, in effect, solve the halting problem for the UAV computer. This is impossible. If a contractor tells you he can do it, send him packing!

Th: (Writing in a notepad) Yes, that information might be a good check on a couple of my contracts. But I still don't completely see how Turing or anyone can claim there are things that no computer can do. What about artificial intelligence (AI)? The AI guys claim that the will eventually be build a thinking computer. If a computer can think, but there are things it cannot do, doesn't that imply there are things we cannot do in the reasoning department? I don't buy that.

Sk: Now you are getting into philosophical realms where there is much uncertainty. There are those who claim that Godel's theorem, a more general form of Turing's theorem, shows that there are things that people can do that no computer can do. There are others who say that these theorems show the limits of human thought as well.

For now I believe that humans are smarter than computers on Monday, Wednesday and Friday, that computers are the equals of humans on Tuesday, Thursday and Saturday, and on Sunday I don't think about it.

While we are on the subject of computers, there are other limitations on the ability of computers to compute, practical limitations.

Th: Of course. In any given year, computers are only so fast, they only have so much memory. But speed and memory capacity have a compound growth curve. Just wait a few years and computers will have the capacity to solve your problem.

Sk: Let me tell you about NP Complete Problems. Even if a problem is computable or solvable by a computer, there are problems that take too long to solve even on the fastest conceivable machines. Certain problems belong the class NP which appear to be exponentially hard. As the size of the problem gets larger, its difficulty grows enormously fast. Examples include the traveling salesman problem of optimizing the route through a number of cities, the knapsack problem of finding an optimal packing of boxes within knapsacks and many others.

I think you can see the relevance of the traveling salesman to route planning in a military campaign and the knapsack problem to logistic resource allocation.

Th: Yes. Logistics and routing are essential elements of a campaign. But people solve those problems every day. Why are they so hard for a computer?

Sk: Again, the answer to that question verges onto unknown philosophical differences between man and machine, but basically a person seems to find a pretty good solution that is not the absolute best solution. The human's "intuition" gives him assurance that the solution he has chosen is pretty good.

A computer on the other hand, using a "dumb" algorithm pretty much has to find the very best solution. It has no idea of what a pretty good solution is. These NP Complete Problems have enormous numbers of possible solutions and the only way to find the best one seems to be to check every possible solution. There seems to be no better algorithm than checking every possibility.

Heuristics are sometimes applied to reduce the number of solutions searched, but that amounts to converting the programmer's "intuition" into code. Would you trust you battle plan to the "intuition" of a programmer or would you rather use your own "gut feel"?

Th: Trust my battle plan to one of those nerds? Never! (Scribbles in notebook again.) I never did like the output of some of those automated planners. But I still think that the
march of computer power will take care of this problem.

Sk: I'm not so sure. You speak of millions of simulators. Optimizing over such a network is an enormous problem. Computer power is rising exponentially today, but may eventually start to hit quantum or other limits. There is an absolute limit to computational power. It is somewhat tongue in cheek and is certainly humongous, but simply put when a computer is fast it has to be small, when it is fast it has to use high frequency, and hence massive, quanta of energy to transmit information. When it is fast enough the massive quanta are packed into such a small space that the threshold for formation of a black hole is crossed. The computer, and all of its results, then disappears down the black hole!

Th: Very amusing! But one of the purposes of DIS is to train officers and soldiers how to make battle plans. The inability of a computer to find the optimum for the NP Complete problems is hardly relevant.

Sk: But who are the soldier's opponents? Is not the development of a Computer Generated Force (CGF) to simulate the enemy, a large part of your DIS effort.

What sort of algorithms does the CGF use? Does it not run up against NP Complete problems? Is the CGF not forced to use heuristics created by those nerdy programmers?

Th: Wait. We team the programmers with subject matter experts to develop the heuristics, nothing nerdy about the soldier expertise we tap.

Sk: Again we hit the philosophical difference between human and machine. The philosophers Dreyfus and Dreyfus (1986) assert that the knowledge that an expert can articulate to a programmer can be no more than journeyman level. True expertise is inherently non-verbal. If they are right, the CGF may avoid nerdiness, but will be stuck at the level of journeyman soldier, failing to provide a truly worthy opposition to your troops.

Th: This is the best we can do for now, and the training provided by DIS with its, journeyman as you say, CGF is still far beyond anything provided heretofore.

Also, I understand that research into neural networks is supposed to remove this limitation. Neural networks are non-symbolic or something like that.

Sk: Neural networks have some great possibilities, but a discussion would lead us pretty far afield. I would like to talk about some other factors that affect decision making.

Th: Continue.

Sk: This may not affect the military too much since the military is a strict hierarchy, but Nobel laureate economist Kenneth Arrow (1951) discovered a fundamental limitation on collective decision making.

Consider an idealized method for ranking alternatives, so as to choose the optimum for decision making. There are some very natural conditions such a ranking must satisfy. It must provide a ranking for all alternatives, that is not break down for some sets of alternatives. It must be rational that is if A is preferred to B is preferred to C then A is preferred to C. Also if A is preferred to B and a new alternative D is introduced, then A will still be preferred to B. Finally, the ranking is not arbitrarily determined by an authority independently of what the alternatives really are.

Th: Yes, everyone does this when they makes lists of options and assign numerical values to the possibilities. You do have to be careful, or sometimes you get circular preferences, A is better than B is better than C, but C is better than A, as you put it.

Sk: The problem Arrow found was when you want to combine the preferences of two or more individuals to make a group preference. When two or more people are involved, Arrow added two more conditions: no single individual is a dictator who solely determines the group outcome, and if one individual changes his preference from B to A, then the group does not do the opposite by changing the group preference from A to B.

Arrow then proved that it is impossible to construct such a group preference if there are two or more individuals in the group trying to decide among three of more alternatives. Sometimes this is referred to as the voting paradox since election systems are designed to combine individual preferences into group preferences in this way. The problem or paradox arises when there are more than two candidates in an election. The group preference may then turn out to be irrational or circular, the addition of the third party may have paradoxical effects (would Clinton have beat Bush without Perot as third party candidate).

Th: This is all very interesting, but as you pointed out in the military the commander makes the decision so this group decision making problem does not arise.

Sk: I'm not so sure. Have you ever been about two minds concerning a decision? Maybe this maybe that? Your two internal minds could be viewed as two individuals and certainly you are often faced with a choice between three or more alternatives. Thus, the conditions of Arrow's theorem apply so that individual choice would seem to inevitably be irrational or paradoxical on occasion.

Harking back to the CGF problem in DIS, an algorithmic CGF would certainly be logical and rational, whereas a true opponent would be irrational and paradoxical at times. It seems to me that a CGF would have to have some element of irrationality if it were to emulate a true human opponent.

Th: (Making another note) Sounds like the CGF needs a random number generator.

Can we get on with more detailed discussion of some of the limits of DIS you brought up earlier.

Sk: I want to make a brief point first with regard to CGFs. CGFs must not be too good. In real battle a human enemy will have difficulties of perception leading to mis-reporting of battle condition. In addition to the obvious visual effects such as sun in the eyes or atmospheric mirages, observational psychology has discovered many visual illusions. The include the Ponzo, Poggendorff, Zollner and Muller-Lyer illusions which geometrical illusions. The reversing Necker cube and the Rubin face-goblet illusion are also perceptual illusions. CGF algorithms will have to take account of these psychophysical facts if the CGF is to behave like a real enemy.

The human perceptual system seems to respond to fractal forms such as the branching of trees and the shapes of clouds. Fractal-like mathematical monsters were created as various counter-examples to geometrical possibilities and are taught in real-analysis as pitfalls in mathematical theorem proving. Mandelbrot recognized that mathematical monsters were good approximations to objects in nature such as a particle path; he defined the concept of fractal and fractal dimension which has since been found to approximate many natural objects. Thus, I think mathematical ideas are very important to the CGF project ....

Th: OK. OK. You are making a blatant attempt to get some funding for your pet mathematical projects out of the CGF effort. Now let us get on with some details of the limits of DIS.

V. VALIDATION

Thorpus: I have a simple question: does the assignment of damage assessment to the application simulating the target constrain us in any way? Should this be changed?

Skepticus: There are of course the problems of insuring that the target must know the characteristics of the weapon with which it is hit.

Aside from this the only problems arise from propagation delay effects. Target may think the weapon hits delta X meters away since it has moved and the firer's DR algorithm is in error and not updated yet. This can lead to a jump in the "perception" by the target, but as long as the rule is enforced consistently no simulation paradoxes should arise.

The situation will be similar to that in relativity. There depending on the state of motion of the coordinate system, two events may appear to differ in their time order. You may recall the paradox of the hanger door from undergraduate physics. A spaceship 100 meters long is traveling so fast that its Lorentz-Fitzgerald contracted length is 10 meters. A space hanger moving slowly perpendicular to the spaceship has a door of width 20 meters so that the opening arrives at just the right time for the spaceship to pass through it. Recall that relativity is symmetrical so that from the spaceship the doorway only looks 2 meters wide, so how can the spaceship pass through the opening? From the door way the spaceship is short so there is no problem, but from the spaceship the opening is impossibly narrow.

The resolution is that the two events, the nose of the spaceship passing the doorway and the tail of the spaceship passing the opening, while simultaneous from the door reference frame, are not simultaneous from the spaceship reference frame. From the spaceship's viewpoint, the doorway is crossed at a nearly perpendicular angle so there is no contradiction between the 2 meter width of the doorway and ship's passage.

Th: Stop! More undergraduate physics! What is the point!

Sk: Merely that the proper way to look at things in physics, since Einstein's relativity, is not as events at a particular point in space and point in time, but as events at a particular point in space-time, and that I think a similar viewpoint will prove very useful in DIS. Actually, I think DIS may require parallel space-time universes as I was suggesting earlier. Each of the parallel space-times would have a different speed of influence tailored to a particular type of interaction between DIS entities.

Now if I just had some time to work out the details ...

Th: You will talk me out of some research funding yet.

VI. LIMITATIONS ON PHYSICAL SIMULATION

Thorpus: We have limited our discussion primarily to the simulation of vehicles and other man-made entities. I wonder about the simulation of natural phenomena. In particular, are we going to be able to represent electromagnetic propagation accurately in DIS? and can we accommodate environmental phenomena such as clouds, ocean currents, lightning, etc. in DIS?

Skepticus: Yes, and no. The individual simulators can incorporate accurate propagation models at the expense of "cray" capabilities. I suspect distributing the EM (or other field) on the DIS net would require as many nodes as there are degrees of freedom in the field, potentially an enormous number.

I have to answer the second question in the same way: "Yes and no".

Th: Come on. Let's have some more detail. I know you will tell me you need some time for more study, but surely you have some ideas.

Sk: As you may have noticed I have been falling back upon physical analogies, to relativity, to quantum mechanics in our earlier discussion, but I'm afraid that here physical analogy gets rather difficult. The phenomena you are asking about are physical, so the simulation is direct and is often the subject of on-going research by scientists in the relevant field, in some cases it is not yet known how to apply DIS, object-style, simulation to simulating the phenomena.

For example, if you want to simulate electromagnetic propagation and stay strictly within the DIS paradigm, the best approach is to model the propagation quantum mechanics style, rather than classical style. As I mentioned earlier, fields in quantum mechanics are model by interchange of virtual particles. Feynman invented a marvelous way to visualize the equations of quantum mechanics in the form of Feynman diagrams. In the case of wave propagation, the Feynman diagram would show photons corresponding to the radar or the laser beam or the radio waves traveling in straight lines in space-time until they scatter of other particles. Eventually the field photons reach the particles of the receiver where they are detected.

Th: Good. Then the photons could be DIS PDUs that are sent from the radar site, for example, are received by the target entity. The target then re-emits the photon PDUs which are received by the radar site entity where the target is detected or not according to algorithm.

Sk: Yes, a radar in a vacuum would work that way, but Feynman diagrams are more subtle yet. Consider the case where the radar wave propagates through an atmospheric inversion layer, which can deflect the beam or give false echoes. Within the Feynman formalism the inversion layer would be represented as a particle as well. In general the medium particles off which the propagating photons scattered are phonons, or particles of sound, since sound propagates by varying the mechanical properties of the medium such as density, temperature, pressure etc. A stationary inversion layer can be conveniently thought of as a stationary sound field, so that the radar wave is scattered from the stationary inversion layer phonons when classically it is refracted by the layer.

In principle then it would be possible to model physical phenomena with continuous spatial extent like inversion layers and the like by creating DIS entities that receive and re-transmit propagation PDUs in much the same way as the phonon-type particles of physics scatter the radiation photons.

With care the physics DIS entities would automatically obey the correct laws of motion as a result of their interaction with other physics entities through PDU traffic. In the case of DIS this is just a mathematical/computational convenience; real clouds do not send messages to other clouds resulting in dynamic interaction leading to weather. In quantum physics, of course, the world does appear to operate at a fundamental level through just such message passing.

VII. CONCLUSION

Th: Well this sounds very good. If I get some physicists laid off from that Super Collider project to work on the DIS team, you make it sound like physical phenomena can be modeled without going outside the DIS message passing dead-reckoning paradigm.

Sk: The success of quantum mechanics makes me think that in principle the DIS paradigm can handle physics, but I have some doubts about the practicality. In the microscopic realm, the world really seems to work by particles exchanging information via virtual particles. For the scale of physics concerned to the battlefield, however, the use of the Feynman formalism is more of a mathematical/computational convenience. The continuous distribution of water vapor etc. that constitutes the cloud or whatever is conceptually Fourier transformed so that the cloud is decomposed into a series of modes. Each mode is then treated as a particle which is instantiated as a DIS entity.

What concerns me is the number of entities or modes that would be needed to represent detailed features of the environment like clouds. If every mode requires an entity this could lead to the sorts of problems we discussed under the size. Granularity becomes an issue as well. The phenomena of chaos wherein systems evolving according to non-linear dynamics become extremely sensitivity to initial conditions could place very high demands on the precision of representation. Modern techniques for decomposing into modes such as the wavelet transform can help to reduce network traffic demands, but this is clearly an area that requires much study.

Th: What if the DIS paradigm is relaxed to allow servers?

Sk: If you had asked me that question at the beginning I would have told you that servers are clearly the way to go. A server can incorporate the latest thought on modeling environmental phenomena from the relevant scientific community, and act as the interconnection media between the DIS simulators. For the EM case the radar emits to a server that simulates the propagation through the environment including effects such as inversion layers, and the server then in turn communicates with the target. All the nasty details of trying to model the environment strictly within server-less DIS are avoided.

However, now that I see the possibility of building a server-less DIS that can model the environment by designed DIS in analogy to the techniques of quantum mechanics and relativity I am not so sure. There is a certain attractiveness to pure server-less DIS, although it looses some of its simplicity if parallel-DIS's with different speeds of influence are required. The big advantage of server-less DIS might be forward compatibility with future versions. If servers are introduced there will be a big discontinuity with past versions so that legacy systems will be orphaned since they will be incompatible with the servers.

That's a decision I will have to leave to the DIS Standards Workshops.

VIII. REFERENCES

Arrow, Kenneth Joseph (1951); Social choice and individual values; New York Wiley, 1951 99pp.

Brown, Robert Grover and Patrick Y.C. Hwang(1992); Introduction to random signals and applied Kalman filtering; New York : J. Wiley; 502 pp.

Clarke, Thomas L (Ed) (1995); Distributed Interactive Simulation Systems for the Simulation and Training in the Aerospace Environment ; SPIE, Bellingham, WA. 327pp.

Dahmann, Judith S. and David C. Wood (Eds) (1995); "Special Issue on Distributed Interactive Simulation," IEEE Proceedings, , Vol 83, pp 1111-1195.

Dewdney, A. K. (1989); The Turing omnibus : 61 excursions in computer science ; Rockville, MD : Computer Science Press; 415 pp.

Dreyfus, Hubert L., Dreyfus, Stuart E. and Athanasiou, Tom (1986); Mind over machine : the power of human intuition and expertise in the era of the computer ; New York : Free Press, 231pp.

Francis, Bruce A. and Allen R. Tannenbaum (Eds) (1995); Feedback Control Nonlinear Systems, and Complexity; Springer-Verlag, New York; 275 pp.

Geiger, Gebhard (1990); Evolutionary Instability: Logical and Material Aspects of a Unified Theory of Biosocial Evolution; Springer-Verlag, New York; 167 pp.

Hong-Mo, Chan, and Tsuo Sheung Tsun (1993); Some Elementary Gauge Theory Concepts; World Scientific, New Jersey; 156 pp.

Quinn, Michael J (1987); Designing Efficient Algorithms for Parallel Computers; McGraw-Hill, New York; 288pp.

Robertson, Robin, and Allan Combs (Eds) (1995); Chaos Theory in Psychology and the Life Sciences; Lawrence Erlbaum Associates, New Jersey; 399 pp.