Using Space-Time Constraints to Guide Model Interoperability

Paul F. Reynolds Jr.
Dept of Computer Science
Thornton Hall, 151 Engineers Way
The University of Virginia
Charlottesville, VA 22904
434-924-1039 (V) , 434-982-2214 (F)
reynolds@virginia.edu

Keywords:
Interoperability, Multi-Resolution Modeling, Motion Retargeting, Space-Time Constraints

ABSTRACT: *Space-Time Constraint optimization techniques have been used very effectively in the field of graphics animation for coordinating real and animated characters, including motion retargeting. Using a script from, say, a ballerina, animated characters (including Coke cans!) can be made to dance the script in a convincingly realistic manner. We argue that the same underlying technology can be employed to coerce interoperation between two or more component simulations, even if they are different resolution (i.e. multi-resolution) representations of the same phenomenon. We describe our approach as it applies to military operations other than war, in particular military operations in urban terrain (MOUT), where we use as our script the behavior of an individual simulated soldier, and then apply it to guide the behavior of a simulated special operations force. We argue that our approach is better than one that coerces the internals of component simulations to be consistent, because it involves less work, and it captures what is important to the end user. Also, we describe a process for creating a robust model, based on interoperating component models, using the space-time technology we describe here.*

1. Introduction

Simulation interoperability – involving construction of new simulations from a set of preconceived, and possibly implemented, "component" models requires the successful capture of critical component model semantics. We argue this capture need not be complete, in the sense that all aspects of component model semantics are represented. It suffices that the constructed simulation operate *as though* the semantics of the component models have been captured directly, and managed consistently. It is key to differentiate between guaranteeing consistency in the joint semantics of component models, and guaranteeing the appearance of doing so. The approach we advocate here, having explored the complete consistency approach and found it to be labor intensive, is to capture constraints that are important to the end user(s), and in doing so, to capture only those semantic consistencies that matter.

For exposition we consider three models: X, Y and Z. Our goal is to make X and Y interoperate in a manner that meets end user requirements. We refer to the satisfactory interoperation of X and Y as Z. If Z operates in a general and robust manner (Exact definitions of general and robust left to the reader. We mean to suggest a model that can operate acceptably over a broad set of scenarios, thus being more than a single point solution for a given scenario or context.) then we refer to the result as "robust Z." To put our terminology in context, we find that most researchers investigating interoperability are attempting to enable construction of robust Zs.

For more than a decade graphics animators have explored automated approaches to motion retargeting – accurately transferring human motions to animated characters. A typical goal is to transfer motions of a ballerina to an animated version of the ballerina, or even more challenging, an animated character with significantly different physical characteristics. In terms of X, Y and Z, graphics animators are attempting to use scripts from X, e.g. a ballerina, to coerce Y, e.g. an animated character, to become Z, an animated character whose dance is faithful to that of X, and whose physical characteristics are faithful to those of Y. The bulk of this kind of work has required a high degree of human intervention. However, recent automation successes have occurred, based on the employment of optimization techniques such as space-time constraints (STC). These successes offer significant promise for the simulation community.

To appreciate the potential of graphics animation technology, consider two simulations, X: a representation of an individual soldier (dismounted infantry – DI) in urban terrain, and Y: a special operations force in urban terrain We assume the latter is a force level simulator with less than completely accurate representation of individual members of the force. We assume the simulation of the DI represents a member of the force. Applying the techniques of graphics animators, we envision retargeting behaviors of the DI for a given scenario, say for an urban pursuit, onto the simulation of the SOF, producing simulation Z. This retargeting would occur in a manner that is consistent with constraints in the simulation of the individual, constraints in the simulation of the force, and constraints associated with physical laws. Ensuring consistency with constraints captured from X and Y, and with physical laws, is the product of employing STC, just as is done in the graphics community.

STC is only a mechanism for optimizing over a set of constraints. The environment we envision, in which a user can proceed from models X and Y, to production of an acceptable robust Z, includes solving the following problems: How do we represent constraints? How do we capture constraints from an existing simulation (e.g. what visualization and constraint capture tools can we employ)? The product of making X and Y interoperate for a given scenario is a Z that is not necessarily robust. It is most likely a point solution for the given scenario. What learning model should be employed to enable the construction of a robust Z – one that can be used in a general way across a broad set of scenarios?

We do not have answers to all of these questions. In the following we describe our approach to building robust simulations from component simulations, with a focus on military operations in urban terrain. (We choose MOUT because of the somewhat controlled nature of operations that we expect to occur, and because it involves a significant amount of entity level and near-entity level modeling, which is ideally suited for the technology we discuss.) We discuss why we have now rejected approaches that would maintain full semantic consistency between component models as a way of enforcing interoperability. We describe the efforts of the graphics retargeting community, and its focus on space time constraints. We discuss technical aspects of space-time constraints, and how STC fits into the environment we envision for producing robust interoperating simulations. Finally we describe open challenges.

2. Creating a Robust Z for MOUT

We describe the environment envisioned for constructing a robust simulation, robust Z, from two or more component simulations. A robust simulation meets its user's requirements for a broad range of scenarios. For exposition, we assume two component models exist, X and Y. We assume Z is derived from component simulation Y, with consideration for critical behaviors exhibited by X. Figure 1 depicts the process of creating Z, and later a robust Z.

We define USER as a person, or group of people, who possess qualifications such as the following: subject matter expert, end user of the combined models, component model designer or owner. USER is empowered to identify critical behaviors of component models X and Y and require that the combined model Z reflect these behaviors. Critical behaviors of X and Y are identified through a CAPTURE process associated with exercising and observing the models.

Representation of critical behaviors is an open issue. Required critical behaviors are in effect constraints, so the issue is actually about representing constraints. Possibilities include rule based systems, predict logic, and numerical equations. The motion retargeting community tends to employ numerical representations of constraints in order to exploit numerical optimization techniques. Further, these constraints are often embedded as code in component models, rather than as separate conditions imposed on an otherwise unmodified version of the component model. Whether the approach taken by the retargeting community is the approach most advisable for those simulating MOUT remains an open question.

Apart from constraints representing critical component model behaviors, context dependent constraints must also be captured. The most notable of such constraints are basic physical laws, such as those regarding gravity, human motion, interaction of physical materials and the like. Which of these laws must be captured as constraints is model dependent. A process must exist for capturing constraints that exist outside of the component models. Borrowing from the motion retargeting community, we call these "meta-constraints."

Given component model constraints, meta-constraints, a base model (presumably Y) from which to build the target model (Z) and a scenario used to assist in identifying component model constraints, Z can be constructed.

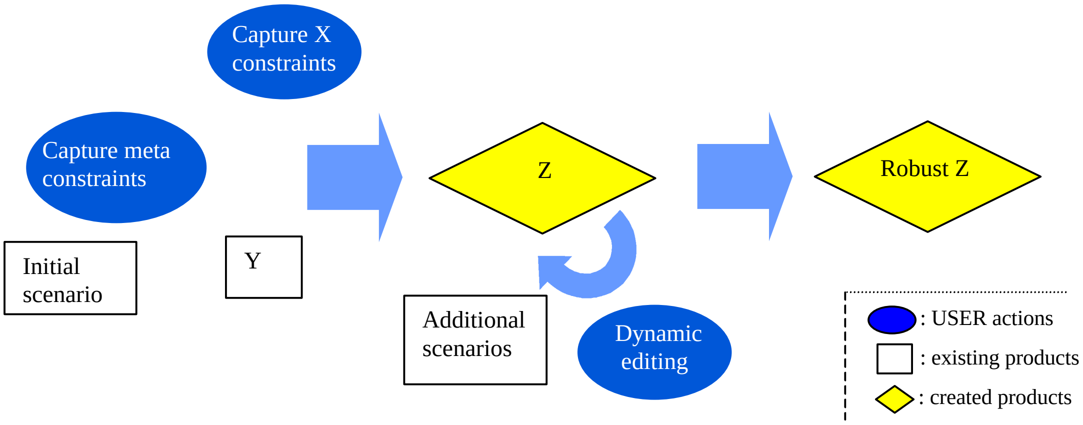

Figure 1: Steps to developing a robust model from component models, scenarios and metaconstraints.

Creation of a robust Z, a model that can operate on the full set of scenarios anticipated by USER, represents a significant challenge. Ideally a robust Z would emerge from the establishment of model constraints for a small set of scenarios. The expectation that general, robust behavior can emerge from a small set of examples has been the linchpin in technologies such as neural networks, artificial life, and computer learning models. With no comment about the viability of these technologies, we acknowledge that generalized learning is an important aspect of the path we envision to constructing a robust Z.

Even if construction of robust Zs proves to be difficult, identifying technology that allows construction of scenario-specific Zs, as is currently done in the motion retargeting community, would be a significant step forward for the simulation interoperability community.

For MOUT captured constraints would relate to position, motion, posture, and (simple) interactions. The dynamic editor would allow a user to modify the behaviors exhibited by the SOF based on constraints derived from properties of the DI. The optimization problem would be to position, move, posture and script the SOF in accordance with all of the physical and DI constraints.

A key challenge is to capture the constraints present in the equivalent of the SOF and DI simulators. For the motion retargeting world the constraints are, as noted above, about physical requirements and limitations. For the MOUT world, they would include doctrine, motion and interaction planning, and overall mission. Ours is a much larger problem than that encountered in motion retargeting because of the possible breadth of constraints.

3. Maintaining Consistency

Interoperability of MOUT simulations representing overlapping phenomena (individual soldiers –DIs– and SOFs operating in urban terrain) is an instance of the multi-resolution modeling (MRM) problem, a key challenge for simulationists. MRM has been studied extensively, including work conducted by Davis [Da92], [Da93], [Da98], Hillestad [Hi92], Petty [Pe95], Natrajan [Na95], [Na96], [Na97], [Na99], [Na00] and Reynolds [Re97]. The bulk of this work attempts to maintain consistency at all times in the execution of an MRM federation. Petty's work on aggregation-disaggregation relaxes the consistency requirement but sacrifices accuracy and performance for convenience and economy. In Reynolds [Re97] it is argued that aggregation-disaggregation is so riddled with problems that it is not worth pursuing. In [Na99] the Unify approach to MRM is introduced. Unify defines a procedure for guaranteeing consistency in multiple levels of resolution at all times, an approach we now believe is overly restrictive, and labor intensive.

Experiments on MRM have been conducted mainly within DoD, and include linkages such as Eagle/BDS-D, Abacus/ModSAF and AIM. These experiments typically employed playboxes or aggregation-disaggregation. As we have discussed in [Re97] these approaches have significant problems that can prevent them from being acceptably accurate.

Natrajan's Unify approach [Na99] to maintaining consistency between multiple models requires the identification of objects, object attributes, events and interactions within and between models. Events and interactions are characterized as actions that can impact attribute values, and can lead to inconsistencies of concurrent representations of modeled entities or phenomena. An *interaction resolver* is employed to ensure that concurrent interactions are modeled in a manner that does not violate consistency between component models. A recipe for consistent MRM appears in [Na00].

The issue with consistency maintenance is its labor intensive nature. We have found that asking USER to capture all objects, object attributes and interactions, and to provide all mapping functions between attributes representing different instances of the same phenomenon represents more effort than USER is willing to devote. This despite the alternative that USER not have any MRM capability at all.

The STC-based work of the motion retargeting community represents an inviting alternative. Under circumstances similar to those experienced by the MRM community, motion retargeting researchers have created animations *consistent with expectations* by identifying and meeting the critical *observable* conditions. They all but ignore underlying consistency issues.

4. Motion Retargeting and STC

Control of animated characters using space-time constraints began with Witkin and Kass [Wi88] and has remained an active area of research in the graphics community since that time. While space-time and motion-retargeting are independent technologies, we present them together here because it is the STC-based retargeting work of Gleicher that has made application of STC to interoperability evident.

4.1 Motion Retargeting

Recently noteable results have been reported by Gleicher [Gl97], [Gl98], [Gl00a], [Gl00b] in the area of motion retargeting. Gleicher's motion retargeting begins with a motion script for an animated character. Our equivalent would be the behavior (motion, posture, and the like) of the individual soldier. Given a motion script for an animated character, Gleicher then uses an editor to describe critical constraints including points in three space through which a second animated character must pass. (We envision a similar dynamic editor that defines critical motion paths, postures, and interactions, although not necessarily visually, as done in the graphics community). Given the new animated character, with a motion script and a defined path, Gleicher's technology retargets the script to the new character, accounting for all of the differences in the character's physique, critical physical laws, and the new path. In our MOUT example, the SOF's positions, motions, postures and interactions would be retargeted to the SOF, while respecting all constraints present in the simulation of the individual soldier. In a sense Gleicher has done multi-resolution modeling in the graphics world; consider that the two characters dancing the same dance could be easily scripted to dance together. Viewing a simply featured character (such as a Coke can) as an abstraction of the original character (one with limbs), the dance would be a form of multi-resolution modeling.

Gleicher's retargeting technology uses constrained non-linear optimization. The constraints reflect conditions such as: admissible parameter values, fixed locations at fixed times, images must stay in certain regions, two points on a character remain a specified distance apart, and others. Constraints must be invariant of other constraints already established.

Given a set of constraints, the challenge then is to establish an objective function, since the constraints alone often admit an infinite set of solutions. For the graphics world an objective may be to "minimize the amount of noticeable change." [GL98]. The motion retargeting method is then a procedure in which an initial position with a given set of constraints is determined. Then motion is scaled for the target character, yielding a motion displacement curve. Finally, a non-linear constraint solver is applied to determine a constraint satisfying motion.

Gelicher observes: "Because the space-time approach looks at the entire motion, it can make choices based on other parts of the motion. For example, it can move footplants based on where the character needs to end up. Such look-ahead and -behind is not possible in approaches that consider each frame independently." Similarly for us, with a scripted mission for an SOF, we can optimize over segments of a mission, using previous and future constraints as guides.

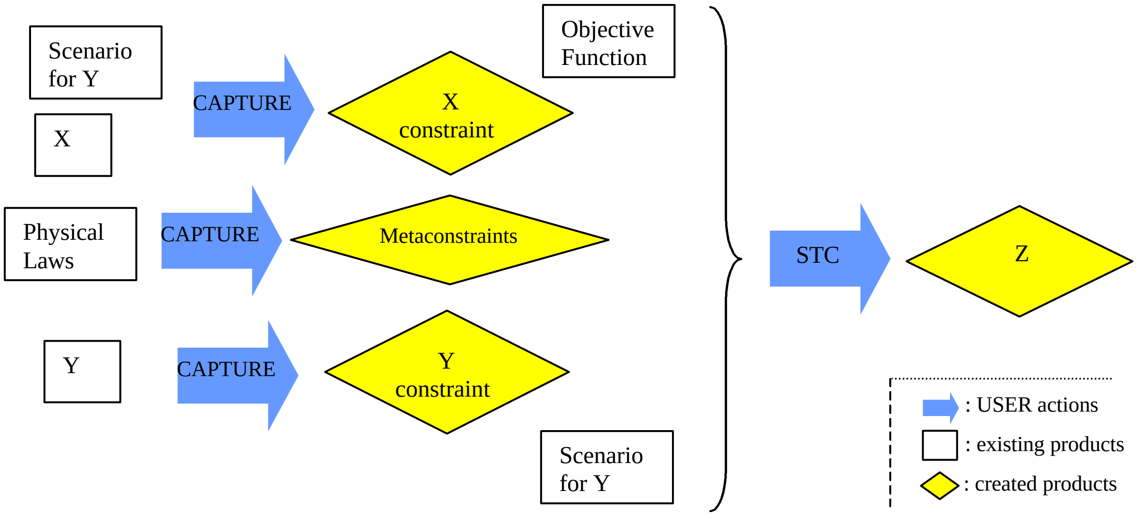

Figure 2: Amplifying on the Role of STC

4.2 Space-Time Constraints

Space-time constraints, introduced by Witkin and Kass in 1988 [Wi88], is a non-linear optimization technique that has been used to demonstrate that acceptable animations can follow from the application of constraints to physical motion. Their technique was to discretize critical aspects of animated characters, represent physical limits and laws so as to constrain the animated characters, and then treat the determination of an acceptable path through a set of possible motions as an optimization problem. Often such a problem is underconstrained, so an objective function is identified, such as the minimization of effort, or minimizing the length of a path. The optimization technique used in space-time constraints is typically a greedy algorithm. The problem is treated as a two point (beginning and end) boundary value problem.

The product of an application of STC to animation is a solution that fits its environment exactly. That is, it is a single point solution for the scenario (e.g. dance script) it was given as part of its constraint set. Generalizing the approach taken to using STC in the graphics community is one of the challenges we face in our proposed approach.

5. Robust Z, MOUT and STC

As we have noted, the capability to create a robust Z, as depicted in Figure 1, will require an ability to generalize learning from a fixed set of scenarios. We have no firm insights about how to address this challenge. Traditional learning methods in AI, neural net learning models and emergent behaviors in artificial life seem good candidates. In the meantime we find the application of STC to developing a scenario-dependent Z a sufficiently rewarding goal, as we expect USER could benefit from reuse of previously identified constraints when developing different Zs for different scenarios. Developing a method to derive Z represents a significant contribution to the modeling community.

We have chosen MOUT as our target application partly because MOUT is characterized by a certain degree of doctrinal behavior. Since doctrine lends itself to representation by constraints, we view it as a convenient simplifier as we investigate development of the technology described here. Of course, the MOUT training, analysis and planning communities will benefit significantly from success in our endeavors.

Our approach to deriving a Z, as depicted in figure 1, requires the capture of constraints related to component model X. For example, assuming X represents the motion behavior of a DI, USER may determine that X’s modeling of a soldier’s response to a perceived mine is critical in Y’s simulation of an SOF. “Reaction to a mine” then represents a set of constraints imposed on the Z derived from one or more of the component models. (Our stated assumption is that Z is derived from Y – an SOF simulation.) So, whenever simulation Z models an SOF approaching a mine, its modeled behavior of individual DIs will accurately reflect how X would have modeled their behavior. If the reaction of adjacent DI’s to the behavior of the DI avoiding the mine is important to USER, then that reaction must have come into Z either because it was already represented in the model from which it is derived, Y, or by way of additional constraints placed on individual DI behavior, possibly through application of X to alternative scenarios.

Certainly USER must consider constraints carefully. USER will have the benefit of observing Z applied to scenarios used to derive constraints from X, so any behavior violating as yet uncaptured constraints could be observed. (Clearly, even developing Z will be an iterative process.)

In figure 2 we have amplified details of the process associated with creating Z in order to show more clearly the role that STC plays in Z's creation. The CAPTURE of X's constraints in the context of a scenario for Y yields a key set of constraints for shaping Z. X's constraints along with a set of metaconstraints define the bounds in which Z, derived from Y, must operate. As depicted in figure 2, the sum of the constraints used to shape Z's design include constraints derived from X, the metaconstraints, constraints associated with simulation Y and the scenario used to capture X's constraints. Coupled with an objective function, say, minimizing friendly force fatalities, STC can be employed to produce the simulation Z that represents SOF simulation Y operating in conformance with constraints imposed by DI simulator X.

There are many challenges associated with creating simulation Z, a MOUT simulation representing the interoperation of component MOUT simulations X and Y. We discuss these next.

6. Challenges

Figures 1 and 2 represent the design of our STC-based approach to interoperating simulations. The product, simulation Z (and later, robust Z), is a scenario-dependent model that properly represents the critical behaviors (as captured in constraints) imposed on it by a knowledgeable USER. The process of constructing Z represents a first effort at applying STC to interoperating simulations (IS). Robust Z is an ideal model, capable of accurately (by USER's standards) representing intended phenomena (in our case, a MOUT SOF) across a broad set of scenarios. The challenges we face as we migrate motion retargeting (MT) technology to the universe of interoperable simulations include the following.

The CAPTURE process conducted in the MT community is considerably simpler than we anticipate it will be for the IS community. MT is focused on graphics animation, so many of their tools can build on already existing graphics interfaces. The MOUT and IS communities have fewer such tools. The generalized visualization and CAPTURE processes for the IS community, even for the limited domain of MOUT, will be an interesting research area in its own right.

MT's Z often incorporates, as code, the scenario used to capture constraints from X. We'd prefer to maintain code independence with respect to scenarios so that learning, for the purpose of generating a robust Z, can take place in a scenario independent manner.

The interaction between X and Z, as we have described it here, is off-line, meaning we are not proposing an equivalent of X and Y interoperating dynamically, perhaps in real time. Instead, our approach is to construct Z, and then use Z to represent interoperating X and Y for a given scenario. Later, a robust Z would represent an interacting X and Y for an arbitrary scenario. Ideally, we would be able to accept an arbitrary scenario, an arbitrary X and arbitrary Y and produce the equivalent of a Z in real time. Therein lies the challenge, including establishing feasibility of the idea.

The MT community has traditionally represented constraints as code in the derived (e.g. Z) model, in the sense represented constraints are relatively indistinguishable from other code capturing the character's behavior. We see this as undesirable because the embedding would likely require USER to have assistance with or knowledge of the implementations of the models X, Y and Z. We have as a goal the independent representation of constraints – independent of the code representing Z. We expect interactions between the two when Z executes, but in a manner that allows their separate representation.

Finally, as we have noted previously, MT doesn't employ learning or generalization (no notion of robust Z). Their successes are point solutions (for given scripts). This is understandable, as simply coming up with the point solutions has been a significant challenge in its own right. They, and we, will ultimately seek generalized solutions, theirs in the form of "Make this animated character dance this dance in the manner Barishnikov would" and ours in the form "Make this SOF perform this MOUT mission consistent with the behaviors of individual DIs."

7. Conclusions

Motion retargeting researchers in the graphics animation community have evolved a technology, STC-based motion retargeting, that appears to be applicable in a generalized sense to the interoperating simulations and multi-resolution modeling communities. In particular, their emphasis on the capture of important requirements imposed by a knowledgeable user, as opposed to blind requirements that all aspects of component models be maintained consistently, may be a key insight for the interoperating simulations community. Through the employment of a technology called space-time constraints, the animation community has successfully captured user requirements as constraints which themselves are used to guide the discovery of an optimal path through space and time for a simulated character. Optimality occurs in the form of e.g. minimizing energy; in the MOUT simulation community it could be minimizing friendly force fatalities.

We have described our approach to transferring the STC-based technology used by graphics animators to the interoperable simulations community, noting that the technology applies equally well to multi-resolution simulations. There are many interesting challenges associated with the proposed transfer, which, many of which we have described here.

7. References

[Da92] P.K. Davis, “An Introduction to Variable-Resolution Modeling and Cross-Resolution Model Connection, *Conference on Variable-Resolution Modeling*, Wash, DC, May 1992.

[Da93] P.K. Davis and R.J. Hillestad, “Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Management,” *Winter Sim Conference*, 1993.

[Da98] P.K. Davis and J.H. Bigelow, *Experiments in Multiresolution Modeling*, RAND, 1998.

[Gl94] M.L. Gleicher, *A Differential Approach to Graphical Interaction*, Ph.D. Thesis, School of Computer Science, Carnegie Mellon, 1994.

[Gl97] M. Gleicher, “Motion Editing with Spacetime in M. Cohern and D. Zeltzer, editors, *1997 Symposium on Interactive 3D Graphics*, pp. 139-148, April 1997.

[Gl98] M. Gleicher, “Retargetting Motion to New Characters,” SIGGRACH 1998.

[Gl99] M. Gleicher and P. Litwinowicz, “Constraint-based Motion Adaptation,” *Journal of Visualization and Computer Animation*, 1999.

[GL00a] M. Gleicher, “Comparative analysis of Constraint-based Motion Editing Methods, *2000 Workshop on Human Modeling and Animation*, Seoul Korea. June 2000.

[GL00b] M. Gleicher, “Motion Path Editing,” 2001 Symposium on Interactive 3D Graphics. ACM Press, pp 195-202.

[Hi92] R.J. Hillestad and M.L. Juncosa, “Cutting Some Trees to See the Forest: on Aggregation and Disaggregation in Combat Models,” *Conference on Variable-Resolution Modeling*, Washington, DC, May 1992.

[Ho92] T.J. Horrigan, “The ‘Configuration Problem’ and Challenges for Aggregation,” *Conference on Variable-Resolution Modeling*, Washington, DC, May 1992.

[Na95] A. Natrajan, “To Disaggregate or Not to Disaggregate, that is the Question,” *ELECSIM*, Internet, April-June, 1995.

[Na96] A. Natrajan, P.F. Reynolds, Jr., and S. Srinivasan, *Consistency Maintenance using UNIFY*, Tech. Rep. CS-95-28, Dept. of Computer Science, Univ of Virginia, 1996.

[Na97] A. Natrajan, P.F. Reynolds, Jr., and S. Srinivasan, “A Flexible Approach to Multi-Resolution Modeling, Parallel and Distributed Simulation, *Parallel and Distributed Simulation*, June, 1997.

[Na99] A. Natrajan, and P.F. Reynolds, Jr., “Resolving Concurrent interactions,” *3rd International Workshop on Distributed Interactive Simulation and Real Time Applications*, October, 1999.

[Na00] A. Natrajan, *Consistency Maintenance in Concurrent Representation*, Ph.D. Thesis, Dept. of Computer Science, University of Virginia, January 2000.

[Pe95] M.D. Petty and R.W. Franceschini, “Disaggregation Overload and Spreading Disaggregation in Constructive+Virtual Linkages,” *5th Conference on Computer Generated Forces & Behavioral Representation*, Orlando, Florida, May 1995.

[Re97] P.F. Reynolds, Jr., A. Natrajan, and S. Srinivasan, “Consistency Maintenance in Multiresolution Simulations,” *ACM Trans. on Modeling and Computer Simulation*, vol. 7, no. 3, July 1997, pp. 368-392.

[Wi88] A. Witkin and M. Kass, “Spacetime Constraints, in J. Dill, editor, *Computer Graphics (SIGGRAPH ’88 Proceedings)*, vol. 22, pp. 159-168, August 1988.

Acknowledgements

This research is sponsored in part by the Defense Modeling and Simulation Office, Dr. Phil Barry.

Author Biography

Paul F. Reynolds, Jr. is a Professor of Computer Science at the University of Virginia. His research has traditionally focused on parallel and distributed systems. He has written oft-cited papers on the topics of time management in and linkage technologies for parallel and distributed simulations. He was an original member of the design team for the DoD High Level Architecture for modeling and simulation. He has done seminal work on multi-resolution modeling. He was a co-discoverer of Isotach Networks.