CHRISTOPHER D. CAROTHERS

Work Address
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332
(404)-894-6858
email: chrisc@cc.gatech.edu
webpage: www.cc.gatech.edu/grads/c/Chris.Carothers/homepage.html

Home Address
136 Carlyle Lake
Decatur, Georgia 30033
(404)-634-7597

Research Interests

- Experimental distributed systems.
- Modeling and simulation.
- Wireless communications systems (PCS).

Education

Ph.D., Computer Science.
Georgia Institute of Technology, September 1997.

M.S., Computer Science.
Georgia Institute of Technology, December 1996.

B.S., Information and Computer Science.
Georgia Institute of Technology, 1991.

Professional Experience

Research Scientist, Georgia Tech — Atlanta, Georgia
January, 1997 – Present.

Member of Technical Staff, MITRE Corporation — Reston, Virginia
June, 1996 – September, 1996.

Member of Technical Staff, Bellcore — Morristown, New Jersey
June, 1993–October, 1993.
June, 1994–September, 1994.

Research Assistant, Georgia Tech Research Institute — Atlanta, Georgia
January, 1992–March, 1993.

Ph.D. Dissertation

- **Thesis Title:** *Efficient Execution of Time Warp Programs on Heterogeneous, NOW Platforms*.
- **Advisor:** Richard M. Fujimoto
- My thesis develops a novel load distribution algorithm that enables Time Warp programs to efficiently execute on heterogeneous, NOW environments in the presence of externally-induced workloads as well as irregular internally-induced workloads (i.e., workloads generated by the application model).

Honors and Awards

- **MITRE Program Recognition Award** for contributions on the DoD High Level Architecture (HLA) project.
- **Sigma Phi Sigma**, Physics Honor Society, Furman University.

Publications

1. **C. D. Carothers**, Brad Topol, R. M. Fujimoto, John T. Stasko, and Vaidy Sunderam, "Visualizing Parallel Simulations in Network Computing Environments", (conference version), *Proceedings of the 1997 Winter Simulation Conference*.
2. **C. D. Carothers**, R. M. Fujimoto, R. M. Weatherly, and A. L. Wilson. "Design and Implementation of HLA Time Management in the RTI version F.0", *Proceedings of the 1997 Winter Simulation Conference*.
3. **C. D. Carothers**, Brad Topol, R. M. Fujimoto, John T. Stasko, and Vaidy Sunderam, "Middleware-Specific Visualization Support for Parallel Simulations in Cluster Environments", (extended abstract). *1997 Cluster Computing Conference (CCC'97)*.
4. **C. D. Carothers**, and R. M. Fujimoto, "Background Execution of Time Warp Programs", *Proceedings of the 10th Workshop on Parallel and Distributed Simulation (PADS '96)*, May 1996.
5. **C. D. Carothers**, and R. M. Fujimoto, "Distributed Simulation: An Effective Modeling Tool for Large-Scale PCS Networks", To appear in the *International Journal in Computer Simulation*.
6. **C. D. Carothers**, R. M. Fujimoto and Y-B. Lin, "Simulating Population Dependent PCS Network Models Using Time Warp", Invited Paper, *Proceedings of the 1995 Winter Simulation Conference (WSC '95)*, December 1995, pages 555–562.
7. **C. D. Carothers**, R. M. Fujimoto and Y-B. Lin, "A Re-dial Model for Personal Communications Services Networks", *Proceedings of the IEEE 45th Vehicular Technology Conference" (VTC '95)*, July 1995, pages 135–139.
8. **C. D. Carothers**, R. M. Fujimoto and Y-B. Lin, "A Case Study in Simulating PCS Networks Using Time Warp", *Proceedings of the 9th Workshop on Parallel and Distributed Simulation (PADS '95)*, June 1995, pages 87–94.
9. **C. D. Carothers**, R. M. Fujimoto and P. England, "The Effect of Communication Overheads on Time Warp Performance: An Experimental Study", *Proceedings of the 8th Workshop on Parallel and Distributed Simulation (PADS '94)* , July, 1994, pages 118–125.
10. **C. D. Carothers**, R. M Fujimoto, Y-B Lin and P. England, "Distributed Simulation of Large-Scale PCS Networks", Invited Paper, *Proceedings of the Second International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, February 1994, pages 2–6.

Publications Under Review

1. **C. D. Carothers**, Brad Topol, R. M. Fujimoto, John T. Stasko, and Vaidy Sunderam, "Visualizing Parallel Simulations in Network Computing Environments" (journal version), submitted to *the Future Generations of Computer Systems (FGCS)*.
2. **C. D. Carothers**, and R. M. Fujimoto, "Efficient Execution Time Warp Programs on Heterogeneous, NOW Platforms", submitted to *IEEE Transactions on Parallel and Distributed Systems (TPDS)*.

Professional Activities

Guest Editor, SCS SIMULATION/Transactions Joint Special-Issue on Parallel and Distributed Simulation to appear in Summer, 1998.

Reviewer, *ACM Computing Reviews*.

Referee:

- *ACM Transactions on Modeling and Simulation (TOMACS)*
- *IEEE Transactions on Parallel and Distributed Systems (TPDS)*
- *Workshop on Parallel and Distributed Simulation (PADS)*
- *International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*
- *Winter Simulation Conference (WSC)*
- *Mobile Computing Conference (MCC)*
- *The Hawaii International Conference on System Sciences (HICSS)*

Student Member, 1996-1997 CoC Faculty Hiring Committee

Member, Georgia Tech Executive Round Table (ERT)

Student Member, The Association for the Computing Machinery (ACM)

References

Richard M. Fujimoto (advisor)
801 Atlantic Drive
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, 30332
e-mail: `fujimoto@cc.gatech.edu`
phone: +44-1684-896793 (on Sabbatical)

Andrew Ogielski
Rutgers University
DIMACS Center CoRE Building, 4th Floor
96 Frelinghuysen Road
Piscataway, New Jersey 08854-8018
e-mail: `ato@liman.rutgers.edu`
phone: (732) 445-5285

Mustaque Ahamad
801 Atlantic Drive
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, 30332
e-mail: `mustaq@cc.gatech.edu`
phone: (404) 894-2593

Yi-Bing (Jason) Lin
Department of Computer Science &
Information Engineering
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.
e-mail: `liny@csie.nctu.edu.tw`
phone: +886-3-5731842

Vaidy Sunderam
Department of Mathematics &
Computer Science
Emory University
Atlanta, Georgia 30322
e-mail: `vss@mathcs.emory.edu`
phone: (404) 727-5926

Summary of Research Activities

Christopher D. Carothers

Webster's Dictionary defines simulation as the *act of pretending or creating a falsehood*. In order to turn these "lies" into "truths", modelers must ensure that the simulation model correctly reproduces the behavior of the system being modeled. As the system under study becomes arbitrarily complex, so does the simulation model. The caveat here is that despite the continuing advances being made in processor speed, complex simulation models consistently require more CPU cycles than the state-of-the-art can provide. Today, many simulation applications require days or weeks of CPU time on the fastest uniprocessor machines. The primary research objective of the parallel and distributed simulation community is to reduce the execution time of these large and complex simulations by employing many computers or processors simultaneously.

A critical issue that must be addressed when executing a discrete-event simulation program on parallel or distributed computers is ensuring that the program is properly synchronized. The distributed simulator consists of a collection of *logical processes* or LPs, each modeling a distinct component of the system being modeled, e.g., a server in a queuing network. LPs communicate by exchanging timestamped event messages, e.g., denoting the arrival of a new job at that server. The synchronization mechanism must ensure that each LP processes events in timestamp order and prevent events in the simulated future from affecting those in the past.

In general, there are two approaches for addressing the synchronization problem. *Conservative* approaches ensure that events are processed in timestamp order by *waiting* until it is "safe" to process each event (i.e., no smaller timestamped events will be received later). *Optimistic* approaches allow events to be processed out of order, but provide a mechanism to detect and erase these incorrect event computations. The seminal optimistic approach is the Time Warp protocol developed by David Jefferson.

My research focuses on Time Warp simulation systems for distributed computing environments. This focus is divided into the following research thrust areas: (i) *Load Balancing for Time Warp Systems*, (ii) *Modeling Methodologies for Time Warp Systems*, (iii) *Visualization Support for Time Warp Systems*, (iv) *Wireless (PCS) Networks* and, (v) the *DoD High Level Architecture*.

Load Balancing

Time Warp is an optimistic protocol for synchronizing parallel and distributed discrete-event simulations. To achieve performance, this protocol typically requires exclusive processor use. If external computations from other users are present, a Time Warp program may become grossly unbalanced, resulting in degraded performance. While this "dedicated platform" paradigm makes perfect sense from an experimental standpoint, it is often not the computing environment one encounters in practice. In particular, multi-user networks of workstations (NOWs) interconnected through LANs have become prevalent.

Moreover, this multi-user computing environment is typically composed of heterogeneous workstations that contain different processors. These processors may have quite different performance characteristics that, if not taken into consideration, can lead to poor Time Warp performance.

In addition to performance-robbing external workloads and processor heterogeneity, the application model itself can be a source of perturbation. In many simulation models, the amount of CPU time required to process an event varies among *logical processes (LPs)*, and the event population may differ across LPs, both of which can degrade performance.

The key observation asserted in my dissertation is that each of these performance problems, while different in source, has a similar manifestation. For a Time Warp program to be balanced, the amount of wall-clock time necessary to advance an LP one unit of simulation time should be about the same for all LPs. Using this observation, we devise a single, unifying algorithm that mitigates these performance problems and enables the "background" execution of Time Warp programs on heterogeneous distributed computing platforms in the presence of external as well as irregular internal workloads.

Using our load distribution algorithm we observed performance improvements (i.e, reduction in execution time when compared to Time Warp execution without our load distribution algorithm) of 127% for internal workloads, 260% for static external workloads, 30% for time-varying external workloads, 100% for heterogeneous processor configurations, and 100% for combination workloads (i.e., time-varying external and static internal workload).

Modeling Methodologies for Time Warp Systems

One of the primary criticisms of parallel and distributed simulation techniques is the high degree of effort required by application developers to realize a parallel simulation model that yields acceptable performance. In order to reduce programming effort, we need to understand how the parameters within simulation models affect performance and then extrapolate those parameters into a set of general modeling abstractions that can be used in a variety of application domains. To this end, I have been aggressively investigating and developing optimistic simulation models of Personal Communication Services (PCS) networks that achieve good performance results on distributed and parallel computing platforms as a means of understanding the modeling factors that effect Time Warp performance.

On a distributed computing environment, we have observed speedups of 7.5 on 8 processors, enabling simulation runs that require 20 hours on a single workstation to be completed in only 3.5 hours. On a shared memory multiprocessor KSR system, we observed almost linear speedup on 49 processors or fewer processors.

Visualization Support for Time Warp Systems

One of the difficult tasks for Time Warp system builders is understanding how different aspects of a simulation model affect Time Warp system performance, particularly in network computing environments. To this end we investigated the use of a visualization system that casts model-specific behaviors into Time Warp specific behaviors, enabling system builders to immediately observe cause and effect relationships between key Time Warp system performance parameters.

To develop our Time Warp visualization system, we modified an existing general-purpose, Graphical Visualization (GV) system, called *PVaniM* and used it to conduct a series of visualization experiments on an existing Time Warp system, called *Georgia Tech Time Warp (GTW)* in a network computing environment. PVaniM was chosen because of its ease of use, low system perturbation, and robustness. However, PVaniM's default graphical views did not provide enough insight into the execution of the GTW system. This was due to the fact that all PDES systems are essentially middleware that support their own applications. We define PDES systems as middleware because they typically reside above the operating system and a communication substrate (e.g., PVM) yet support their own applications. We contend that complex network computing middleware, such as Time Warp systems, requires its own *middleware-specific* visualization support. Middleware-specific views focus on illustrating the operation of PDES middleware. They may be used with all simulation applications and allow the user to see how various applications perform on the PDES middleware. It is worth noting that middleware-specific views are not simply a set of application-specific views. Application-specific visualizations for parallel simulations refer to customized views whose appearance is tied to simulation model data. These types of views are typically provided by visual interactive simulation systems.

The primary contribution of this work is that by enhancing a general-purpose network computing visualization system with middleware-specific views, one is able to better satisfy the requirements and needs of parallel discrete-event simulation (PDES) middleware. Specifically, we document how we have used a GV system to locate performance aberrations, and gained greater insight into how to optimize an existing Time Warp system for a heterogeneous network computing environment.

Wireless (PCS) Networks

Using the PCS models I developed for our distributed Time Warp system, two network performance analysis studies were conducted in collaboration with researchers at Bellcore. The first case study examines the effect of re-dial in a personal communications services (PCS) network. The re-dialing problem for PCS is different from the plain old telephone service (POTS). When using POTS, if a customer receives a busy signal in making a call, or the call does connect in a timely manner, the customer may make a re-dial call in an attempt to get a connection. Likewise, it is anticipated PCS network subscribers may also make re-dial attempts when the original call attempt is unsuccessful. However, unlike POTS, PCS has the added complexity of keeping a call connected while the originating caller and/or the terminating caller is moving. Because of this complexity, the effect of re-dial on nomadic communications is a challenging problem that has implications on the design and analysis of PCS networks. Accordingly, this paper presents a model for studying the effect of re-dial on PCS networks. Using this model, several analytical results are presented. These results are then used to validate a Time Warp-based, distributed simulation of the PCS re-dial model. Using the distributed simulation, we vary (i) the probability of re-dial, (ii) the re-dial rate, and (iii) portable mobility, and examined each parameters effect on (a) the probability of forced termination, (b) the probability of new call block (c) the call incompletion probability (d) expected number of re-dials before a new call attempt is accepted or rejected and (e) calls accepted per simulated second. For both studies, $1/\mu$ is defined to be the mean call holding time. The primary results from this study were:

- Re-dialing, when done at a low rate ($< 1.0\mu$) and high probability (> 0.5) helps to reduce the call incompletion probability, p_{nc}.
- Re-dialing, when done with a high probability ($\alpha > 0.5$) can cause an increase in the probability of force terminations, p_f.
- Our study indicates that re-dialing may improve the call incompletion probability in the workload range of interest. However, from the caller's view point, frequent re-dialing implies a poor quality of service. Thus, the PCS designer should carefully select the radio port capacity that provides a POTS level of service quality to PCS subscribers, as well as limit the frequency and number of re-dial attempts.
- We expect that re-dialing will have a negative impact on PCS networks with a heavy offered load or high mobility of subscribers. However, in the parameter ranges of interest, we do not see the negative impact. Accordingly, this will simplify the design tradeoffs.

In the second case-study, the existing PCS models are modified to account account for population-dependent mobility patterns. This study analyzes the effect of these kinds of mobility patterns on network performance. To generate different mobility patterns, the mobility distribution function was varied across 5 different probability distributions: (i) exponential, (ii) Pareto with α equal to 1.6, (iii) Pareto with α equal to 5.0, (iv) Pareto with α equal to 100.0, and (v) uniform. From this study, the primary results were:

- The Pareto distribution for medium mobility rates ($0.75\mu < \eta < 2.50\mu$) appears to yield significantly higher PCS blocking statistics than the exponential, especially when the Pareto distribution is deterministic ($\alpha = 100.0$).
- For high mobility rates ($\eta > 5.0\mu$), PCS blocking statistics appear to decrease. However, because only a few higher mobility rates are covered in this study ($\eta = 5.0\mu$, 10.0μ, and 15.0μ), it is possible that PCS blocking statistics cycle up and down as the mobility is increased. Further study is needed.

DoD High Level Architecture

To address the need for interoperability between new and existing U.S. Department of Defense simulations, the Defense Modeling and Simulation Office (DMSO) commissioned the High Level Architecture initiative. The HLA consists of 3 parts: (1) HLA rules, (2) HLA Interface Specification (IFSpec), and (3) Object Model Template. The software that meets the interface requirements set forth by the IFSpec is called the Runtime Infrastructure (RTI). For more information on the HLA and its parts, the reader is referred to the DMSO web-page at `http://www.dmso.mil`.

While at MITRE I served as the principal developer for implementing the Time Management services in first version of the RTI, dubbed F.0. To achieve the HLA's interoperability requirements, time management *must* appear transparent to all federates (i.e., a single simulator that is part of a federation of simulators) engaged in a given federation. This transparency is needed to facilitate federate integration with a federation as well as keep federate developers from having to understand the intricacies of distributed systems programming, thus reducing development efforts and costs.

Supporting this level of transparency presented a number of challenges for the Time Management design team that had to be overcome. In particular, the RTI must allow for: (i) federates that may or may not be constrained by a time-flow mechanism, (ii) constrained federates with different and dynamically changing time-flow mechanisms (e.g., event driven or time-stepped federates or those that switch between these two during execution), (iii) federates executing on parallel/distributed platforms using a conservative or optimistic synchronization protocol including those that dynamically change between them, (iv) federates with different event ordering requirements (e.g., receive or time stamp order), (v) federates using a mixture of event orderings and transportation services (e.g., reliable or best effort), and (vi) dynamically changing federations where federates can join and resign from a federation at any time.

Future Research Activities

Christopher D. Carothers

Below is a sample of the research avenues I wish to explore in the future. However, because simulation by its very nature is a "tool" that is to be used by others, I fully expect that many of my research directions will be driven by collaborations with students and colleagues from both inside and outside the systems group as well as from other science and engineering disciplines.

Stability Assurance Mechanism for Distributed Simulations

When viewed from a higher level, Time Warp load management techniques, such as the one discussed previously, are Stability Assurance Mechanisms (SAM), which monitor the Time Warp system and make changes in its behavior to maintain efficient execution. Other SAMs include flow-control, adaptive memory buffer management techniques (for shared-memory multiprocessors), and "risk" or optimism throttling protocols. An open question is how do these various SAMs interoperate? At the very least, these mechanisms should be designed such that they do not interfere with one another or feedback on each other in a manner that degrades performance. However, an even more interesting question than the issue of interoperability is the existence of a unifying SAM that encompasses all types of SAMs for Time Warp systems. Additionally, I plan to investigate the existence of a SAM that encompasses all distributed simulation protocols in general.

Modeling Methodologies for Time Warp Systems

One recurring theme in Time Warp research has been the importance of developing a suitable mapping to Time Warp's execution paradigm. This research suggests that there is no single best binding of model objects to Time Warp LPs and events and that the best binding depends on key model characteristics, such as lookahead, event granularity, LP state size, and locality of communications. Simulation models that examine different system performance characteristics may require a different object binding to the underlying Time Warp system to achieve good performance. Thus, the binding of model objects to Time Warp LPs and events should be flexible. How this flexibility should be realized such that it remains transparent to the applications developer and minimizes extra overheads in the Time system is an open issue that remains to be solved.

Distributed Simulation Models for PCS Networks

In collaboration with wireless researchers and analytic modelers, I plan to extend the capabilities of my previous PCS models by developing more realistic and complex models as a means of evaluating the techniques developed in my research of Time Warp modeling methodologies. These improved models will focus on understanding network performance under conditions which cannot be solved using traditional analytical techniques. The problems encountered from this and other collaborations would be used to fuel future research in distributed simulation systems.

Building a Better RTI

Despite the availability of RTIs for use by DoD simulation community, there exist a number of interesting issues that need to be addressed. First, many in the community are concerned about RTI interoperability. That is to say, how should the meta-RTI interface be defined such that all RTIs are able to interoperate as well as remain network and platform independent and still allow some flexibility in the implementation.

Another open issue concerns load balancing an HLA federation. What makes this problem interesting is that not every federate is able to assume "ownership" of (i.e., take the responsibility for simulating) another federate's objects. Consequently, the load distribution algorithm must find a good balance given the federation's ownership constraints. How to accomplish this is an open question.

Teaching Statement

Christopher D. Carothers

Philosophy

I look forward with enthusiasm to teaching at both the undergraduate and graduate levels. I view teaching very much as the Nobel Laureate, Dr. Richard P. Feynman saw it:

> I DON'T believe I can really do without teaching. The reason is, I have to have something so that when I don't have any ideas and I'm not getting anywhere I can say to myself, "At least I'm living; at least I'm doing something; I'm making *some* contribution"....
>
> If you're teaching a class, you can think about the elementary things that you know very well. These things are kind of fun and delightful. It doesn't do any harm to think them over again. Is there a better way to present them? Are there any new problems associated with them? Are there any new thoughts you can make about them?...
>
> The questions of the students are often the source of new research. They often ask profound questions that I've thought about at times and then given up on, so to speak, for a while... The students...*remind* me of a problem by asking questions in the neighborhood of that problem. It's not so easy to remind *yourself* of these things.
>
> So I find that teaching and the students keep life going, and I would never accept any position in which somebody has invented a happy situation for me where I don't have to teach. Never. [1]

Like Feynman, I believe teaching is a fun, rewarding activity that keeps a researcher grounded in the fundamentals and acts as a catalyst for generating creative, high-quality research.

Experience

As a PhD student and now as a research scientist, I have served both as a teacher and advisor to undergraduate and graduate students. As a Teaching Assistant for the Computer Organization course (undergraduate level) and Parallel Computer Architecture course (graduate level), my duties included substitute lecturing, designing and grading projects and homeworks, holding office hours, and giving tutorials on the use of our parallel simulation software.

In addition to my experience as a teaching assistant, I have served as a student advisor. During the summer quarter, the College of Computing hosts the minority internship program. The goal of this program is to get minority students interested in doing research and attending graduate school. I participated in this program during the summer of 1995 by serving as the student advisor to one student who selected parallel simulation as her project. What I found particularly rewarding about this experience was watching the students learn and hearing about the positive experiences contained in each student's project report. My student in particular throughly enjoyed the program and latter went on to use me as a reference.

As part of my research scientist duties, I have presented tutorials on our research program in Parallel and Distributed Simulation to industrial visitors such as Intel and Schlumberger as well as to incoming PhD and Master's students.

Finally, at the 1997 Winter Simulation Conference, I presented a State of the Art Tutorial on Visualizing Parallel Simulations in Network Computing Environments.

Course Interests

My teaching interests include undergraduate courses on programming languages, data structures, systems programming, operating systems, computer organization, concurrent computation and at the graduate level courses focusing on operating systems, computer architecture, and parallel and distributed systems. Moreover, I have a strong interest in curriculum development at both the undergraduate and graduate levels. In particular, I would seek to create, should they not already exist, courses relating to the area of simulation, such as modeling methodologies and simulator construction (both sequential and parallel simulator systems).

[1] From *"Surely You're Joking Mr. Feynman! Adventures of a Curious Character" by Dr. Richard P. Feynman, pages 165-166*