Modeling Cardiovascular Flow with a Migrating Agent System

Susan L. Mabry[a], **Lubomir F. Bic**[b] **Kenneth M. Baldwin**[c]
[a]Whitworth College, Math/Computer Science, Spokane, Washington
[b]University of California, Computer Science, Irvine, California
[c]University of California, Biophysics/Physiology, Irvine, California

Keywords: Software, Agents, Cardiovascular Physiology

Abstract

New dimensions and innovative methods are inaugurated for biomedical modeling and simulation with a migrating agent system, SimAgents. Agen-based technologies are one of the most important computing shifts in recent years. This paradigm offers flexibility and open-endedness not found in other processing paradigms. Migrating agent attributes align closely to characteristics of biological processes. The SimAgents framework was designed to accommodate flow problems. A full body, pulsatile cardiovascular model is implemented in a distributed, parallel and dynamic manner closely resembling natural domain processes.

Dynamic and interactive qualities are strengths of particular interest to the medical simulation community. In SimAgents, agent functionality enacts simulation steering such as short-term autoregulation and introduction of external interaction during runtime without halting execution. External events could originate from a user or conceptually from device readings. Concurrent flow properties are employed for bifurcations and merges, pressure waves and reactive mechanisms. Peripheral flows are modeled in a parallel compensatory manner, analogous to the natural system. This System lays a foundation for adaptive monitoring, interactive testing, device prototyping, integrated analysis and regressive analysis.

Introduction

Cardiovascular modeling has long been used for hypothesis study and biomedical systems analysis. It has been recognized that the cardiovascular system has inherently concurrent components and would benefit from distributed, parallel processing however, computing approaches have been limited. An agent-based framework introduces new modeling dimensions and opens new avenues for application of the models. Agent functionality is particularly suited to flow problems having distributed entities, parallel flow patterns and dynamic behaviors. Obviously, cardiovascular flow has such characteristics and can clearly use agent functionality advantageously.

SimAgents (Mabry 99) is a fully distributed, migrating agent system implemented in Java (SunMicrosystems) designed especially to meet needs of flow class simulation problems. The potential of emerging agent-based systems for simulation domains has been an application area largely overlooked. Because of high degrees of flexibility and open-endedness, migrating agents are extremely promising for biomedical applications. A full cardiovascular model has been implemented. Unique is the representation of complex physiological functions in a truly distributed and dynamic manner with control flow closely resembling the natural domain. Key to an effective implementation is the method of decomposition and partitioning afforded by a migrating agent system. The cardiovascular system is translated to a logical structure overlying a physical network of processors. Natural behavior flow of the cardiovascular system determines navigation paths of agents. Dynamically composed agents enact autonomous events such as short-term autoregulation. Interaction with the system supports the ability to introduce new events or actions during execution with external simulation steering.

In the paper, described are the basic cardiovascular flow problem followed by a correlation between the migrating agent paradigm and physiological processes. An introductory overview of SimAgents is given. Mapping of the cardiovascular model into SimAgents is discussed. Findings of the work and conclusions are presented.

The Cardiovascular Problem

The basic cardiovascular model implemented in SimAgents has been adapted from previous work, CVSys (Mabry et al 98). The domain is inherently concurrent with distributed physiological components and varying autonomous, asynchronous reactive events. The theoretical model is based on Rampling's premise, *the cardiovascular system consists essentially of a double, reasonably synchronized, pumping system each side of which feeds the other through a peripheral circulation made up of a vast number of series and parallel circuits* (Rampling 93). The circulatory flow is modeled as a bio-fluid, mechanical model with the heart treated as two pumps, with valve opening/closure functionality of left side of the heart pumping to the systemic circulation and right side pumping to the pulmonic circulation. After leaving the heart, flow is pulsatile and bifurcates to smaller arterial vessels, to distributed peripheral regions, through arterioles, organ beds, then merge back into venules, from small venous vessels into larger venous vessels until returning to the right heart.

Overlying the basic closed loop circulation are short-term reactive control mechanisms. Supported is the premise made by Karlson (Karlsson 95) stressing that any effort at modeling the arterial tree should resolve characteristic features such as distributed resistance and the ability to incorporate local variations in segmental compliance. Activation of distributed regulatory functions and resultant varying resistances and flows to parallel organ beds are addressed to a limited, proof-of-concept degree in this model. The framework is provided for more thorough representation.

The fundamental mathematical model is based on equations (Rideout 91), (Fung 96),(Hoppensteadt and Peskin 97), (Mabry et al 97), (Sagawa et al 78) with model assumptions accepted in comparable models carried into the mathematical model. The foundation is expanded to parallel perfusion of organ beds. Nonlinearization that naturally occurs in such aspects as pulse wave, venous valves, valve openings and closure; are discretized over regional segmentations and over timesteps. Resistance in arterial regions considers turbulence reflected in the computed Reynolds number. In the left and right ventricles, a half-sin wave is used in conjunction with pressure to enforce pulse wave. Fluid dynamics of the blood circulation computed in local regions are flow, pressure, volume, resistance, compliance and vessel radius. Normalized parameter values are derived from many references as delinated in (Mabry 99). Levels of physical exertion are used to characteristically perturb the model (Rowell 97).

Correlation of Migrating Agents to Cardiovascular Physiology

There is an abstract correlation of the underlying paradigm for migrating agents to processes of physiology. Biological structures and migrating agent-based structures exhibit common characteristics to varying degrees listed in Figure 1. Most complex of the cardiovascular system are reactive mechanisms and autoregulation. Short-term regulation is comprised of complicated, interwoven control systems interacting with other physiological components. Levels respond with compensatory actions to internal and external stimuli. There are multiple dimensions of determining factors affecting system dynamics and rates of state changes. In addition to complex reactive mechanisms, each person has differing base physiologies and then are further complicated by different pathophysiological conditions. The problem has characteristics and behaviors of multiple, integrated yet decentralized subsystems with regional elements having autonomous control flows; multiple controlling aspects of time, event and spatial nature; varying workloads; open-endedness and interactivity among both internal and external entities. Similar properties are displayed in the migrating agent paradigm. Clearly, characteristics of biological structures resemble features of a migrating agent-based processing system. Migrating agents provides levels of expressiveness and flexibility otherwise unattainable for such an application.

Characteristic	**Cardiovascular Biology**	**Migrating Agents**
Decentralized Structure	Distributed elements, Distinctive structures, behaviors, control	Distributed objects. autonomous specialized agents
Mobility	Independent moving entities	Navigation over networks, carrying state and behaviors
Stationary	Spatial entities with roles	Stationary objects mapped as logical nodes, stationary agents
Autonomous behavior	Subsystems, self-driven, regulatory actions	Autonomous agents, asynchronous behaviors
Varying Workload	External events, pathological conditions, regulation	Dynamic firing of agents. Varying invocations
Interaction, Communication	Nervous systems, reactions, flows, transmission properties	Passing values among agents, Communication, invocation
Event-driven activity	Autoregulation, responses to events, differentials, changes	Firing reactive agents; Internal external steering; Discrete event simulation
Spatial driven activity	Regional autoregulation, regional functions	Local region functionality, tasks
Cooperation	Among entities and processes towards common goal of maintaining circulatory stability	Agent tasks toward common system goals. Distributed tasks and capabilities
Integrated, multiple sub-systems	Functional subsystems, components and regions	Architecture layers, Objects and agents, Agents, Distributed problem solving

FIGURE 1. Characteristics Correlation of Migrating Agents and Cardiovascular Biology

The SimAgents System

SimAgents consists of an AgentEngine runtime infrastructure and of an Application Development Framework. The Application Development Framework is comprised of autonomous migrating *agents* and cooperative stationary *regions*. Objects representing regional entities are mapped to distributed network host sites. Regions retain local states, properties and conduct local behaviors. Agents *flow* along *paths* or *itineraries* of regions, performing actions and coordinating local behaviors. This *flowing* movement and related actions at stationary regions are analogous to entities moving through their own natural physical domains. The agent-based system can be thought of as a coordinating group of agents working in a framework of spatially oriented regions operating with their own individual agendas.

Agents are task oriented active objects, carrying boundary value objects and coordinating regional methods. Individual agents migrate through the logical network of regions according to their own behavior and tasks. Agents are context sensitive, detecting their physical location or an event, and acting accordingly.

Regions are stationary, passive objects comprised of state variables and local methods. One or more regions may reside on a physical processor host. In the partitioning scheme, regions are organized by spatial properties and/or by behavior roles. Local behaviors are invoked by visiting agents. Visiting agents can carry computed values forward to other regions during the course of their natural flow.

A *path* is a name for a predefined series of ordered links between regions. Functionality and behavior may be coupled to well-defined flows in a path abstraction. An *itinerary* consists of an explicit designation, also ordered, of one or more regions that an agent is to traverse. An itinerary may be assigned dynamically, “on-the-fly” without regard to a territorial flow. An agent is simply fired onto a path or itinerary.

A logical network distinct from the physical network is established providing the perspective from which the user views. A logical network consists of regions and paths of regions, of which agents will travel. Once an initial configuration is declared, the user is aware of logical distribution of objects but need not be concerned with establishing communication channels or with tracking distributed locations. Figure 2 illustrates the separation of logical network and underlying network of host processors, also shows agent migration among established regions.

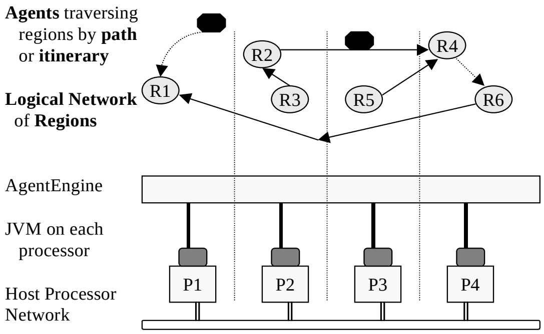

FIGURE 2. Logical Network over Network of Physical Processors. Agents traverse paths or itineraries among the framework of regions.

Dynamic agent composition is accomplished with a *fire* statement. *Fire* invokes, initializes, registers and puts into motion a new agent. Context sensitive agents perform local computations or actions relative to their perceived location and then are implicitly transported to the next region in its given path or itinerary. Firing of agents during runtime, without halting normal execution, is particularly effective for interactive simulations, providing two forms of simulation steering. Internal open-loop steering reacts to internal state changes and is accomplished through dynamic firing of an agent or of multiple agents upon a detected condition at a region. External event steering reacts to events introduced from a user interface. The programmer is shielded from migration details, from distributed physical locality and from distributed communication protocols. Java structured code further encourages a programmable environment.

Mapping the Cardiovascular Model into SimAgents

The cardiovascular system is translated into regions, agents, paths and itineraries. Blood flow is programmed from the agent’s point of view. The physiology is implemented as a series of heart chambers, arterial regions, venous regions, distributed and parallel organ bed perfusions with arteriole regions and venule regions. The ovals of Figure 3 depict the decomposition of physiology into computational *regions. Agents* representing blood flows, traverse from one region to another along emanating paths (shown in arrowed lines), analagous to natural blood flow. Agents migrate or flow along linked paths, traversing regions, branching as appropriate; much like blood pulsating through regions and flowing through branches of the circulatory system. This layer represents the basic closed loop blood flow throughout the body.

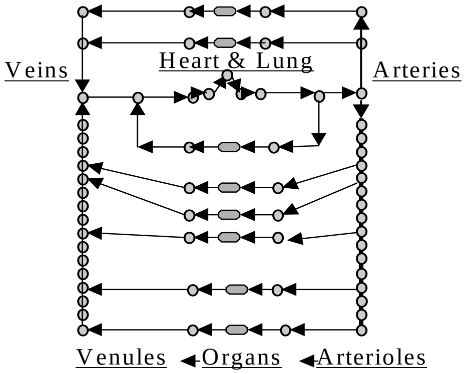

FIGURE 3. Basic closed loop circulatory physiology partitioned as stationary *Regions* in SimAgents.

The problem also has behaviors of multiple, integrated yet decentralized sub-systems with multiple controlling properties. Overlaid flow properties are evident in bifurcations and merges, pressure waves, and reactive flow signals such as baroreflex or cardiopulmonary responses. Flows are all programmed into specialized migrating agents, autonomously performing behaviors. Simulation steering is evidenced through numerous dynamic firings of agents. Internal steering is accomplished by agents enacting reactive compensatory mechanisms over the layer of basic circulatory flow. External steering allows reactions to events introduced from a user interface during runtime. Numerous types of flows can be overlaid onto such a system because of the open-ended, extensible qualities of migrating agents.

The migrating agents paradigm is especially suited to modeling the peripheral organ bed perfusion, an area often overlooked or lumped together because other computing paradigms do no provide an appropriate structure for these compensatory but naturally distributed parallel flows. The body's multiple regulatory systems act to maintain systemic blood pressure by regulating regional flow. The peripheral organ circulation beds and related control mechanisms offer the greatest opportunity for parallelism in the simulation system with concurrent fluctuating levels. Internal simulation steering expands the basic circulatory model, serving to provide short-term self-regulating properties affecting perpheral flow properties. Figure 4 illustrates a conditional open-loop steering example. In the local pressure calculation of *aorta*, if the pressure exceeds a threshold value then a regulatory agent is fired, representing a baroreflex mechanism. A fired agent travels to a region called *CNS* where further adjustments are calculated and fired as agents. Note the concurrent and naturally dynamic activity.

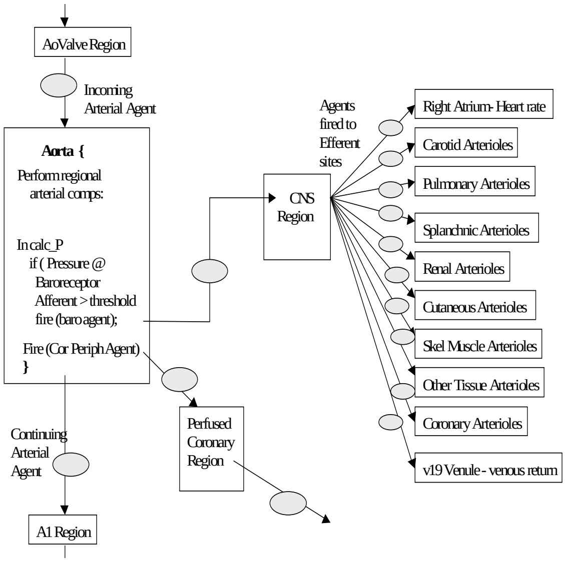

FIGURE 4. Conditional open-loop compensatory agent flows from aorta region.

External simulation steering introduces yet another strength that is of particular interest to medical simulations. Interactive entry of changing parameters are supported during execution. Figure 5 demonstrates the introduction of external factors from an interface, concurrent to already executing agents and regions.

With an entry from a user interface, simulation steering affords the opportunity to dynamically change patient conditions or introduce new events, during execution of the system without restarting the simulation. A further exciting approach, with timing mechanisms and data conversion, inputs could be supported from monitoring devices.

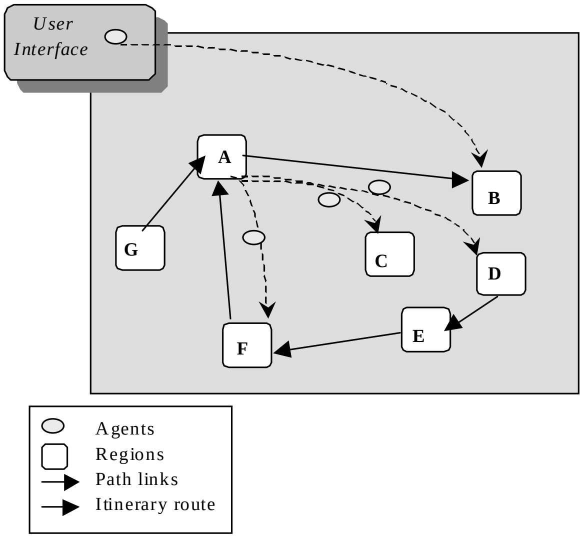

FIGURE 5. External simulation steering from Interface during runtime without halting execution.

Findings

This class of simulation models has previously been difficult to achieve. The SimAgents computational model closely corresponds to characteristics of autonomous flowing entities. A high degree of expressiveness yet simplicity results from close alignment of computing processes to actual domain behaviors. Consequently, such a system is a natural solution for this class of modeling and simulation problems. The distributed, agent-oriented processing has not led to differences in the formulas. Rather differences result in the granularity and order of processing. We are able to partition the spatial domain into more natural segmentations and process events in the natural order of physiological processes.

In comparing other, more traditional computing approaches for modeling, a *sequential* approach does not support distribution, concurrency and distributed controls over processing, hence it cannot preserve a natural order of events. With increased model complexity such limitations severely impact representative capability and management. The application is fit in a box of processing constructs rather than aligning processing to the natural order of domain events. For computation intensive models, high performance *parallel* processing such as has been used by (Hoppensteadt and Peskin97), suites a matrix intensive approach of fixed processing. Distributed computing has been desired for intermediate models requiring greater flexibility. However traditional distributed environments (using message-passing) have not been easily programmed and their open-endedness has been limited. In comparing *message-passing* with an *agent-based SimAgents*, major areas of agent-based distinctions include 1.) an agent-oriented approach rather than a machine-oriented approach, centered around agent flows rather than machine communication 2.) A logical network that provides transparency from underlying physical network details 3.) An easily programmable environment rather than distributed message passing and processor mappings.

The current version of SimAgents has not been tuned for performance and suffers high overhead in the agent firing operation. However, the performance numbers are strong enough to suggest that a combination of optimizations that are currently being implemented will make SimAgents competitive in performance while retaining its design principles. On a full cardiovascular testbed problem, the following results in Figure 6 were demonstrated in speedup over 100 MB/s Fast Ethernet connections. In the case of a steady state circulatory model without dynamic responses, a complete heart cycle is represented by 80,000 operations. Experiments with internal open-loop reflex responses and external event entries after the initial stabilization also prove interesting, although difficult to interpret effects of the varying level of computations and processing performance. While for time critical systems this approach is clearly not effective, it is significantly faster than pure interpreted approaches and with proposed optimizations, SimAgents is expected to be highly competitive in performance with other distributed systems.

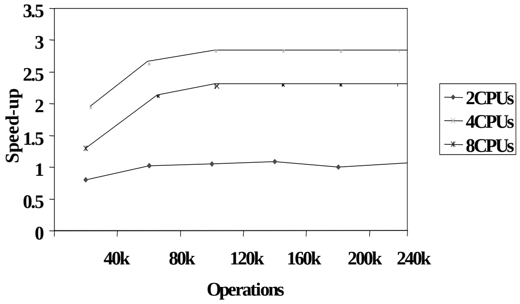

FIGURE 6. Speedup for SimAgents configuration upon 1, 2, 4, and 8 processors

Of application interest to biomedical communities is the fact that because SimAgents is developed entirely in Java, it is extremely portable to a multitude of computing platforms. An ongoing research topic associated with this system is the infusion of artificial intelligence methods into specialized agents. Specialized agents could then be a strong candidate system for intelligent or adaptive patient monitoring. There are several aspects of this system, making it attractive for “real-world” usage. Models developed in this paradigm could be adapted for laboratory or clinical usage such as adaptive monitoring, interactive testing, device prototyping, device validation and integrated analysis.

In the resulting implementation, overlaid pulse waves are reflected as agents traverse region to region. Changing heart rates are transmitted via agent migration and reflected throughout the cardiac cycle. A graph example of a resultant pressure wave is shown for a specific vessel region over a given time in Figure 7.

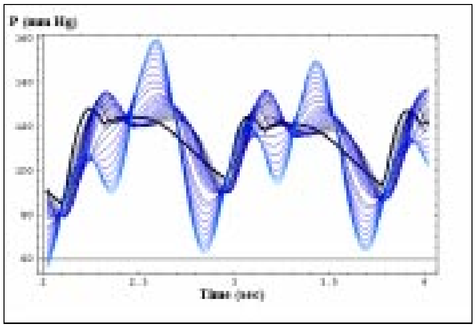

FIGURE 7. Pressure wave for vessel region over time (Mabry and Bic99).

Conclusions

In this paper, we have discussed how SimAgents provides unique computing approaches for cardiovascular simulation. Parallel processing as with massively parallel processing approaches have been used to improve performance; however does not address the autonomous control flows found in such problems. Nor do they allow a natural decomposition of the problem or open-ended interaction. Although distributed processing is desirable for a more natural depiction of the domain, traditional distributed approaches have presented both a difficult programming environment and obstacles in handling region-to-region differential equations of flow problems. These aspects are all addressed through application of migrating-agent features in SimAgents. Modeling aspects of a cardiovascular model were directly enabled by an agent infrastructure included multiple autonomously controlled flows, distributed organ perfusion and short-term compensatory mechanisms in an interactive, asynchronous manner without halting the runtime system. Migrating agent processing offers degrees of expressiveness and flexibility not available through other computing paradigms.

A testbed cardiovascular implementation has been described. Although limited in its present scope, the implementation is a large-scale application that establishes a framework for a much greater scope of cardiovascular representation. Migrating agent processing presents new ways to effectively represent a complex problem. It introduces avenues for greater interactivity and thus new interactive usage of simulation models.

References

K. Sagawa, H. Suga and K. Nakayama. Instantaneous pressure-volume ratio of the left ventricle versus instantaneous force-Length Relation of papillary muscle. In *Cardiovascular System Dynamics.* Eds. J. Baan, A. Nordergraaf and J. Raines. MIT Press, Cambridge, MA. 1978.

V. Rideout. *Mathematical and Computer Modeling of Physiological Systems*. Prentice Hall. New Jersey. 1991.

M. H. Rampling. Effect of blood rheology on cardiac output. In *Cardiac Output and Regional Flow in Health and Disease.* Kluwer Academic Publishers. Netherlands. 1993

M. Karlsson. Modeling and simulation of the human arterial tree - a combined lumped-parameter and transmission line element approach. *Computer Simulations in Biomedicine.* Eds. H. Power and R. T. Hart. Computational Mechanics Publications, U.K. 1995.

Y. C. Fung. *Biomedics: Circulation.* Springer-Verlag. New York. 1996.

The Java Language Specification, Sun Microsystems, Inc. Sun Microsystems Press, 1997.

S. Mabry, S. Rodriquez and J. Heffernan. Integrated Medical Analysis System (IMAS). In *Proceedings of WSC 97*. December, 1997.

L. Rowell, D. O'Leary and D. Kellogg, Jr. Integration of Cardiovascular Control Systems in Dynamic Exercise. In Handbook of Physiology, Section 12, Chapter 17, American Physiology Society. 1997

F. C. Hoppensteadt and C. S. Peskin. *Mathematics in Medicine and the Lifre Sciences.* Springer-Verlag. New York. 1997.

S. Mabry, L. F. Bic and K. M. Baldwin. CVSys: A Coordination Framework for Dynamic and Fully Distributed Cardiovascular Modeling and Simulation. *Proceedings of Biomedical Sensing and Imaging Technologies.* 1998.

S. Mabry. *SimAgents: Migrating Agents for Simulation Models.* PhD Dissertation, Department of Information and Computer Science. University of California, Irvine. March, 1999.

S. Mabry and L. Bic. Bridging Semantic Gaps with Migrating Agents. *Proceedings of Parallel and Distributed Computing Systems (PDCS'99).* Cambridge, Mass. November, 1999.

Author Biographies

Susan L. Mabry, received her MS and PhD degrees in computer science from the University of Southern California, 1993, and the University of California Irvine, 1999, respectively. She is currently Asst. Professor of Computer Science at Whitworth College, after several years of program management in Industry. Research interests include distributed program models, agents and agent processing as applied to modeling and simulation, particularly of physiology.

Lubomir F. Bic, received his MS and PhD degrees in computer science from the Technical University Darmstadt, W. Germany, 1976, and University of California, Irvine, 1979, respectively. He is currently Professor of Information and Computer Science at the University of California, Irvine. Research interests include projects aimed at developing new models of computation and tools for the programming of parallel and distributed systems.

Kenneth M. Baldwin. Dr. Baldwin is currently Professor of Physiology and Biophysics at the University of California, Irvine. Research interests include blood flow and cellular processes, muscle plasticity, protein synthesis and degradation.