Proceedings

The Fifth International Conference on Autonomic and Autonomous Systems

ICAS 2009

20-25 April 2009
Valencia, Spain

Editors/Chairs
Radu Calinescu
Fidel Liberal
Mauricio Marin
Lourdes Peñalver Herrero
Carlos Turro
Manuela Popescu
Data Consistency in Distributed Virtual Reality Simulations applied to Biology

Mikaël Bourhis\textsuperscript{1,2}, Gireg Desmeulles\textsuperscript{1,2}, Stéphane Bonneaud\textsuperscript{1,2}, François Guerrero\textsuperscript{1,3}, Vincent Rodin\textsuperscript{1,4}

\textsuperscript{1} European University of Brittany - UEB
\textsuperscript{2} ENIB, EA 3883 LISyC, CERV F-29280 Plouzané, France
\textsuperscript{3} EA 4324 ORPHY, UFR Sport et Education Physique F-29200 Brest, France
\textsuperscript{4} UBO, EA 3883 LISyC, F-29200 Brest, France

\{bourhis, desmeulles, bonneaud\}@enib.fr
francois.guerrero@univ-brest.fr
vincent.rodin@univ-brest.fr

Abstract

We perform biological simulations in the virtual reality context. In order to run large simulations, we choose to put together a set of standard computers and create a grid in charge of distributing the biological simulations. We propose to make the distribution on to the ReISCOP generic model, developed in our laboratory, which allows us to easily design biological simulations. This method is also based on the replication of passive elements. It is a spatial distribution in which local simulations are periodically synchronized and the consistency of replicated data is checked. This synchronization is not a strong synchronization but a weak one. The consistency method is built on the transmission between nodes of the grid of the state variations of the data. The software used for distributing simulations is named DIVA, and is an individual based software located on each node of the grid. DIVA confers a peer to peer architecture upon the grid.

Keywords: virtual reality, distributed simulation, peer to peer, consistency, biology

1. Introduction

The study of biological systems cannot be summarized to the sole experiments realized on laboratory benches. More and more, biologists use computer science tools. Particularly, in the genetic field, computer science is used to analyze nucleobase sequences and other DNA and protein elements. Computers allow to process a lot of data produced by the experiments. This field of computer science is called bioinformatics. Moreover, computer science forms a good alliance with mathematics when it comes to analyzing, understanding and simulating bio-chemical reactions.

Another less common approach in computer science for the study of biological systems is virtual reality. The main goal of the research taking place at the European Center for Virtual Reality (CERV) is the study of complex systems using virtual reality tools and methods. Part of this research applies to biology. For instance, simulations of blood coagulation [15], myeloma [16], allergic urticaria [9] and arteries vasorelaxation are being fulfilled. In order to build larger simulations than what is currently done, we argue that the grouping of several computers into a grid is required. We want to aggregate computing power. However, accessing data on remote nodes needs additional resources compared to an ordinary simulation. Those needs are significant when a data consistency mechanism has to be frequently executed.

This article describes our solution to access the data while introducing the smallest possible error to the running simulation. In the first section, we exhibit the ReISCOP generic framework on which are based our biological simulations on the one hand and the distribution of these simulations on the other hand. In a second time, we expose the main issues concerning distributed computing in order to position ourselves. Afterwards, we detail our consistency mechanism used for distributing data and we end this article with two applications that validate our approach.

2. Our point of vue on biological simulations

CERV’s projects, studying biology and natural ecosystems, are fulfilled by the “In Virtuo” team\textsuperscript{1}. Using virtual reality tools offers more than the solving of an equation of

\textsuperscript{1}http://www.cerv.fr/en/activities/invirtuo.php
Figure 1. Structural coupling in the ReISCOP model: the figure shows two Organizations (A and B) with a structural coupling. The Structure of Organization A is composed by the Constituents set \{C1; C2; C3; C4\}, while for Organization B it is \{C3; C4; C5; C6\}. The set \{C3; C4\} represents the structural coupling between A and B. The two systems (according to the two Organizations) have mutual influences through the modifications of C3 and C4 states.

a biological model, like in silico simulations; it also enables the user to enter a virtual environment that contains his model with which he can interact at any time: the human is part of the model’s simulation loop. This way of experimenting a model is called in virtuo experimentation [17], in reference to in vivo and in vitro experimentations that respectively take place on living systems and test-tubes.

Furthermore, a recent study [8] on the autonomy in biological systems models (focused on interactions and coupling between different autonomous parts of a biological system) led to the ReISCOP model and the associated API (Application Programming Interface). This API allows to easily specify or reuse autonomous entities and other objects characterizing in virtuo experimentations, e.g. cell, chemical reaction, chemical diffusion, and also utilities like SBML\(^2\) parser. This ReISCOP generic model (for Interactions Reification; Structure; Constituent; Organization; Phenomenon) allows to describe a system as a composition of its sub-systems. Each sub-system (called Organization) is composed of active elements (called Interactions) that act on passive elements (called Constituents). The set of passive elements can be handled by Interactions taking place in different Organizations. In this case, Organizations are structurally coupled (figure 1). The fact that Interactions are true computer objects and that they realize actions is another essential point of this model. These Interactions are not the result of the action of one Constituent on another. The Interactions are objects which have attributes and activities (periodic actions). And this is called Interaction reification. These Interactions act on specific Constituents. Phenomena instantiate Interactions taking place between the designated Constituents when required conditions are found. Therefore, this modelling is an interaction based modelling. Figure 2 shows the UML class diagram of the ReISCOP model. Several specific models in the chemistry, mechanics or biology fields come from this generic model.

3. Computing and simulations distribution

To design and realize our distribution software, we have been dealing with distributed virtual reality on one hand and grid computing on the other hand.

3.1. General ideas

Distributed virtual reality: the main issue is the interconnection and the coordination of virtual reality simulations through a network. Users can access the simulated environment anywhere. From a historical point of view, it is the DARPA (Defense Advanced Research Projects Agency) which has begun researching in this field by creating SIMNET [4]. In this system, each site broadcasts the positions and speeds of simulated entities. To reduce continuing network transfers, each site estimates the positions of entities simulated by others: this is referred as the dead reckoning. Thereafter, this method has been normalized (DIS: Distributed Interactive Simulation) and extended into the Real-Ghost notion. Since then, many systems have been developed using HLA (High Level Architecture) [7] or CORBA. One can quote for example DIVE [11] and VIPER [18]. We focus especially here on the following concepts: the simulation of physical objects in a shared environment, the transfer and propagation of events occurring on these objects. Furthermore, the systems are multi-users, synchronized and executed in real time.

Grid computing can be separated in two groups: global computing and metacomputing. Global computing consists

\(^2\)http://sbml.org
3.2. Positioning

We argue that grid computing working scheme might be a relevant solution for the distribution of the processing of virtual reality simulations. And more precisely, peer to peer architecture is exciting because it awards to each node autonomy and nodes are independent from the rest of the system. Although one can find various works concerning virtual environment distribution by the mean of a peer to peer architecture (P2P), our concern is not to build a multi-user distributed virtual reality system. In the article [3], we explain in more details our choice of a peer to peer architecture and the arguments that allow us to extend grid computing into multi-agents system — the term “agent” refers here to an individual based autonomous entity. The distributed simulations on a personal computer grid is supervised by our software called DIVA. Each computer runs the same DIVA software and becomes a node of the grid. By extension, the term “DIVA” also represents the simulation station where the DIVA software is running. All the nodes have the same functions and are peers, because each node can perform load balancing, synchronization and conservation of temporal and spatial consistencies. These peers are active and pro-active thanks to regularly executed activities, they take decisions in order to cooperate with others: they are true agents. Therefore, because of those characteristics, we denote them as Distributed In Virtuo Agents.

The next section deals with the data consistency of a virtual environment distributed on several nodes, it is the main contribution of this article.

4. Distributed ReISCOP using DIVA

4.1. Consistency model

In our model, the active elements, i.e. Interactions, are not duplicated. Only parts of the environment can have many images or replicas. These parts, the Constituents for instance, are passive elements. These images of the Constituents must not cause substantial bias in the simulation and must be consistent. The consistency of a virtual world is effective when the scene is identically seen by all entities that observe it. The consistency is broken when a change regarding one of the images is not brought to the others.

Spatial distribution is a common idea. In fact, a simulation with a cubical environment will be split into a set of small cubes, while a simulation with an elongated environment – like biological vessels – will be split into slices. This approach has been chosen for distributed virtual reality simulations based on HLA or distributed video games. Commonly, this approach is called data decomposition. Furthermore, in the context of video games or virtual reality, those data are physical objects which are located in space. This distribution involves a significant use of computing resources due to synchronizations. For example, if two entities interact together when simulated on different computers without shared memory, it is necessary for the computers to exchange faithfully the actions of the two entities. A great quantity of exchanges between the two simulation stations can therefore be induced. In the case of in virtuo experiments, we choose to implement a weak synchronization in order to reduce the network load.

Weak synchronization is different from strong synchronization. In the case of a strong synchronization, the data on simulation entities and their states are transmitted and updated at each change. On the opposite, the weak synchronization implies a periodical transfer of these data from time to time. More precisely, in our simulations, a scheduler runs each activity at a specific moment. A synchronization is considered as strong when it is executed at each step of the scheduler. So, in between two synchronizations, entities of the simulation can change their state only once. With a weak synchronization, we release this constraint in order to make a synchronization with a lower frequency than the scheduler’s. This process allows to reduce network exchanges between two stations but it is not safe: consider a station, we know for a fact that the vision of the simulated entities on remote stations is false. Recall that, here, entities of the simulations are Constituents of ReISCOP model. Constituents are passive objects that are subject to Interactions activities and thereafter their state changes. As mentioned above, an Interaction can be on only one simulation station at once, whereas Constituents can have images on many stations. Therefore, we must synchronize these images from time to time.

Let’s consider for instance two Interactions of diffusion acting on a chemical species (it is the Constituent) in a 1 × 1 × 3 size environment (see figure 3.a). When time is
Figure 3. Example of a bio-chemical environment distribution on 2 simulation stations: a/ sequential simulation; b/ the first step of the distributed simulation: Interactions are moved but not replicated – what about the shared elementary volume? –; c/ constituents of shared elementary volume are replicated: 2 images for the same Constituent with identical initial values.

\( t = 0 \) second, the chemical species concentration in the middle area is initialized to 100. The activity of the diffusion Interaction is to balance the chemical species concentrations at each scheduler step (the processing of the diffusion is based on the Fick’s law). Figure 3 sums up our distribution method for this example. We are going to detail the distributed simulation with three kinds of synchronization: strong synchronization, weak synchronization and weak synchronization with our specific consistency algorithm. Figure 4 shows the results:

- 1/ We consider here the strong synchronization with a Constituents state propagation at each scheduler step: thanks to Interactions of diffusion, concentrations balance is reached and the matter quantity invariant is the same.

- 2/ Now the weak synchronization: Constituents synchronization (by their state transfer) is faulty when the synchronization action is not carefully adjusted on the scheduler of Interactions. The invariant is incorrect.

If the Constituent state is not transferred to all images when modifications or disruptions occur, the synchronization cannot lead to correct results. To solve this problem, we have built a mechanism called images consistency algorithm (algorithm 1) that reduces as much as possible the error (as shown in figure 4-3c) appearing during a weak synchronization. This mechanism is not based on the Constituent state transfer, but on the transfer of its state variation. Unlike the distributed virtual reality simulations and the Real/Ghost model, none of our images have a reference state. And the transfer of the state only is not enough to have a consistent simulation. Precisely, the Constituent state is a set of real values needed in order for it to be characterized in the distributed simulation. For a located chemical species in an elementary volume, the state is the chemical concentration. For a cell body, the state is its spatial coordinates. So, the state variation is just the difference between the current state and a given previous state. This algorithm requires additional memory in order to store the Constituent’s previous state (see the algorithm 1). Notice that the consistency algorithm is the same whatever the DIVA, whatever the situation. No role (client or server, kind of the image) is specified. It shows the similarity of the nodes.

### Algorithm 1: Consistency algorithm

begin
  // Comment: “concernedDIVAs” is a set of
  // DIVA sharing at least one image
  concernedDIVAs ← FindDIVAs()
  for i ∈ Images do
    LocalDelta_i ← CurrentState_i − LastConsistencyState_i
    waitedDeltaNb_i ← card(concernedDIVAs)
    DeltaSum_i ← LocalDelta_i
    for d ∈ {concernedDIVAs} do
      send i to d
      send LocalDelta_i to d
    while \( \sum_{i \in Images}WaitedDeltaNb_i \neq 0 \) do
      image ← receive
      remoteDelta ← receive
      DeltaSum_image ←
      DeltaSum_image + remoteDelta
      WaitedDeltaNDb_image ←
      WaitedDeltaNDb_image − 1
      for i ∈ Images do
        CurrentState_i ←
        LastConsistencyState_i + DeltaSum_i
        LastConsistencyState_i ← CurrentState_i
  end

end
Figure 4. A synchronization example between 2 simulation stations in the framework of one chemical species and two diffusion Interactions distribution. 1/ Strong synchronization: at each Interaction activity, chemical species concentration transfer occurs (the state of the Constituent) 2/ Weak synchronization: the concentration transfer is being done regardless of Interactions activities. 3/ Weak synchronization with consistency algorithm.

4.2. Implementation

To fulfill the Constituents’ consistency, it is necessary that DIVAs, sharing images, stop for a short time, while the consistency algorithm is executed and the useful data transferred. In order for DIVAs to stop together and achieve transfers at the same time without a central supervisor, each DIVA implements a rendezvous mechanism. This rendezvous mechanism allows to set up a synchronization barrier. When DIVAs reach the time of the rendezvous, they execute the consistency algorithm. In the current version, DIVAs share a token and send it to each other. When one DIVA receives the token, it takes the initiative to start the rendezvous algorithm. The others answer to it and a common date is decided. This date is the latest Local Virtual Time of the concerned DIVAs. The details of the rendezvous algorithm are shown in algorithm 2.
Algorithm 2: Rendezvous

GoToSynchronizationBarrier():
begin
  UnfreezeLocalSimulation()
  runUntil(Rdv\_date)
  FreezeLocalSimulation()
end

ReceiveAllAnswers():
begin
  answers\_stack ← {}  
  while card(answers\_stack) ≠ card(concernedDIVAs) do
    answer ← receive from emitter
    answers\_stack ← answers\_stack + \{answer\}
  end
end

Main algorithm:
begin
  if OwnToken() = true then
    FreezeLocalSimulation()
    // Comment: "concernedDIVAs" is a set of
    // DIVA sharing at least one image
    concernedDIVAs ← FindDIVAs()
    Rdv\_date ← 0
    for d ∈ {concernedDIVAs} do
      send LocalVirtualTime to d
    ReceiveAllAnswers()
    Rdv\_date ← max(answers\_stack ∪ LocalVirtualTime)
    for d ∈ {concernedDIVAs} do
      send Rdv\_date to d
    GoToSynchronizationBarrier()
    ReceiveAllAnswers()
    for d ∈ {concernedDIVAs} do
      send StartConsistencyOrder to d
    ConsistencyAlgorithm()
  else
    date ← receive from emitter
    tokenOwner ← emitter
    FreezeLocalSimulation()
    if date > LocalVirtualTime then
      send LocalVirtualTime to tokenOwner
    else
      send LocalVirtualTime to tokenOwner
    Rdv\_date ← receive from emitter
    GoToSynchronizationBarrier()
    send Rdv\_date reached to tokenOwner
    StartConsistencyOrder ← receive from emitter
    ConsistencyAlgorithm()
end

Figure 5. Layout of the diffusion experiment: a 25 × 3 × 3 size environment and two probes. Injection of a chemical substance at the center.

When the rendezvous is finished (including the consistency), the DIVA that has the token sends it to another DIVA. Considering a Local Area Network, the token transfer is simple and does not need specific tools for its routing.

5. Applications

In the first part of this section, we show the results of a simple simulation that validates our approach. Even though this first application is quite trivial on the biological point of view, it enables us to fully control the simulation process, consequently giving us the opportunity to confirm our solution and illustrate the principle of image consistency. This example is based on the diffusion of a chemical species in a chemical environment. In the second part of this section, we introduce a bigger and more complex application: the artery vasorelaxation.

5.1. Diffusion experiment

Consider a simulation that only uses the diffusion Interaction in a bio-chemical environment. In a 25 × 3 × 3 size environment, we inject a quantity of 36.0 × 10^3 units of a chemical species in the middle part of the simulated environment at time t = 0 second. We let diffusion Interactions (acting between each mesh of the discrete environment) execute their activities. To get the following results, we put two virtual probes in the environment: the first one is placed in the central location and the second one is outlying, at 32% of the environment’s length (see figure 5).

The simulation is executed three times: the first time with one computing node only, the second time with five nodes and the last time with five nodes but without the consistency algorithm. Results in figure 6 show that curves are overlaid in the first and second cases. On the other hand, in the third case, when the DIVAs do not organize rendezvous for consistency to be fulfilled, the injected chemical species

...
can only diffuse itself in the environment simulated by the node, which corresponds to the environment’s center. The chemical species quantity reaches $666.7$ instead of $160.0$ (asymptotic value when $t = +\infty$). The curve corresponding to the second probe is not drawn in the third case because the measured value is always equal to zero. In conclusion, our solution is validated by those adequate results: the weak synchronization and the consistency algorithm build a correct distributed simulation.

5.2. Endothelium

For this virtual reality simulation applied to biology, we work in collaboration with biologists who provide us with data. This simulation corresponds to an *in vitro* experiment in which biologists inject a substance (acetylcholin) in the blood. The substance is caught by endothelial cells. This thin layer of cells that line the interior surface of blood vessels is called endothelium. Thereafter, a biochemical cascade fires up in the endothelial cells. At the end, endothelial cells drop out another substance (NO), which causes the relaxation of surrounding smooth muscle cells. An application example is described in [13].

In this simulation, three types of cells and blood are implemented. Among the cells, red blood cells only have a moving activity and no other internal mechanisms. On the opposite, smooth muscle cells catch the NO substance and hence expand. And, concerning endothelial cells, they have the most complex behavior: fifteen chemical reactions handling about fifteen chemical components are at work inside each cell. The distributed simulation currently takes place on five nodes. We can see on figure 7 many sets of colored boxes (purple, whitish, green and black). Each color represents a different computing node. In spite of quantitative results not similar to *in vitro* experiments (our parameters are not quite fine-tuned), qualitative results are right: smooth muscle cells are expanding according to the NO quantity produced by endothelial cell.

6. Conclusion

In the context of distributed simulations on a grid, the consistency of replicated entities and data (with more or less correctness) is one of the main issues, because compromises must be found between the most meticulous consistency and the overload of the computing system (CPU, network, memory). Indeed, consistency mechanism between two remote data requires some network transfer.

We propose a specific implementation of the data consistency algorithm for biological simulations in virtual reality. At first, we described the generic model used to build our biological simulations. Our practical method for the distribution, and especially the spatial distribution, is also based on this model. We do not replicate the active elements of the simulation (i.e. Interactions), but the passive elements (i.e. Constituents). Consequently, these passive elements have many images. The images are modified independently from one another. On the global simulation point of view, the images must be “synchronized” to keep their significance or their consistency must be restored.

We established that the only transfer of replicated data states during strong synchronization (i.e. at each step of
the Interactions scheduler) did not distort the global simulation. But during weak synchronization, this method was bound to fail. We chose a weak synchronization, because it gives more flexibility to computing nodes and reduces the network load. Therefore, our consistency algorithm is based on the transfer of states variations of the images. In order to validate our approach, we presented a simulation where only these mechanisms took place. We got identical results for the distributed and non-distributed simulations on a 5-nodes grid. At last, we introduced the simulation of the artery vasorelaxation for which the biological credibility is necessary, because this study is the result of a cross-disciplinary collaboration between computer scientists and biologists. The complexity of models in our Virtual Reality simulations and the complexity of usual mathematical models used in biological simulations (like ODE systems) are at the same level because our modelling—at the mesoscopic scale—is based on them. Nevertheless, in order to improve the consistency of images, we plan to add a dead-reckoning algorithm to it. The goal of this algorithm is to reckon remote state variation of an image and add it to local variation state between two synchronization steps. And next, we will increase the number of peers in order to run similar simulations on a larger set of personal computers.

7. Acknowledgements

The authors express their gratitude to Nelly Hervé for her linguistic help. This project benefits from the financial support of the Regional Council of Brittany and the European Social Fund (ESF) through a PhD grant.

References


