HLA-FEDERATE REPRODUCTION PROCEDURES IN PUBLIC TRANSPORTATION FEDERATIONS

Thomas Schulze
Steffen Straßburger
Otto-von-Guericke University of Magdeburg
School of Computer Science
PSF 4120
39102 Magdeburg, Germany
e-mail:{tom, strassbu}@isg.cs.uni-magdeburg.de

Ulrich Klein
Fraunhofer Institute for
Factory Operation and Automation IFF
Factory Ecology Department
Sandtorstrasse 22
39106 Magdeburg, Germany
e-mail: Uklein@iff.fhg.de

KEYWORDS

Transportation, Discrete simulation, Distributed simulation

ABSTRACT

This paper summarizes our conceptual and prototypical work related to the dynamic composition of heterogeneous distributed simulations using the High Level Architecture for Modeling and Simulation (HLA). Previous studies of the authors and within the HLA community have shown that federates other than of simulation type are feasible. The reproduction of federates during runtime (so-called "cloning") has been identified as basic dynamic pattern, allowing the creation or exchange of federates of the same or different type. The paper provides an overview of different cloning strategies within or external to a given federation and applies them to the public transportation domain. Finally, the potential of cloning strategies on the component (HLA: federate) level is described.

1 MOTIVATION

With the advent of the High Level Architecture it became feasible for the first time to build simulations and simulation based information systems (federations) in a distributed manner using heterogeneous simulations and other software or hardware components (the federates). In view of the HLA support for interoperability and reusability, federates can be (re-)combined to form simulations / information systems. As long as software adheres to the HLA specifications (e.g., strictly using the object model templates and the HLA RTI interface), it can act as a federate. Previous studies by the authors and the HLA community have shown that federates can also consist of online information systems, databases, etc. Different federates with identical object models can be used to offer the same objects in a real operational system or in training mode (e.g., one time using an online GPS-based fleet management system and a simulation in training mode). This flexibility is a major breakthrough in simulation based system design but in most cases has been limited to the pre-run-time. Once a set of federates is used for starting a federation execution, it remains the same (late entering federates are allowed, though).

The cloning approach presented in this paper tries to extend the flexibility of system composition to run-time. Federates can be created as copies of existing federates, and while the status and the external representation (in HLA the object model) remains the same, a different internal implementation can be used. This approach is not limited to one time management, rather it includes the parallel management of different time axes in order to allow e.g. ad-hoc forecast functionality. In this way a method for simulation model initialization is also derived, preventing the overhead incurred by starting at an empty-and-idle status.

2 CLONING TECHNOLOGIES IN SIMULATION ENVIRONMENTS

2.1 Related Work

The term "Cloning" in relation with distributed simulation was first introduced in literature in 1997 by (Hybinette and Fujimoto 1997, 1998). In their approach logical processes (LPs) of a parallel simulation are cloned, i.e., copies of logical processes of a parallel simulation are created during the execution of the simulation. The copies (clones) of the original logical processes would initially have the same state as the original processes, but possibly behave differently at a later point in time because of some user interaction. The intention of this is to conduct multiple what-if-analysis at the same time, e.g., by having multiple clones each evaluating the consequences of different strategies at the same time.

The main motivation for the approach taken by Hybinette/Fujimoto is to have as few as possible copies of logical processes running at the same time due to performance and resource issues. This is done by association of several "virtual" LPs to one "physical" LP until a virtual LP behaves differently and needs to be created physically.

Kindler (Kindler 1999) and Novack (Novack 2000) developed together the nested-simulation or reflective-simulation approach. The goal of this approach is to start a new simulation inside an existing simulation. The internal simulation delivers information that will be used by the original simulation. It is necessary to copy the state of parts or the whole original simulation model for starting the internal simulation. This approach has been developed for monolithical simulation models only.

2.2 Cloning in HLA-Federations

Since the idea of cloning simulations for the reasons of conducting multiple what-if-analysis and providing as-fast-as-possible forecasts could also be very useful and valuable in the context of HLA federations, concepts and prototypical solutions for developing a similar technique for HLA federations have been investigated. Different cloning technologies for HLA federations have been developed at the University of Magdeburg. The following techniques are suggested:

Internal Cloning: Internal cloning refers to the process of making a copy of one or more HLA federates and have them join the same federation as the original federates (Figure 1). The federates must take appropriate steps for distinguishing between messages (i.e., updates, interactions, and the like) between original federates and cloned federates. Possible solutions would incorporate special attribute and parameter tags for each update or interaction which is issued. The simpler case would be to clone federates which have only a passive behavior. How this approach could be used is described below in the discussion of the streetcar federation prototype, which serves as the demonstration federation for the cloning techniques described here.

External Cloning: External cloning refers to the process of making a copy of one or more HLA federates and have them join a different federation than the original one (Figure 2). The major advantage of an external cloning procedure is that it allows the cloned federates to apply a different time management scheme, e.g., they could switch from a real-time simulation to an as-fast-as-possible forecast. This is also an approach providing more flexibility compared to the original cloning technique from Hybinette and Fujimoto, where all LPs were executed in one simulation requiring all simulations to run on the same time axis.

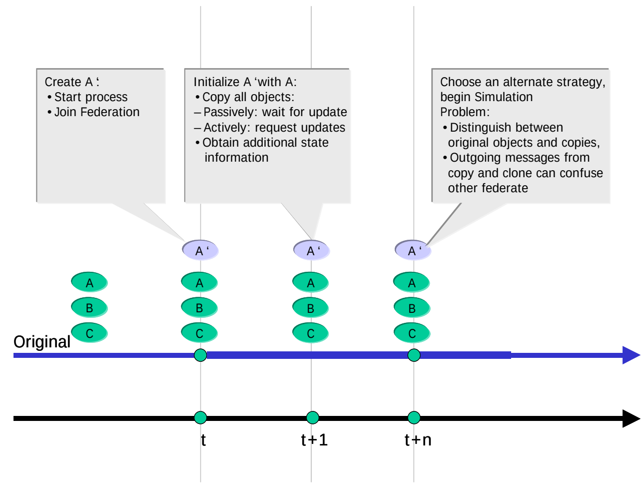

Figure 1:Suggested Procedure for Internal Cloning

External cloning also eliminates the need to distinguish between message traffic between original and cloned federates, which is an important issue in the case of internal cloning.

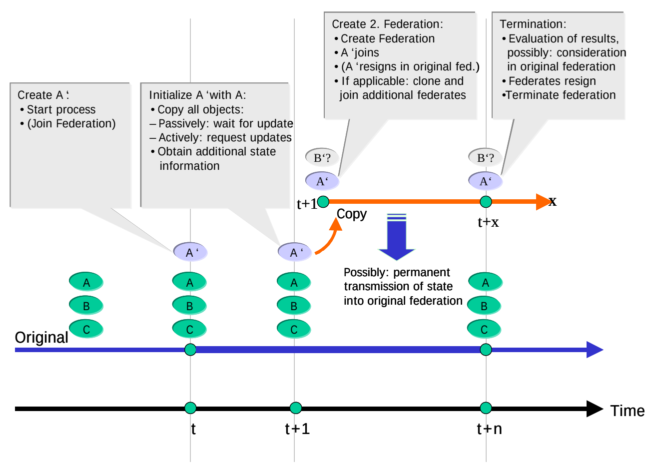

Figure 2: Suggested Procedure for External Cloning

Exchange of Federates at Runtime: Since exchanging federates with the same simulation object model (SOM) at runtime of a federation is a process which is (up to a certain extend) related to the issue of cloning of federates it will also be discussed here. The suggested procedure for exchanging federates at runtime includes the following steps (Figure 3):

- Create a copy (clone) or start a replacement federate
- Join this federate to the federation
- Acquire objects from original federate
- Resign the original federate

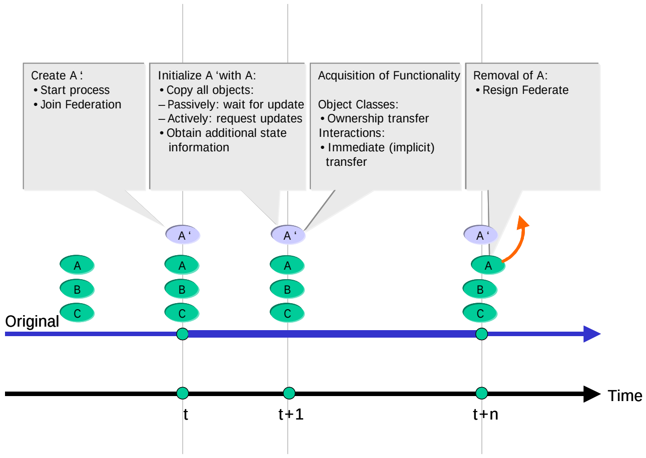

Figure 3: Suggested Procedure for Exchanging Federates at Runtime

2.3 Implementation Issues

In the cloning approach by Hybinette/Fujimoto the communication infrastructure (the Georgia Tech Time Warp Operating System, GTW) had to be enhanced for realizing the cloning. For the cloning of HLA federates, this cannot easily be done. Therefore different alternatives have been investigated.

A major prerequisite for the cloning of HLA federates is a technique for saving and restoring the state of a running simulation model. In the ideal case a simulation system would offer a built-in function for doing this itself. Examples for such simulators are GPSS/H and Simplex. GPSS/H offers the checkpoint function which saves a memory footprint to the hard disk and thus allows a later restoration of the model state. Simplex allows the user to schedule breaks for a certain simulation run. When reaching a break, the entire model state is also written to hard disk and can be resumed later.

Most commercial simulation systems do not offer a functionality for saving and restoring the state of a running simulation model. Therefore alternative means for copying the state of a simulation model have been investigated. The simulation system which served as the reference system for these investigations was SLX. The most important reason for this choice was the availability of a reliable HLA interface for this tool (Straßburger *et al*. 1998) and the possibility to easily be able to implement modifications or enhancements, in case they should deem necessary.

The difficulty when implementing the cloning or replication process for a simulation model is to cover all relevant model elements. Besides replicating the obvious items, e.g., object instances registered with a federation, other items need to be included, too. This relates for instance to certain simulator internals, like the state of random number generators, state of statistics already collected, etc.

Another question relates to the fact of how the state information from the original HLA federate to the replicated federates is transmitted. One could use external means, e.g., files, pipes, or a network connection. On the other hand one could also use means provided by HLA and its RTI itself. This is the path we chose for our prototypical implementation. The approach used here has some similarities to the object transfer approaches discussed in (Myjak *et al*. 1999a) and (Myjak *et al*. 1999b).

In our approach the original federate starts the cloned federate upon the occurrence of a certain event, e.g., by receiving an certain interaction, or command via the HLA Management Object Model (MOM). The clone is based on the same SLX source code like the original federate. After startup the clone detects the fact that it is indeed a clone and undergoes a special initialization phase. In this phase the clone requests updates for all object instances the original federate has registered in the federation. Also, additional information regarding the internal status of the original federate is transmitted. The clone is thus enabled to build up an almost identical copy of the internal state of the original federate. Some limitations apply regarding statistics and other details. More information can be obtained from (Albrecht 2000).

After the state of the original federate is replicated in the clone, the clone can enter its normal execution cycle. The further process depends on the aims for the cloning process. In an internal cloning procedure, the clone would co-exist with the original federate. This is most trivial in the case that the clone is passive and does not send messages (updates/interactions) to other clones. If this is not the case, further efforts need to be taken to distinguish between messages from the clone and the original.

An alternative aim could be to perform a forecast simulation. In this case, the clone could resign from the original federation and create/join a new one (possible in conjunction with other clones).

The third alternative would be a replacement of the original federate. In this case the clone would additionally request the transfer of ownership of object instances from the original federate. Then the original would resign and the clone would take over the responsibility for modeling the acquired object instances.

3 CLONING IN PUBLIC TRANSPORT FEDERATIONS

The HLA-related cloning is a new technology enabling a broad range of new and flexible applications. We are convinced that cloning technology will promote the use of simulation in traditional and new application areas. On a more general level application-oriented simulation models can be divided into three categories: design of systems, management of systems, and staff training applications. This paper presents three cloning technology examples from the second and third general application areas described above. The examples are focusing on public transport federations. Public transport has been selected because most people have related background and their own experiences. Furthermore, public transportation prototypes already have served as demonstrations for other advanced HLA-based work, such as online-/realtime-federates.

3.1 Support for operative management - forecast

An increasing number of simulation models is used in the operative management of public transportation systems. These models help for example managers to evaluate the system capacity for new transportation tasks, for breakdown- or accident-related changes in the number of available vehicles and for changes in the availability of staff. They give support to the management of systems for analysis of throughput and detection of bottlenecks. The management wants to evaluate operating decisions in relation to the performance of the system. In general the aim of models of this category is to support the operative management. The management wants to gain experiences from the future. We call these dedicated models Management Simulation Models (MSM).

MSM require greater level of detail than models for transportation system design. For example, complex control mechanisms and strategies may have to be implemented. On the other side, the MSM must be initialized with the state of the real system. For this task two possibilities exist: The simulation can start from a null-and-idle-status and run until it reaches the current state of the real system (e.g., using a trace file), or the model can be initialized directly with the state of the real system.

We prefer the last possibility using the external cloning procedure. The basic idea is to use one current-state-federation as a mirror of the current real system state. A real-time simulation model (federate) exists inside this federation and this federate will be updated permanently by online-data. The time advancement in the simulation federate will be paced by the real-time. This way it can be ensured that the simulation model maps the current system state. When the management wants to get decision support, the simulation federate will be cloned and it will be placed in a new federation.

The new federation has to contain the cloned federate and optionally other federates for human-interactions and visualization. This simulation federate operates as fast as possible and has to return a forecast considering the new conditions. The advantage of this solution is that the simulation model is indeed initialized by the current state of the real system.

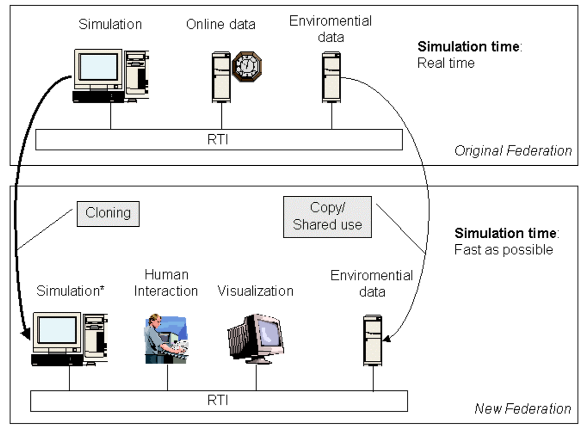

Figure 4: Suggested Procedure for External Cloning

3.2 Support for operative management – different strategies

Operative management is often confronted with the issue of choosing between different decision alternatives. Simulations can help to support the process of choosing the best suited decision. This can either be done by providing as-fast-as-possible forecasts, as discussed in the previous section, or by evaluating many different alternatives at the same time.

For the latter approach we suggest the internal cloning procedure: Several copies (clones) of a simulation model are created within the same federation. Each clone evaluates a different strategy. One implication of running within the same federation is that usually the same time management needs to be applied. This is also the approach which was chosen by Hybinette and Fujimoto.

In case of real-time proportional federations, this approach is therefore best suited for situations, where no immediate decision at a fixed time needs to be performed. In situations where an immediate decision is necessary, many as-fast-as-possible forecasts (faster than real-time) for all different strategies are needed.

Running multiple simulation copies within the same federation has the advantage that each copy can use any other information provided from other federates. This is especially important in the case of on-line transportation simulations using some sort of real-life data source. In this case each simulation copy can use this on-line data and perform its strategy evaluation based on this data.

The evaluation of different strategies based on this approach can be useful for strategic planning of traffic schedules, e.g., by comparing how different routing alternatives would influence the throughput of the system. The optimization of riding times, avoidance of traffic delays, and the minimization of necessary transportation devices can be additional target dimensions.

3.3 Support for staff training

Training is one of the most important application areas of simulation technology in the military and civil domain. This is due to the rareness of the events to be trained, costs or danger incurred by real training or the non-existence of the appliances or systems to be used or trained.

The training of staff can be executed in a pure training environment or in combined training and real life environments.

In the fist case the trainee operates in a traditional closed pure training environment. The trainee is integrated as an human player in the federation. He has to react on actions that will be delivered by a data supplier federate. The federation is completed by any other passive information and visualization federates. A training session can start with a data supplier federate focused on simple tasks. The trainee can thus be limbered up. After the warm-up period this supplier federate will be exchanged by an other. This new federate can deliver the real data online from the underlying transportation system. The trainee does not perceive that he has to respond to the real conditions. The trainer observes the reactions of the trainee and he/she can exchange the data supplier federate again. The exchange of federates during runtime will be applied.

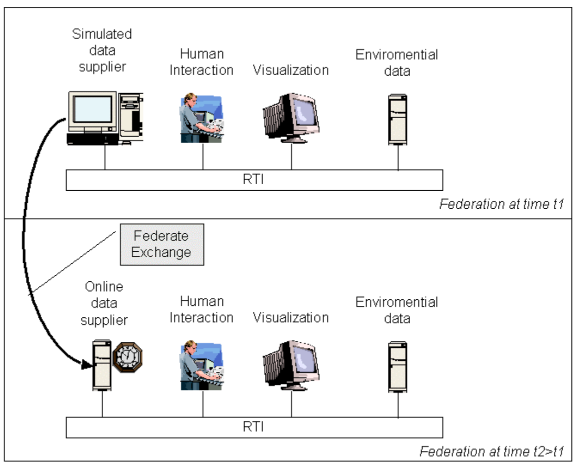

Figure 5: Suggested Procedure for Federate Exchange

In the second case the trainee acts in an open or integrated training environment. He is integrated in the real control and command structure but his reactions will not influence the real process. The control and command federation consists of the active human dispatcher federate, the online data supplier federate and other necessary federates for visualization and data bases. When the trainee will be integrated the dispatcher federate will be cloned and this federate will be the trainee-federate in the federation. The trainee does not work in an artificial training environment. He is integrated in the real system, using it in an "personal" training mode. The internal cloning will be used for this application.

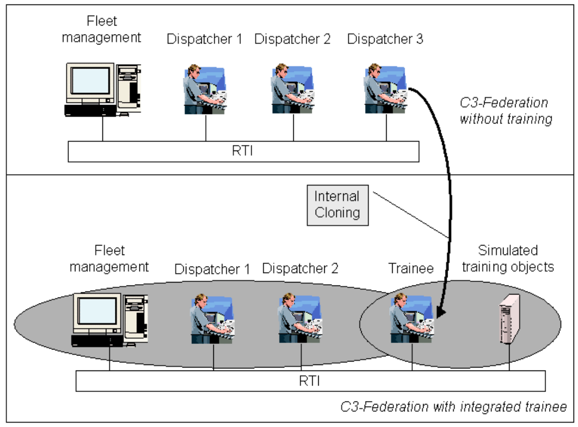

Figure 6: Suggested Procedure for Internal Cloning

4 PROTOTYPE

The prototype for test and demonstration of the cloning approaches has been a federation modeling the streetcar system of the city of Magdeburg (Schulze *et al:* 1999).

4.1 Internal Cloning

In this target scenario the simulation federate of the federation is cloned. The clone joins the original federation and requests information about all registered objects from the original federate, i.e., all streetcars currently moving through the city. After receiving updates for all objects the clone starts simulating the streetcar traffic based on the current schedule or any other chosen routing strategy. In opposite to the original federate the online positions for the vehicles will no be regarded. The cloned federate simulates the public traffic system without the real perturbations. Since the clone has a passive behavior, i.e., it does not send updates to other federates except the visualization component, no special mechanisms for distinguishing between original and cloned objects need to be introduced. The visualization federate can show both object types and thus visualize the different routing strategies chosen for the streetcars.

4.2 External Cloning

In this target scenario the simulation federate is cloned, too. It also requests information about all registered objects from the original federate. After that it resigns from the original federation. The clone is then used to perform an as-fast-as-possible forecast of the streetcar traffic. This enables the user to obtain an impression how a certain situation will develop and if bottlenecks will be likely to occur. For this forecast the cloned federate needs to advance simulation time faster than real-time. Therefore it can

- run independently from other federates without joining a new federation
- join a new federation and cooperate with other (cloned) federates, e.g., the visualization component.

4.3 Exchange of federates at runtime

In this scenario the on-line data source of our federation is exchanged by an off-line data source. This replacement is a straight-forward process: The off-line federate is started. It retrieves all necessary state-information from the on-line, e.g., the current simulation time, time stamp of the last update/interaction sent. Then the on-line federate resigns and the off-line federate starts sending position updates. This procedure is useful for operator training. By switching from a real system to a simulated scenario (without the need for the operator to know if the scenario is real) reactions to certain situations can be tested and trained.

5 CONCLUSION

The paper derived the cloning process on the component (HLA: federate) level as basic pattern of dynamic system composition applications, e.g., dynamic system reconfiguration, ad-hoc forecast analysis, and training applications.

The distinction between external description (in HLA the object models) and internal implementation details allows the creation or exchange of federates with the same external description and the same or different internal "nature" (e.g., simulation, online process, database or human-in-the-loop). This approach also covers the use of different time axis in order to allow, e.g., as-fast-as-possible forecasts during a real-time main application.

The public transportation domain has been chosen as application area for the prototypical work of the authors. Future steps include real-life prototypical applications of the different cloning strategies and the development of cloning and dynamic system management tools. The potential of cloning and other dynamic patterns is considered enormous, requiring considerable research and tool support.

ACKNOWLEDGEMENTS

This work has been supported by the Magdeburger Verkehrsbetriebe GmbH (MVB) by provision of their public transportation schedule and online positioning system. The geographic baseline data used in the visualization federate of the public transportation prototype has been provided by the Landesamt für Datenverarbeitung und Vermessung, Saxony-Anhalt, Germany.

REFERENCES

Albrecht, G. 2000. *Konzeption und Implementierung von Cloning-Techniken in HLA Federations*. Master’s Thesis. Institute for Simulation and Graphics. Otto-von-Guericke University Magdeburg. July 2000 (In Progress).

Hybinette, M., and R. Fujimoto, 1997. Cloning: A Novel Method for Interactive Parallel Simulation. In *Proceedings of the 1997 Winter Simulation Conference*, pp. 444-451. ed. S. Andradóttir, K.J. Healy, D.H. Withers, and B.L. Nelson.

Hybinette, M., and R. Fujimoto, 1998. Dynamic Virtual Logical Processes. *Workshop on Parallel and Distributed Simulation (PADS’98)*, May 1998.

Kindler, E., 1999. Nested simulation of container yards. In *Proceedings of the Simulatiuon und Visualisierung ’99 Magdeburg*, ed. O. Deussen, P. Lorenz und V. Hinz, 247-259. SCS-Europe BVB, Ghent, Belgium 1999.

Novack, P., 2000. Reflective simulation with Simula and Java. In *Proceedings of the Simulatiuon und Visualisierung 2000 Magdeburg*, ed. T. Schulze, P. Lorenz und V. Hinz, 183-196. SCS-Europe BVB, Ghent, Belgium 2000.

Myjak M., S. Sharp, T. Lake, K. Briggs, 1999a. Object Transfer In HLA. In: *Proceedings of the Simulation Interoperability Workshop*, Spring 1999.

Myjak M., S. Sharp, W. Shu, J. Riehl, D. Berkley, P. Nguyen, S. Camplin, M. Roche, 1999b. *Implementing Object Transfer In the HLA*. . In: *Proceedings of the Simulation Interoperability Workshop*, Spring 1999.

Straßburger, S., T. Schulze, U. Klein, J.O. Henriksen. 1998. Internet-based Simulation using off-the-shelf Simulation Tools and HLA. *In Proceedings of the 1998 Winter Simulation Conference*, eds. Medeiros, D.J. and E. Watson, Washington D.C.

Schulze, T., S. Straßburger, U. Klein, 1999. On-Line Data Processing in Simulation Models: New Approaches and Possibilities Through HLA. In *Proceedings of the 1999 Winter Simulation Conference*, pp. 1602-1609. December 5-8, 1999. Phoenix.