Anticipating Critical Events to Customise and Improve the Performance of Federation Runtime Infrastructure

Romeo Bandinelli [1], Mauro Iacono [2], Alessandra Orsoni [3]

[1] *Dipartimento di Energetica*, "Sergio Stecco", Università di Firenze, Italy.
[2] *Dipartimento di Ingegneria dell'Informazione,* Seconda Università di Napoli, *Italy.*
[3] *Kingston University*, School of Business Information Management, Kingston, England, United Kingdom
Emails: romeo.bandinelli@siti.de.unifi.it; mauro.iacono@unina2.it; a.orsoni@kingston.ac.uk

***Abstract* - This paper proposes a methodology to improve the time-performance of distributed simulation based on the HLA architecture using commercial simulators. In the context of industrial production, for instance, the timing of information transfer among federate simulators becomes an important issue whenever production control and scheduling are very closely linked. In such a context the re-scheduling of an existing production plan, as determined for instance by a rush order, is usually decided upon at the control level in the hierarchy of federates; however it cannot be communicated to the different federate plants until a status/parameter update is instantiated and established. The study presented in this paper specifically focuses on the impact of machine failures at the federate plant level. The random nature of the failure processes associated to the machines of the federate production plants makes it difficult to establish when such occurrences may take place: therefore the events driving the exchange of state variables among federates cannot be set with certainty ahead of time. This research proposes a one-off methodology to statistically determine when critical events, such as machine failures, are likely to occur. Based on such information it is possible to intersperse additional events among those associated to the usual production/process milestones and thus significantly reduce delays in the transfer of critical status parameters among federates and waste of simulated time.**

Keywords - distributed simulation, production planning, remote scheduling, federation run-time performance.

I. INTRODUCTION

Distributed simulation finds wide application in the modelling of complex systems that can be represented as aggregations of communicating subsystems, characterised by their own internal dynamics. In some application contexts simulation results are useful, and thus valuable, only if available by a specified time, which makes the speed of simulation response a critical measure of performance. Distributed simulation significantly enhances time-performance but introduces other problems related to the need for (simulators) synchronization. In the simulation of reconfigurable coordinated manufacturing plants, the random occurrence of component failures can cause waste of simulated time because failure events cannot be scheduled a priori at the global level. Component failures are generated locally during simulation and communicated to the system control level (and thereby to the other simulated plants) with necessary delays that may invalidate significantly large batches of simulated time. On the other hand, a finer-grained communication among simulators can reduce waste, but has a negative impact on time-performance, as it significantly increases the total simulation time. This paper proposes a simulation technique in which component failures are anticipated and managed at a global level to minimise wasted computing time, while enabling the identification of possible reconfigurations of the real system to reduce wasted production time. The study presented in this paper addresses the time-performance impact of machine failures occurring at the federate plant level. Because component failures are stochastic in nature it would be impossible to establish their exact timing ahead of time, therefore the corresponding events cannot be scheduled with certainty on an a-priori basis. The methodology proposed in this paper statistically determines when critical events, such as machine failures, are likely to occur for a particular federation and uses this information to intersperse additional events among the ones regularly scheduled to meet the federation requirements. This way the necessary delays in the transfer of critical status parameters among federates can be significantly reduced. Specifically, the methodology focuses on the timing of such events so as to anticipate and customise their occurrence according to the probability distributions representing the process statistics pertaining to each individual federate. The relevance of the study can be appreciated both in terms of runtime performance and in terms of accuracy of representation, as it makes the model more responsive to and representative of the actual federate processes. Preliminary tests have been performed to compare the runtime performance of the federation "as-is" and that of the federation with "added-in" events.

II. REMOTE SCHEDULING

Simulation effectively supports decision making in the management of complex industrial realities. This is the case of coordinated manufacturing plants, which are integrated in a production system and contribute to the execution of a production plan [Williams and Narayannaswamy, 1999]. Such systems can be efficient and cost-effective if their production plans can be flexibly adjusted to cope with local unit/component failures while maintaining high throughput standards [Proud, 1999]. This production management approach can be implemented if the timing of critical failures can be anticipated and an appropriate reconfiguration of the production plan can be identified to best cope with the particular failure mode. A distributed simulation approach representing each production plant as a simulation federate best reflects the actual structure of complex production systems and can be effectively used for the validation of production plans in degraded conditions whenever fast responses to on-line failures are needed [Orsoni et al 2003]. Production plans are generated at the control level, typically implemented as a simulator of its own, which is also in charge of coordinating production and information transfer among the federate plants [Chang and Maskatsoris, 2001].

The generation of alternative production plans by simulation is based on the current state of the whole system and must be carried out ahead of time in order to make efficient production levels sustainable when dealing with local failures. A quick-response distributed simulation approach is thus needed to minimise the waste of production time [Bandinelli et al 2003]. A primary cause of waste, as far as distributed simulation time, is the inadequate timing of communication among simulators. Coarse-grained communication minimises the time spent on the transfer of information among simulators: however its effectiveness is limited by the synchronisation needs of the application. In fact, if the need for synchronization is not frequent and all the simulators have comparable computing loads in between synchronization times, the occurrence of asynchronous events may invalidate the computation for the entire interval and require it to be re-computed. This is the case of events that cannot be scheduled a priori, such as failure events in the application domain of industrial production. The adoption of optimistic techniques, such as Time Warp, is also time consuming (as non-useful computations are carried out and extra simulation time is needed to rollback the status and process anti-messages) and it is not applicable when using commercial simulators, which typically do not allow for check pointing or saving and restoring their status.

III. STRUCTURE OF THE APPROACH

As mentioned in the previous sections, one of the issues concerning the correctness of the simulation results is the effectiveness of the simulated time advancement mechanisms. The occurrence of an event, possibly a critical one, within the scope of one simulator may not be recorded by the other federates for a long time, as the update of state variables specific to other federates is only possible when a given phase of the simulation is completed, and the corresponding data is shared among the different federates [Hwang, 1993]. Simulation phases are necessarily processed as batches that are executed in parallel by the different federates; after the completion of a phase, every federate is updated with the set of simulation results (produced by other federates) that were specified a-priori as relevant [Juhasz et al., 2003]. Waiting for the completion of a phase before checking for the occurrence of a critical event, which could invalidate the whole phase, is thus very time-consuming and must be avoided whenever the specific application makes the time spent on simulation as relevant as the actual simulation results in terms of effectiveness and performance impact on the real system [Gunasekaran et al. 2001]. In the context of industrial production this issue becomes important when dealing with remote production scheduling, especially if the control of production and scheduling are very closely linked. In such a context the re-scheduling of an existing production plan, as determined for instance by a rush order, is usually decided upon at the control level in the hierarchy of federates; however it cannot be communicated to the different federate plants until a status/parameter update is instantiated and established [Bandinelli et al. 2003].

When dealing with scheduling in a distributed simulation environment, independent of the type of simulator(s) and infrastructure, the architecture is relatively defined. At fairly short time intervals every single plant simulator/emulator needs to communicate with the scheduling system, which could be modelled using either a scheduling package or a simulator of its own. The approach considered in this research is based upon a distributed architecture using a RTI software tool for the purposes of information transfer. The HLA-based RTI can implement two distinct time advancement mechanisms, the NextEvent and the TimeStepped paradigm, respectively. Both mechanisms assume that each simulation federate at any point in time is aware of the next event requiring an information transfer to the other federates. Moreover, in a HLA federation, once a federate has communicated that it does not have information to exchange until a certain time, that time cannot be modified in any possible way. During the time period elapsing between one communication and the next one, each federate can advance in time, simulating the behaviour of either the plant or the scheduler, without any external exchange of information. This architecture does not reflect the reality of a production system, not even of a distributed one, where in contrast communication among production units or between each unit and the production management function may take place at any point in time without notice.

Figure 1 illustrates an example time advancement step. Let us consider a federation including two federates only: one modelling the behaviour of the physical system (PS) and the other the behaviour of the control system (CS). It is assumed that the PS simulator clock is one time step ahead

DOI 10.5013/IJSSST.a.22.02.01

ISSN: 1473-804x online, 1473-8031 print

of the federation: that it is stopped while the CS simulator clock is advancing in time, and that it advances to the federation clock time, only when the CS simulator clock is stopped. It is also assumed that the time advancement mechanism is of the NextEvent type, and thus that the PS, whenever a local event occurs, sends a request to the RTI to advance to the corresponding time, thus enabling the CS to schedule for the next event. This way the time advancement is driven by the PS, which iteratively notifies the CS of the time when a significant event takes place and thereby forces the simulation to stop. These assumptions are arbitrarily made (the system could work the other way around as well) however they do not impose any limitations or additional constraints as far as the analysis is concerned.

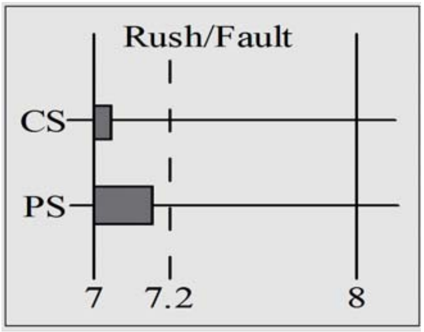

Figure 1. Delay in information transfer

On the other hand it is never possible that the clock of more than one federate simulator be ahead of the federation clock: if that was the case, the federates could not be re-aligned without violating the time-advancement rules set in the RTI when the NextEvent paradigm is implemented.

Based on Figure 1, the PS simulator has reached time 7 and stops the local clock to both communicate its change in state to the CS and receive back information on how to carry out the current production plan. The CS, knowing the status of each order at time 7, updates and communicates the next production steps to the PS according to the new state of the plant. After this, while the CS clock is not advancing in time, the PS simulation is resumed until the time when another significant event takes place. In the figure this corresponds to time 8. Once the change in state occurred at time 8 has been communicated, the cycle is re-iterated.

As indicated in the figure, it is possible that at time 7.2 the CS receives a change order notification involving for instance the production of an additional, unplanned for, batch. This change involves the rescheduling of the entire production plan: however, due to the constraints of the architectural set up, the new production schedule cannot be communicated to the PS before time 8.

It is apparent from the example that the simulation suffers from a communication delay of 0.8 time units, which does not reflect an actual delay in the real system.

IV. TIMING CRITICAL EVENTS

The critical events that more strongly influence simulation performance because of delays in the transfer of the relevant information include machine failures recorded in one of the federate production plants, decisions on how to change the current production plan in response to machine failures (as the time needed to implement the change is not known a-priori), and quick production changes due to rush orders. Because such events are unprecedented they do not usually/necessarily overlap with other production milestones set for the federation as a whole, it may take a long time before the different federates become aware of such occurrences/variations. This research aims at implementing an effective fault prediction technique to enhance the performance of an existing federation for a specified production system.

The application of the fault prediction technique should be as transparent as possible to the federate simulators in order best reuse the consolidated simulation infrastructure. For such purposes a methodology has been designed involving the introduction of an additional simulator in the federation to generate and schedule “Predicted Failure” (PF) events. The introduction of these PF events causes the federation to synchronize the status of the simulators every time a PF event is scheduled: each PF event is to be generated right after (but as close as possible to) the time of a local failure occurrence in one of the federate simulators (“Simulated Failure” (SF) event), which would not typically cause synchronization. This mechanism effectively reduces the wasted computing time as long as the failure prediction technique is correctly implemented, and as long as the delay in FP event generation (“Predicted Failure Event Delay”, PFED) with respect to the corresponding PS event is suitably small. The proposed methodology consists of the following steps:

1. Fault analysis for each plant and definition the probability distributions representing the failure rates of each plant component.
2. Analysis and definition of the impact of a given plant’s failure on the productivity of other plants and thereby of the production system as a whole.
3. Implementation of an additional federate simulator capable of generating PF events.
4. Identification of the optimal PFED in relation to the overall simulation speed.
5. Validation of the new federation.

These steps and the corresponding input requirements for the implementation of the methodology are sketched in Figure 2.

In simple application cases the types of failure modes associated to the physical system and their secondary and tertiary impact on the productivity of the entire system are self evident, and can often be identified just by inspection.

In more complex cases, though, thorough fault analysis techniques are required to carry out steps 1 and 2 of the proposed methodology. As part of this ongoing research several techniques are being explored for Failure Events Prediction. The first step (Fault analysis and component characterisation) can be approached using two different techniques. The first one is based on Fault Tree Analysis (FTA), commonly used both in academia and in the industry to obtain the overall fault characterisation of a system . building from both the fault characterisation of its basic components and their architecture. In production applications, each of the manufacturing plants is first analysed as an independent subsystem in relation to all its possible configurations. Such a strategy offers all the advantages of a formal analysis technique but highly increases the complexity of the simulator and requires a significantly long and expensive upstream analysis.

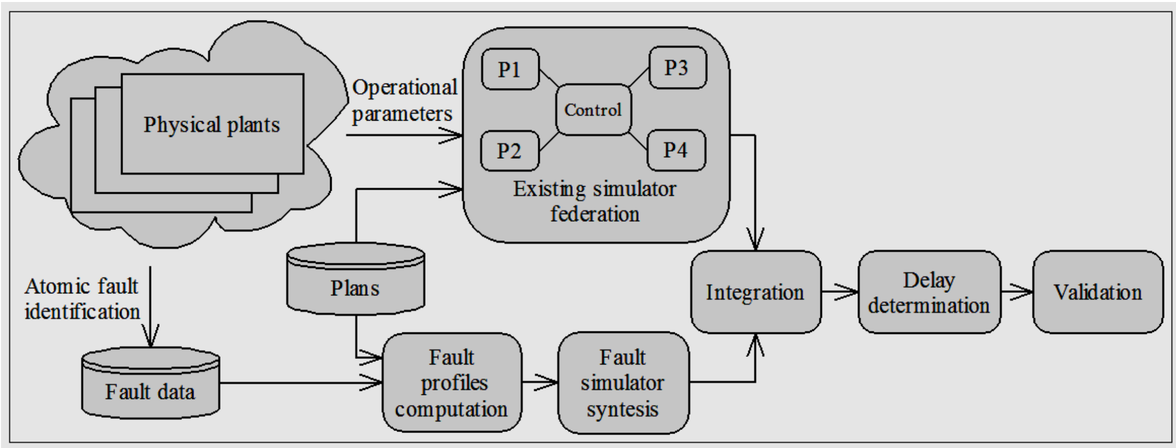

Figure 2. The steps of the methodology and its data structure

The second strategy to address the problem is based on AI techniques and specifically relies on the anticipatory capabilities of Artificial Neural Networks (ANNs). The networks, trained and tested on a large enough set of data, generated running the baseline federation and recording the actual times, types and location of failure occurrences, produces series of “next failure” events based on the characterisation of the last failure and on the current status of the federate plants.. Because the number of input parameters may easily become very large, as the complexity of the federation is increased, the implementation of this AI-based approach requires the development and training of multiple networks, ideally one for each federate simulator. The performance of this approach highly depends on the design of the network architecture and on the choice of learning algorithms, while its implementation is subject to a time-consuming phase of data sets generation for the training and testing of each individual network. The benefits of this approach lie in its modularity: changes in one of the federate plants only affect the relevant ANN, which can be separately re-trained, while all the other ANNs are entirely re-usable. The second step addresses the impact of inter-system model dependencies at the production level. The approach based on FTA techniques follows the basic steps illustrated in the previous section. An ANN-based approach is still applicable but quite impractical as far as its implementation because the patterns of failure/plant reconfiguration are too numerous even when considering rather simple plant federations, thus making the training and testing phases both time consuming and possibly ineffective. The third step consists in the design of an additional simulator (Fault Simulator, FS) in charge of evaluating the current status of all the federates and the current production plan. Based on this information, the simulator determines the expected failure modes of the system and generates the corresponding PF events. A block descriptor of the additional simulator is sketched in figure 3. The figure represents the simulation control for the federation, the existing simulators (bundled together as a single block because of their substantial homogeneity) and the flow of required information as they are exchanged between blocks. The whole process, as also visible in figure 2, takes as input production plan information as well as system fault data. The FS is designed as a state machine: the internal state of this simulator is determined by the actual failure characteristics of the real plant in its current production configuration; its input is the system status (including the production plan), and its output is the set of PF events. HLA-compliance provides great advantages during the implementation phase of this simulator: this new simulator can either be built using the same simulation engine used for the other federates or, alternatively, it can be built from scratch in compliance with the federation standard.

The second solution allows for increased flexibility at the cost of explicit management of HLA compliant communications. A suitable approach for the fourth step is heuristic in nature and based on statistical inference. In each

DOI 10.5013/IJSSST.a.22.02.01

ISSN: 1473-804x online, 1473-8031 print

federate simulator the representation of machine failures relies on the random extractions of punctual values from the corresponding MTBF (Mean Time Between Failures) distributions. A possible approach for the generation of failure events at the federation level involves the isolation of the machines that are critical to the production process (e.g. those that have blocking consequences for the entire production line): for each one of such machines, multiple (typically 10 to 20) extractions are performed and averaged at time t_0. The machine characterized by the smallest average $MTBF_0$ is the first one to fail at time t_1 obtained subtracting from the average a fraction of the corresponding standard deviation (of the order of 50%). The same procedure is iterated starting from time t_1 until the desired simulation horizon is covered. The generic expression for time t_{i+1} is $t_{i+1} = t_i + (MTBF)_i - \sigma_i$, where $(MTBF)_i$ = min (avg_MTBF_1… avg_MTBF_N), N = number of critical machines in the federation, σ_i = standard deviation corresponding to $(MTBF)_i$. A variation, which is faster to implement but less accurate, is that of choosing a fixed time interval between expected failures equal to the smallest mean of the MTBF distributions associated to the critical machines. The fifth step can be implemented based on well-established VV&A techniques [Sargent, 1999] using data obtained from the real plant as a reference, while the performance benefits of the new simulation system can be assessed with respect to the baseline federation.

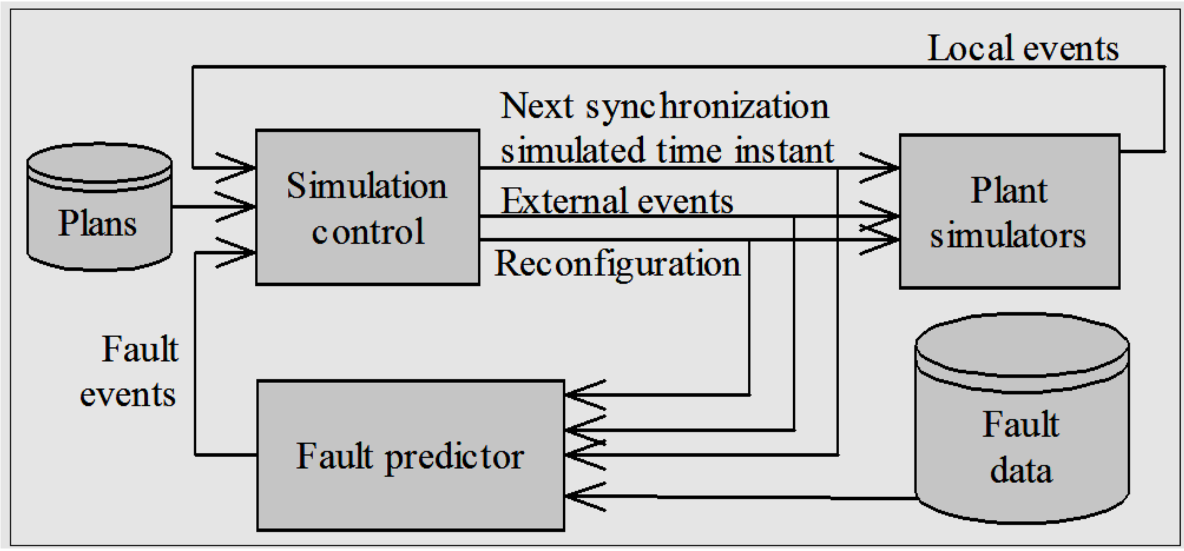

Figure 3. Block diagram of the additional simulator

V. EXAMPLE APPLICATION

In order to test the methodology a simple application case was designed to analyse the impact of the predicted failure approach on the time-performance of the baseline federation. The application context, or in other words the physical system, is a flexible manufacturing plant consisting of eight work centres, two for each type of operation/machining job. The modelling of machine failure events associated to each work centre is based on the MTBF and MTTR probability distributions of each machine. Random extractions out of each MTBF distribution, according to the procedure illustrated as step 4 of the methodology illustrated in the third section, concur to determine the time of the next failure event to be scheduled at the control level.

The performance measure used to compare the time-performance of the new, improved federation to that of the baseline federation is the ratio between the respective communication delays taken as averages over the simulated production time. In order to account for the variability of the measured time delays, a 95% confidence interval was set around the measured average and using its lower and upper bound as reference, a range of percentage waste reduction was obtained. The application of the proposed methodology to the industrial case described above leads to a reduction of wasted simulated time in the range of approximately 28-35%. Wasted time due to delays in information transfer is tightly coupled both to the topology of the plant and to the characteristics of the production plan.

VI. CONCLUSION AND FUTURE WORK

This paper presented a methodology to enhance the time-performance of federate simulations in industrial scheduling applications. The methodology exploits the available knowledge on the failure characteristics of the manufacturing system at the cost of the implementation of one additional simulator, however it preserves the existing simulator federation thanks to HLA standardization.

While preliminary results on a simple application case show a relative improvement in the runtime performance of the federation, ongoing studies are addressing the issue of optimising the timing of the "added-in" events so as to reflect as closely as possible the actual events recorded during the simulation run. Future work includes application to a more complex real-world manufacturing system as well as a further investigation on the steps of fault model identification and heuristic delay determination.

REFERENCES

[1] Bandinelli R., M. Rapaccini, S. Terzi, and M. Macchi 2003, "Proposal for a Framework for Production Plants Remote Control: a Preliminary Test Case". In *Proc. 15th ESS Conf.* (Delft, NL) Pp343-348.

[2] Chang Y and H. Maskatsoris 2001 "Supply Chain Modeling using Simulation". *International Journal of Simulation Systems, Science and Technology, 2*(1) Pp 24-30.

[3] Gunasekaran A., C. Patel and E. Tirtiroglu 2001, "Performance Measures and Metrics in a Supply Chain Environment". *International Journal of Operations and production Management, 21*(1) Pp71-78.

[4] Hwang K. 1993, "Advanced Computer Architecture – Parallelism, Scalability, Programmability". (New York, NY: McGraw-Hill).

[5] Juhasz Z., S. Turner, K. Kuntner and M. Gezson 2003, "A Performance Analyzer and Prediction Tool for Parallel Discrete Event Simulation". *International Journal of Simulation Systems, Science and Technology, 4*(1) Pp7-22.

[6] Orsoni A., A. Bruzzone and R. Revetria 2003, "Framework Development for Web-based Simulation Applied to Supply Chain Management". *International Journal of Simulation Systems, Science and Technology, 4*(1) Pp1-6.

[7] Proud F.J. 1999, "Master Scheduling. A Practical Guide to Competitive Manufacturing". Second edition (New York, NY: J.Wiley & Sons).

[8] Sargent R.J. 1999, "Validation and Verification of Simulation Models". In *Proc. Winter Simulation Conf.* (Piscataway, NJ) Pp39-48.

[9] Williams E.J. and R. Narayannaswamy 1999, "Application of Simulation to Scheduling, Sequencing, and Material Handling". In *Proc. Winter Simulation Conf.* (Piscataway, NJ) Pp861-865.

AUTHORS' BIOGRAPHIES

Romeo Bandinelli took his Laurea Degree in Mechanical Engineering at Florence University in April 2002 discussing the thesis "Remote Factory Control with Distributed Simulation". Actually, he is a PhD student of University of Florence, Department of Energetic, Plants and Industrial Technologies Section. His current research interests are Parallel and Distributed Simulation applied to industry and supply chain context, ICT, business process re-engineering.

Mauro Iacono is adjunct professor and research associate at the Second University of Naples (SUN). He obtained his Laurea degree in Ingegneria Informatica in 1999 from the Federico II University of Naples. After that, the joined the SUN where he received his PhD in Electronic Engineering in 2002. His current research interests are: performance evaluation by multi-formalism modelling techniques, complex systems and reactive systems engineering.
He has been a consultant of the Italian Ministry for Innovation and Technology (MIT) within the e-government national plan.

Alessandra Orsoni is currently a senior lecturer in the School of Business Information Management at Kingston University (Kingston, UK). She received both her MS in Mechanical Engineering and her ScD in Engineering Systems Design and Innovation from the Massachusetts Institute of Technology (Cambridge, MA). Prior to joining Kingston university she was a research associate in the Department of Materials Science and Metallurgy at the University of Cambridge (Cambridge, UK) and worked as an assistant professor in the Department of Production Engineering at the University of Genova (Genova, Italy). Her research interests include modeling, simulation, and AI techniques applied to the design and management of complex industrial systems
.