COMPARING CONVENTIONAL AND DISTRIBUTED APPROACHES TO SIMULATION IN A COMPLEX SUPPLY-CHAIN HEALTH SYSTEM

Korina Katsaliaki[1], *Navonil Mustafee*[2], *Simon J. E. Taylor* [2] *and Sally Brailsford*[3]

[1]Business School,
Middlesex University
London NW4 4 BT, UK
k.katsaliaki@mdx.ac.uk,

[2] Centre for Applied Simulation Modelling
School of Information Systems, Computing and Mathematics
Brunel University
Uxbridge, Middlesex, UB8 3PH, UK.
navonil.mustafee@brunel.ac.uk, simon.taylor.brunel.ac.uk

[3] School of Management,
University of Southampton
Southampton SO17 1BJ, UK
korina@soton.ac.uk, s.c.brailsford@soton.ac.uk

ABSTRACT:

Decision making in modern supply chains can be extremely daunting due to their complex nature. Discrete-event simulation is a technique that can support decision making by providing what-if analysis and evaluation of quantitative data. However, modelling supply chain systems can result in massively large and complicated models that can take a very long time to run even with today's powerful desktop computers. Distributed simulation has been suggested as a possible solution to this problem, by enabling the use of multiple computers to run models. To investigate this claim, this paper presents experiences in implementing a simulation model with a "conventional" approach and with a distributed approach. This study takes place in a healthcare setting, the supply chain of blood from donor to recipient. The study compares conventional and distributed model execution times of a supply chain model simulated in the simulation package Simul8. The results show that the execution time of the conventional approach increases almost linearly with the size of the system and also the simulation run period. However, the distributed approach to this problem follows a more linear distribution of the execution time in terms of system size and run time and appears to offer a practical alternative. On the basis of this, the paper concludes that distributed simulation can be successfully applied in certain situations.

Keywords: Distributed Simulation, Supply Chain Systems, Healthcare Operations, Simulation software, Simul8.

Running title: Comparing conventional and distributed simulation

INTRODUCTION

At its simplest, a supply chain is the entire process by which a product is manufactured and sold, starting "upstream" with the procurement of raw materials and moving downstream through manufacture, assembly, distribution, sale and support (Stevens, 1989). However, supply chains are in reality much more complex (Surana et al., 2005). For example, there are backwards flows of information that influence the behaviour of the chain and also supply and demand are often variable. Moreover, it is now recognised that the supply chain functions much better if viewed as a whole, rather than by a "silo" approach where each component in the chain acts independently of its neighbours (Cooper and Ellgram, 1993). Modern supply chain management approaches favour a global, holistic view in which the individual echelons share information and trust each other, rather than simply trying to optimise their own local processes (Chapman and Corso, 2005). Most of these multi-echelon and complex supply chains, including that of blood, can benefit from Operational Research techniques. Among the quantitative methods, simulation is undoubtedly one of the most powerful techniques to apply, as a decision support system, within a stochastic supply chain environment (Terzi and Cavalieri 2004) Discrete-event simulation is a commonly used modelling technique (Robinson, 2005; Law and

Kelton, 2000; Pidd, 1998) which has been used to analyse supply chains (Banks et al., 2002). It is a tool which can provide multi-decisional support in the context of "what-if" analysis and evaluation of quantitative benefits.

Practitioners in this area typically use visual interactive modelling environments or *Commercial Off-The-Shelf* (COTS) *Simulation Packages*, such as AnyLogic™, Arena™, Simul8™ and Witness™, that exploit developments in visual interactive modelling to facilitate, for example, model building and experimentation (Swain, 2003). The size of a supply chain can be potentially quite large and can consist of many complex elements. The simulation of a supply chain can therefore demand the creation of large models that, it is argued, are beyond the capability of a single computer to simulate (Taylor, et al., 2002; Lendermann, et al. 2001; Gan, et al. 2000). This claim is also one of the main drivers for distributed simulation, a technique where models are implemented over many computers in a parallel or distributed fashion with the goals of reducing the execution time of a single simulation run, sharing the memory needs of a simulation across several computers and the linking of simulations sited in different locations (Fujimoto, 1999; 2003).

Katsaliaki and Brailsford (2006) describe experiences with the use of simulation to investigate the supply chain of blood from the National Blood Service (NBS) Centre to hospitals. The purpose of the experiments is to achieve improved inventory control, ordering and distribution policies, which will bring less wastage, less shortage and provide better quality service. This is was done by reconfiguring the processes and parameters of the system. A problem identified in this work is that as the system being modelled grows in size and complexity, the time taken to perform one simulation run increases to a point that makes the use of simulation infeasible. Distributed simulation, as presented in literature, appears to offer a solution to this problem by sharing the processing of the supply chain model over several computers. Note that this is far more complex than just executing individual replications or experiments on different computers. As will be seen in our paper, when a model is split into "submodels" that run on different computers, the simulation of these submodels must be coordinated so that every event in the simulation is executed in correct order. This is a non trivial problem and it is the aim of an international standards group to develop a generalised solution. To investigate this claim, this paper therefore presents experiences in implementing a “conventional” simulation in a distributed setting. The paper first introduces the supply chain of blood and the conventional simulation approach that was taken. The distributed approach is then described. The paper then compares the two approaches on the basis of simulation run times for increasingly larger models. Results are presented and it is shown, at least in this case, that run times for the conventional approach appear to increase almost linearly with the size of the system and also the simulation run period. However, in the distributed approach, run times follow a more linear distribution of the execution time in terms of size and run time. After a brief discussion, the paper concludes that this technique can be successfully applied in certain situations to the simulation of large and complex supply chains and further research is needed to investigate the generality of this technique. We hope that the contribution of this paper will add to the growing body of literature aimed at developing a generalised solution to distributed simulation in this area.

BACKGROUND TO THE SUPPLY CHAIN OF BLOOD

The initial study was carried out in collaboration with the National Blood Service (NBS). The NBS consists of 15 Process, Testing and Issuing (PTI) Centres which together serve 316 hospitals across England and North Wales. Each PTI Centre thus serves around 20 hospitals. We worked in particular with the Southampton PTI Centre.

The NBS collects whole blood by voluntary donation, mainly from local venues such as church halls or places of employment. The blood is transported back to the nearest PTI Centre where it is tested for ABO and Rhesus grouping and infectious diseases such as HIV. A unit (450ml) of whole blood is then processed into around 115 different products, of which the main three are red blood cells (RBC), platelets and plasma. RBC have usually a shelf life of 35 days and platelets of 5 days, but plasma can be frozen and stored for up to a year. In this study we include only RBC and platelets, and these are two of the basic entity types in our model. RBC and platelets together comprise 85% of issues and are the chief source of wastage and shortages.

Blood products are stored in the PTI Centre’s blood bank until they are requested by the hospitals served by that Centre. There are mainly three types of delivery: the routine scheduled delivery, which is made on a daily basis on a multi-stop mode serving many hospitals at once and is free of charge; emergency delivery, prioritized for immediate dispatch; ad-hoc delivery, additional to routine deliveries and made to an individual hospital if it places an order. There is also a nationally coordinated scheme for transferring excess stock

The ordering system is highly complex. Local practice varies and all hospitals have slightly different ordering policies. Hospitals determine their own optimal stock levels according to their estimates of demand. An order is placed with the local PTI Centre when inventory falls below a predetermined order point, or when rare products not held in stock are requested for particular patients. Different types of order can be placed, each with different associated costs.

Individual doctors are responsible for the quantity of blood products ordered for each patient in the hospital. In theory, doctors order blood according to the Maximum Surgical Blood Ordering Schedule (MSBOS) (BCSH, 1996) which specifies how much blood is required for a given operation. The MSBOS is conservative, to allow for cases where extra blood might be needed if complications arise, but many doctors still over-order to be on the safe side. Patients should ideally be given blood of the same type but “mismatching” is possible in emergencies – for example, O-negative blood can be given to anybody.

Hospitals normally receive their orders daily and the blood remains in the hospital bank until it is cross-matched (tested for compatibility) for a named patient. It is then placed in “assigned inventory” for that patient for a fixed time after the operation. If it is not used, it is returned to “unassigned inventory” and can be cross-matched again for another patient. On average a unit will be cross-matched three to four times before it is used or outdated. In practice, only half of the cross-matched blood is actually transfused. This clearly represents a huge potential for savings since the cost of a single unit of RBC is around £132.

THE CONVENTIONAL APPROACH

This system is clearly stochastic since the demand for blood is variable (even for elective surgery) depending on the type of operation and the occurrence of complications requiring extra transfusions. The supply is also variable since it relies on volunteers showing up to donate. Other organisational issues arise since the NBS manages the supply side but the hospitals manage the demand side. Discrete-event simulation was chosen to investigate the problems of this supply chain as complex stochastic multi-product, multi echelon perishable inventory problems have been shown to be intractable by analytic techniques (Donselaar et al, 2006; Goyal and Giri, 2001). Simulation was also the technique which the majority of the researchers have adopted to tackle this problem since some decades ago.

The model was built using the Simul8™ COTS simulation package (www.simul8.com). The supply chain model is very large and complex, and requires extensive data. Nineteen months' data from the Southampton PTI Centre was provided and analysed using the NBS information system PULSE. This gave details of the products supplied to each hospital, by date, time, delivery type, quantity and blood group. Questionnaires were sent to the hospitals supplied by the Southampton centre, and interviews conducted with NBS staff and hospital blood bank managers. There are two main categories of entities in the model; items and orders. Items are the individual blood units (RBC and platelets) delivered from the NBS Centre to the hospitals in a one-way direction, since returns of products are not allowed. Orders are placed by the hospital blood bank mangers to the NBS Centre for blood products, and represent the backwards flow of information. Requests are matched with items according to their characteristics (attributes) as in a Kanban system and delivered as appropriate.

While the model runs, data are reported in an Excel file, such as the day and time of placing an order with the Centre, the type of order (routine, ad-hoc or emergency), the requested product and the amount by blood group. The model time units are minutes, and the remaining shelf-life of blood products is counted in minutes. However, the hospitals' blood bank stock for placing orders to the NBS is checked only every hour. Moreover, the decision to run the model in minutes was enforced by the fact that many processes, such as physician requests and delivery times, could be better approximated in small units of time. In addition, an attempt to run the model in hours did not significantly accelerate the overall running time.

The smallest version of the model contains the processes of the NBS Centre, from collection of whole blood to delivery of blood products, and the processes within a single medium-volume hospital. The model captures physicians' requests for blood and the processes whereby the hospital blood bank checks its stock levels and places orders. Figure 1 shows a simplified illustration of this simulation model. For multiple hospitals, Figure 2 shows an example of the relationships between the NBS supply centre and the hospitals it serves, which in the "conventional" approach is simulated on a single computer.

Figure 1 and 2 about here

In our conventional, single computer approach the execution times were as follows. A single NBS supply centre with a single hospital as outlined above took approximately 14 minutes to run for a whole simulated year in a 1.7GHz processor desktop PC with 1GB RAM. The runtime rises dramatically when we add more hospitals to the model. For a model with a single supply centre and two hospitals the execution time was 78 minutes, with three hospitals it was 17.5 hours and for a single supply centre and four hospitals the execution time was 35.8 hours (even after considerable help and advice from the package vendor on model profiling). The enormous number of entities in the system, each of which carries many attributes, increases the computation time exponentially in the beginning and then linearly even though there are no such elements in the functions of the model to dictate the particular size of the increase. This increase results from a combination of two different factors. Firstly, the behaviour of the system being modelled is such that all entities (blood units) in the system have a limited shelf life. This behaviour is modelled in the NBS simulation by continually scheduling events that decrease the shelf life of each entity by the minute. This results in an increasing number of computations as the number of entities flowing through the system increases. Further, this also contributes to a large event list that takes an increasing amount of time to manipulate when new events are scheduled. Secondly, the large event list and large amount of information generated by the model further exacerbates this processing load as these cannot be accommodated in the computer's random access memory (RAM) alone. When RAM is exhausted, a computer's operating system will use "virtual memory", an area of storage on the computer's hard disk, to store additional data. Moving data between RAM and virtual memory is slow and adds to the processing time of any computation. Thus, as a model and its event list becomes large, more information is generated resulting in more transfers between the RAM and the virtual memory, thereby contributing to a further increase of execution time.

We now consider a distributed computing approach to this problem of long execution time.

THE DISTRIBUTED APPROACH

How can we distribute our "conventional" model across several computers? The general approach taken by the field of distributed simulation is to divide and modify a model so that parts of the model (effectively submodels) reside on different computers linked together by a communication network. (Note that this is very different to running separate replications or experiments on different computers) These "submodels" become *Logical Processes* (LPs) or *federates* that interact by timestamped messages that represent the interaction of one submodel with another (say, when an entity leaves one part of a model and arrives at another) (Chandy and Misra, 1979; Fujimoto, 2001). More recently, the "practice" of distributed simulation has been effected on a wide scale by attempts to standardise approaches. These standardisation efforts took place in the mid-to-late nineties and resulted in the IEEE 1516 standard *The High Level Architecture* (HLA) that supports the general needs of distributed simulation (IEEE 2000). This is rooted in the early work by Chandy and Misra but considerably updates it with advances in distributed computing. The HLA defines the runtime infrastructure (HLA-RTI) software and the format of the data that is used by a collection of models (federates) running on different computers to interact (such a collection of models is termed a federation). As we use the HLA in this paper, we will use the HLA term *federate* to refer to submodels running on different computers.

There have been several attempts to create distributed simulations of manufacturing systems and supply chains using the HLA. The first major work in the area was done by Straßburger (2001). Various strategies have been investigated since then. For example, McLean and Riddick (2000), Mertins et al. (2000), Hibino et al. (2002), Linn et al. (2002), Rabe and Jakel (2003), Straßburger et al. (2003) and Taylor, et al. (2005^a) discuss the use of the HLA to support the distributed simulation of manufacturing systems. This work, while contributing to the development of distributed simulation, presents a problem. Virtually every author cited above uses *different* approach to distributed simulation of manufacturing and supply chains. In an attempt to standardise the approach, an international standardisation group, Simulation Interoperability Standards Organization's COTS Simulation Package Interoperability Product Development Group (CSPI PDG) (www.cspi-pdg.org), have produced a set of draft standards in this area and are described in Taylor, et al. (2005^b) and Taylor, et al. (2006). Related work in this area include, Wang, et al. (2005) discuss a general experimental test-bed used to investigate these standards. Gan, et al. (2005) present a case study in the use of the standards to support semiconductor supply chain analysis using the COTS simulation package Autosched AP™. The distributed approach used in this paper is based on the draft standards being developed by the CSPI PDG and can be found in full in Taylor, et al. (2006).

In our distributed NBS simulation the HLA-RTI is presented as a black box. Figure 3 shows the conventional model of figure 2 using our distributed approach. To give a brief overview of how the distributed approach works, consider a normal discrete-event simulation. Such a simulation uses an algorithm (simulation executive) to repeatedly select the next event from the event list, process it and then places any new events that have been generated in the correct order on the event list. Events on an event list are held in ascending time (or timestamp) order and we say that a simulation is correct if these events have been processed in the right order. In a distributed simulation, this becomes more complex. If a submodel (federate) generates an event that takes place at another submodel at a given timestamp (say, an entity leaves one model and arrives at another submodel at the timestamp), then we must guarantee that that event is processed at the correct time in *the receiving submodel*. This is difficult to do as in distributed simulation we typically do not have a global clock, i.e. each submodel advances in simulation time individually (this is done in the hope of achieving better performance). Events sent between submodels are done so by using timestamped event messages. The correct processing of events is guaranteed by using an appropriate synchronisation protocol that ensures the event represented by a timestamped event messages are processed by a receiving submodel in the correct order. In the HLA, the synchronisation protocol is implemented in the HLA-RTI. The HLA allows several alternatives to send and receive event messages. Our approach uses HLA *interactions* as suggested by the CSPI PDG and are described more generally by (Kuhl et al, 1999). In summary, each submodel (federate) represents either the NBS supply centre or a hospital running in a separate copy of Simul8 running on a separate computer. When a submodel determines that an entity representing orders or blood is to be "sent" to another submodel, a timestamped event message is sent to that submodel. Operationally, the event message is sent as a HLA interaction and the HLA-RTI ensures that the receiving submodel processes the event (the arriving entity) in the correct order with its own events. Let us now consider the performance of this distributed approach.

Figure 3 about here

EVALUATION OF EXECUTION TIMES

To investigate the distributed approach against the conventional approach, four scenarios were investigated. These were one NBS supply centre serving one, two, three and four hospitals respectively. Before experimentation commenced, the outputs for the conventional and distributed models were compared to check that the same results for a year's run was produced (except as noted below). This was done to check that the minor modifications to link Simul8/Excel/HLA-RTI in the distributed model did not artificially increase/decrease the workload. All experiments were conducted on Dell Inspiron laptop computers running Microsoft Windows XP operating system with 1.73GHz processors and 1GB RAM connected through a 100Mbps CISCO switch. The same computer specifications were used to guarantee consistency in runtimes. The results of the execution times for each of the models are based on the average of 5 runs. The deviation from the average of the total execution time between the runs was small. This deviation was less than 2.5% for each of the 5 runs in the 1% significance level except for the conventional model with one NBS supply centre and four hospitals which was around 19%.

Furthermore, the hospitals which were added to the models were all of the same volume (medium). For instance, physician requests were around 1000 blood units for each hospital per month, with each hospital diverging by a small percentage (≤6%) from the mean. Accordingly, the distribution parameters for every hospital were slightly different. Also, every hospital adhered to its own rules about routine ordering times, re-ordering stock points, time of returning unused units to the Blood Bank, etc. The reason for this imprecision was to ensure a spread of courses of action happening during the day, such as the time the orders are matched with the units, a fact that affects the performance of the models including the runtime.

As previously noted, the conventional model with one hospital took approximately 14 minutes to run for a whole simulated year. The run time rose to 78 minutes when the model ran with two hospitals and to approximately 17.5 hours with three hospitals. The addition of the fourth hospital rockets the execution time to 35.8 hours. The distributed model with one NBS supply centre and one hospital ran in approximately 8.5 hours, with two hospitals in 9.8 hours, with three hospitals in 12.7 hours and with four hospitals in 16.5 hours. Figure 4 shows these execution times in seconds for both methods.

Figure 4 about here.

DISCUSSION

It is apparent that the versions with one or two hospitals are less time consuming to run using the conventional approach. Conversely, when a third and fourth hospital are added then the distributed method bests the runtime of the conventional approach. There also appears to be a high escalation of the runtime in the conventional version while increasing the number of hospitals in the model. This is quite a contrast to the substantially smaller and smoother rise in the runtime in the distributed method. Further, a more exhaustive analysis of the results reveals another significant feature. Every model for each method was monitored for its execution time per simulated month until the end of the run (1 year) as Figure 5 shows. The graph clearly demonstrates that for the conventional method there is an upwards incremental trend in the runtime per added month. Especially for the model with one NBS supply centre and three hospitals, the monthly runtime rockets up from month 10 and over. For the model with four hospitals, this trend is apparent right from the first month, for the reason which was explained in the last part of the conventional model section. The fluctuations in the runtimes between consecutive months are due to random variation.

Figure 5 about here.

Figure 5 shows the percentage difference of the time taken for the conventional models to execute every consecutive month of the year versus the corresponding difference of the distributed models. Figure 6 shows, for the conventional models, the percentage difference of the runtime between each month and the month that took less time to run. The latter is equal to 0%. The month that took less to run in the first two models (with one NBS supply centre and one or two hospitals) is month four, and for the three hospital and four hospital models is the first month. It can be seen that the Y axis scale takes values from 0% to 1100% while in Figure 7 the equivalent range for the distributed models is from 0% to 71%. The maximum percentage differences for one to four hospital models using the conventional method against the distributed method are 53% to 5% (10.6:1), 315% to 8% (39.4:1), 1100% to 26% (42.3: 1) and 176% to 71% (2.5:1) respectively. The small ratio observed in the last case is due to the fact that the model takes considerably more time to run even from the first month.

Figures 6 and 7 about here.

These findings indicate that for the conventional method an expansion in model size will be accompanied by an approximate exponential increase in both the total runtime and the time between iterations when the results are being collected. On the other hand, for the distributed method an increase in the number of hospitals (and therefore of computers) will be followed by a much smaller increase in total runtime, with no extensive increase in the time between iterations. Therefore, if more than two hospitals are added to any model, the distributed method would be a better platform in which to develop and run the simulation experiments. Overall, the distinctive trend that the two methods follow concerning runtimes seems to be continuous; in other words the more hospitals we add to the model, the more the differences in the runtimes between the two methods favour the distributed approach. As noted in the discussion of the conventional approach, the increase in runtime appears to be primarily due to the processing of a large event list caused by a combination of the volume of entities and the "counting down" of the shelf life. The large event list in turn causes a high number of transfers between RAM and virtual memory which, compounded by the processing of the event list, further causes long runtimes. Our results suggest that the distributed approach allows the processing and memory demands made by large event lists to be shared over several computers.

The complete model, in the case of the Southampton NBS Centre which we are concerned with, should include 16 hospitals. According to this study, it is clearly not feasible to run such a model in a single PC, but the use of distributed simulation allows us the possibility of running the full model.

CONCLUSIONS

This paper has described an investigation into using conventional and distributed approaches to simulating the supply chain of blood from a National Blood Service (NBS) Centre to hospitals with the COTS simulation package Simul8™. For this scenario at least, when the supply chain grows in size and complexity, distributed simulation appears to offer a viable alternative to conventional simulation by sharing the processing and memory requirements of the simulation across multiple computers.

The potential for such an approach in healthcare simulation modelling is interesting. There is an increasing recognition that healthcare systems do not exist in a vacuum and that even seemingly well-defined subsystems such as emergency departments, operating theatres or out-patient clinics have complex interconnectivity with other parts of the overall healthcare system, both within the hospital and outside its walls. This can lead either to the development of enormous models which attempt to capture these relationships, or to oversimplification by ignoring them and making the model boundaries artificially narrow. In this feasibility study we have demonstrated that distributed simulation offers a viable solution to this problem, using low cost off-the-shelf software which is widely available and increasingly used in the NHS.

In terms of further work, the key question is whether all large, complex simulations could equally benefit? The development of a distributed simulation requires some extra investment in time which is hopefully balanced by an increase in execution speed and therefore a decrease in the time taken to get results from experimentation with the simulation. However, it is not clear if this will be the case in all distributed simulations of this type. Further work is required to determine if such factors exist and how they can be determined *before* work on creating a distributed simulation begins. Previous work has indicated that the configuration of the supply chain described in this paper has the right characteristics to benefit from a distributed approach (Taylor, et al., 2005^{c}). It appears that the supply chain "topology", the relationship of the submodels and their interconnection (how the entities are passed between submodels), and the relationship between processing and synchronisation loads play and important part. For example, in our supply chain the simple topology and the high amount of processing done by each submodel relative to the amount of synchronisation required to correctly execute the distributed simulation mean that good performance was attained for the larger models. What of other supply chain configurations? The development of metrics to indicate what could be distributed and what should not be distributed will help bring this technology closer to the simulation practitioner.

ACKNOWLEDGEMENTS

The authors thank the following people for their time and their help with data and model validation: Crispin Wickenden and Andrew Oliver from the NBS; the employees of the Southampton PTI Centre, particularly Mike Northcott; Tracey Lofting from Southampton General Hospital; Rob Hick and Judith Chapman from the Blood Stocks Management Scheme. The authors would also like to thank Dr. Mark Elder (founder and CEO of Simul8 Corporation) for providing the Simul8 licences and his generous on-going support. Finally, the authors would like to acknowledge the contribution of Dr. Allan Tucker, John Saville and Dr. Steven Swift from Brunel University who generously offered assistance during the experimentation process.

REFERENCES

Banks J., Buckley S., Jain S., Lendermann P. and Manivannan M. (2002). Panel Session: Opportunities for Simulation in Supply Chain Management. In: Yücesan, E., et al. (ed). Proceedings of the 2002 Winter Simulation Conference. Association for Computing Machinery: USA, 1652-1658.

BCSH Blood Transfusion Task Force (1996). Guidelines for pre-transfusion compatibility procedures in blood transfusion laboratories, Transfusion Medicine, 6: 273-283.

Chandy K. M. and Misra J. (1979). Distributed Simulation: A Case Study in Design and Verification of Distributed Programs. IEEE Transactions on Software Engineering. 5(5): 440-452.

Chapman R.L. and Corso M. (2005). From continuous improvement to collaborative innovation: the next challenge in supply chain management. *Production Planning & Control* **16**: 339-344.

Cooper M.C. and Ellgram L.M., (1993). Characteristics of supply chain management and the implications for purchasing and logistics strategy. *International Journal of Logistics Management*, **4**:13-24.

Donselaar, K.H. van, Woensel, T. van, Broekmeulen, R.A.C.M., Fransoo, J.C. (2006). Inventory control of perishables in supermarkets *International Journal of Production Economics*, **104**(2): 462-472.

Fujimoto R.M. (1999). Parallel and distributed simulation, in *Proceedings of the 1999 Winter Simulation Conference*, IEEE, Piscataway, NJ, 122–131.

Fujimoto R.M. (2001). Parallel and Distributed Simulation Systems. In *Proceedings of 33rd Conference on Winter Simulation*, December 09 – 12, Arlington, Virginia. Winter Simulation Conference. IEEE Computer Society, Washington, DC, 147-157.

Fujimoto R.M. (2003). “Distributed Simulation Systems,” in *Proc. Winter Simulation Conference*, 124-134.

Gan B.P., Lendermann P., Low M.Y.H., Turner S.J., Wang X. and Taylor S.J.E. (2005). Interoperating Autosched AP using the high level architecture. In *Proceedings of the 2005 Winter Simulation Conference*. Association for Computing Machinery Press, New York, NY., 394-401.

Goyal, S., Giri, B., (2001). Recent trends in modeling of deteriorating inventory. *European Journal of Operational Research* **134**(1): 1-16.

Hibino H., Fukuda Y., Yura Y., Mitsuyuki K. and Kaneda, K. (2002). Manufacturing adapter of distributed simulation systems using HLA, in *Proceedings of the Winter Simulation Conference*, 1099-1107.

IEEE 1516 (2000). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA), NY: Institute of Electrical and Electronics Engineers.

Katsaliaki K. and Brailsford S.B.(2007). Using Simulation to Improve the U.K. Blood Supply Chain, *Journal of the Operational Research Society,* **58**: 219-227.

Kuhl F., Weatherly R. and Dahmann J. (1999). Creating Computer Simulation Systems, An Introduction to the High Level Architecture. Prentice Hall PTR.

Law AM and Kelton WD (2000). *Simulation Modeling and Analysis, 3rd ed.* McGraw-Hill: New York.

Lendermann P., Gan B.P. and McGinnis L.F. (2001). Distributed Simulation with Incorporated APS Procedures for High-Fidelity Supply Chain Optimization. In

Proceedings of the 2001 Winter Simulation Conference, Association for Computing Machinery Press, New York, NY., 1138-1145.

Linn R.J., Chen C.S. and Lozan J.A. (2002). Development of Distributed Simulation Model for the Transporter Entity in a Supply Chain Process, in *Proceedings of the Winter Simulation Conference*, 1319-1326.

Mertins K., Rabe M. and Jäkel F.W. (2000). Neutral Template Libraries for Efficient Distributed Simulation within a Manufacturing System Engineering Platform, in *Proceedings of the Winter Simulation Conference*, 1549-1557.

McLean C. and Riddick F. (2000) The IMS MISSION architecture for distributed manufacturing simulation, in *Proceedings of the Winter Simulation Conference*, 1539-1548.

Pidd M (1998). *Computer Simulation in Management Science, 4th ed*. Wiley: Chichester, UK.

Rabe M. and Jäkel F.W. (2003). On standardization requirements for distributed simulation in Production and Logistics. Building the Knowledge Economy, Twente, The Netherlands: *IOS Press*, 399-406.

Robinson S. (2005). Discrete-event simulation: from the pioneers to the present, what next? *Journal of the Operational Research Society* 56, 619-629

Stevens G.C. (1989). Integrating the Supply Chain, *International Journal of Physical Distribution & Materials Management*, **19**: 3-8.

Straßburger S. (2001). Distributed Simulation Based on the High Level Architecture in Civilian Application Domains, Ghent, Belgium: Society for Computer Simulation International.

Straßburger S., Schmidgall G. and Haasis S. (2003). Distributed manufacturing simulation as an enabling technology for the digital factory, *Journal of Advanced Manufacturing Systems* **2**(1): 111-126.

Surana A., Kumara S., Greaves M., Raghavan U.N (2005). Supply-chain networks: a complex adaptive systems perspective, *International Journal of Production Research* **43**(20): 4235-4265.

Swain J.J. (2003). Simulation Reloaded: Sixth biennial survey of discrete-event software tools, *OR/MS Today*, **30**(4): 46-57.

Taylor S.J.E., Bohli L., Wang X., Turner S.J. and Ladbrook J. (2005^{a}). Investigating Distributed Simulation at the Ford Motor Company, In *Proceedings of the Ninth IEEE International Symposium on Distributed Simulation and Real-Time Applications*. IEEE Computer Society, 139-147.

Taylor S.J.E., Turner S.J. and Low M.Y.H. (2005^{b}). The COTS Simulation Interoperability Product Development Group. In *Proceedings of the 2005 European Interoperability Workshop*. Simulation Interoperability Standards Organisazation, Institute for Simulation and Training, Florida, 05E-SIW-056.

Taylor, S.J.E., Turner, S.J., Mustafee, N., Ahlander, H. and Ayani, R. (2005^{c}) COTS Distributed Simulation: A Comparison of CMB and HLA Interoperability Approaches to Type I Interoperability Reference Model Problems. *SIMULATION,* **81**(1): 33-43.

Taylor, S.J.E., Wang, X., Turner, S.J. and Low, M.Y.H. (2006). Integrating Heterogeneous Distributed COTS Discrete-Event Simulation Packages: An Emerging Standards-based Approach. *IEEE Transactions on Systems, Man and Cybernetics: Part A*, 36, 1, 109-122.

Terzi S., Cavalieri S. (2004). Simulation in the supply chain context: a survey, *Computers in Industry* **53**: 3–16.

Wang, X. Turner, S.J. Taylor, S.J.E., Low, M.Y.H., Gan, B.P. (2005). A COTS Simulation Package Emulator (CSPE) for Investigating COTS Simulation package Interoperability. In *Proceedings of the 2005 Winter Simulation Conference*. Association for Computing Machinery Press, New York, NY, 395-411.

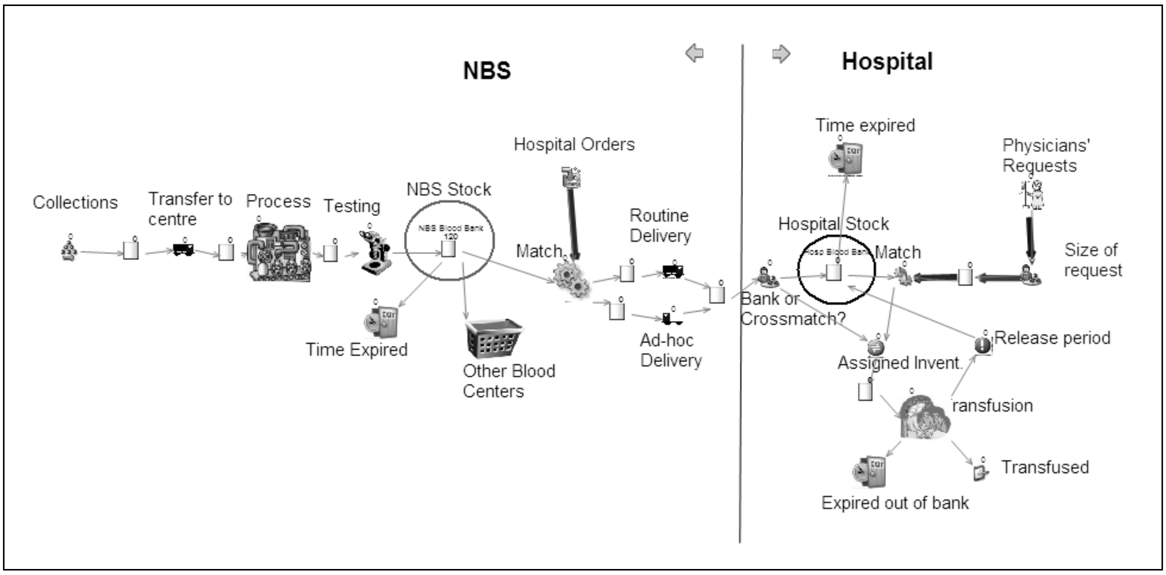

Figure 1 Screenshot of a simplified version of the Simul8 model showing one hospital.

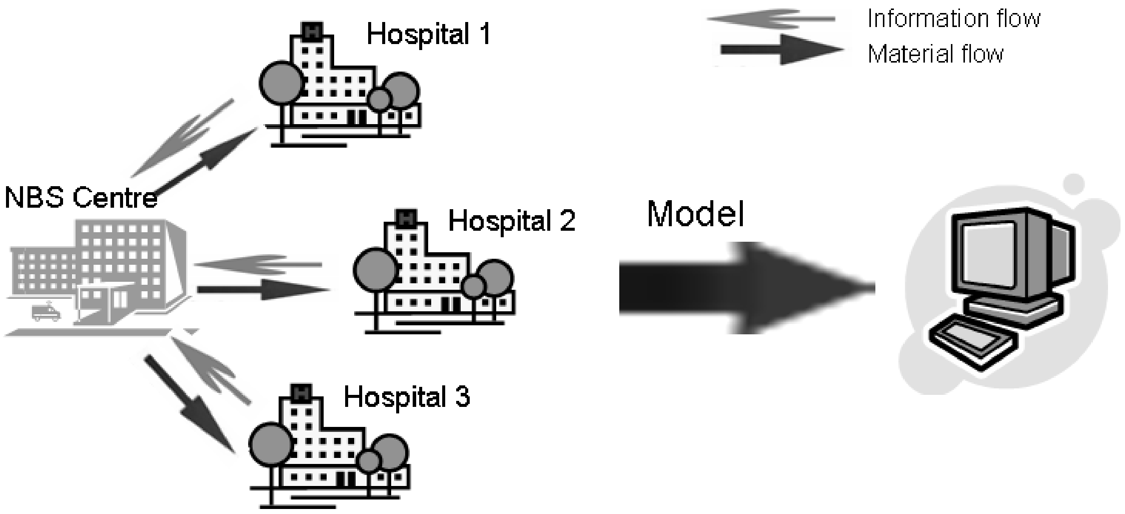

Figure 2 Conventional simulation approach.

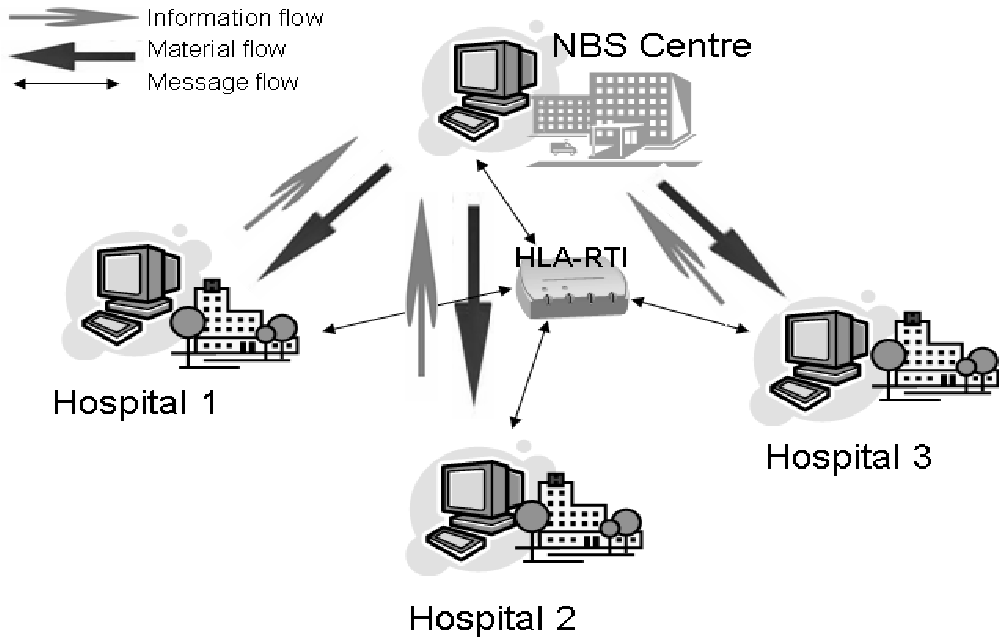

Figure 3 Distributed simulation approach.

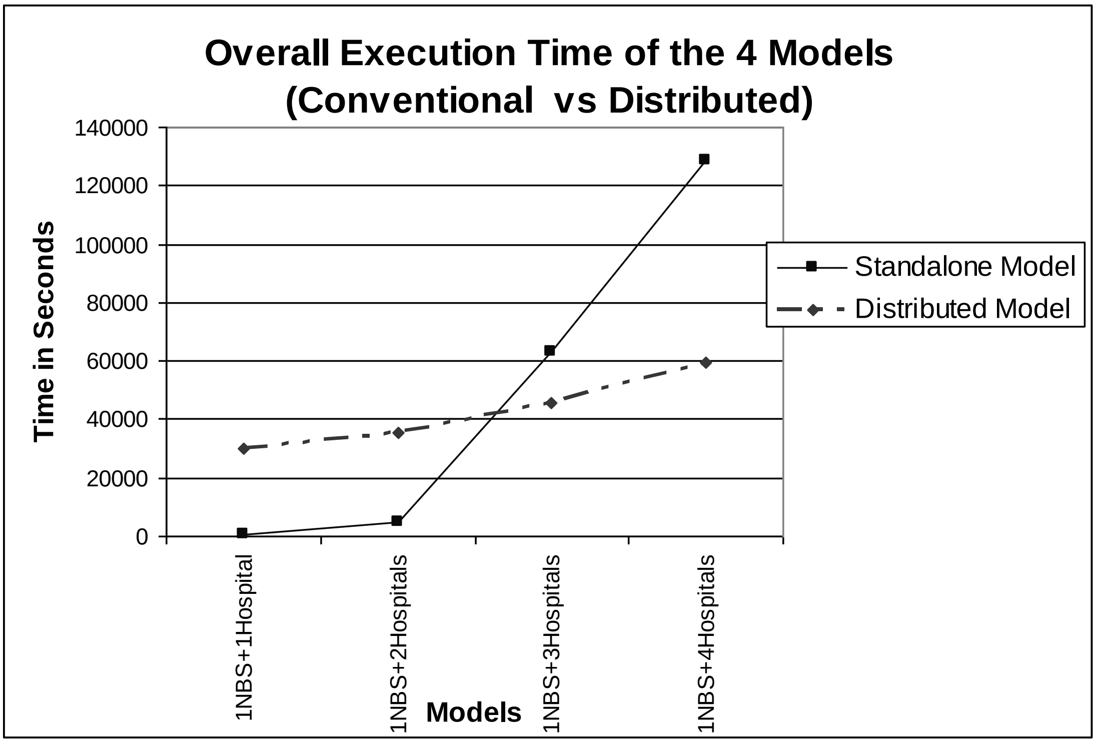

Figure 4 Runtimes of conventional and distributed method for one NBS Centre with one to four hospitals.

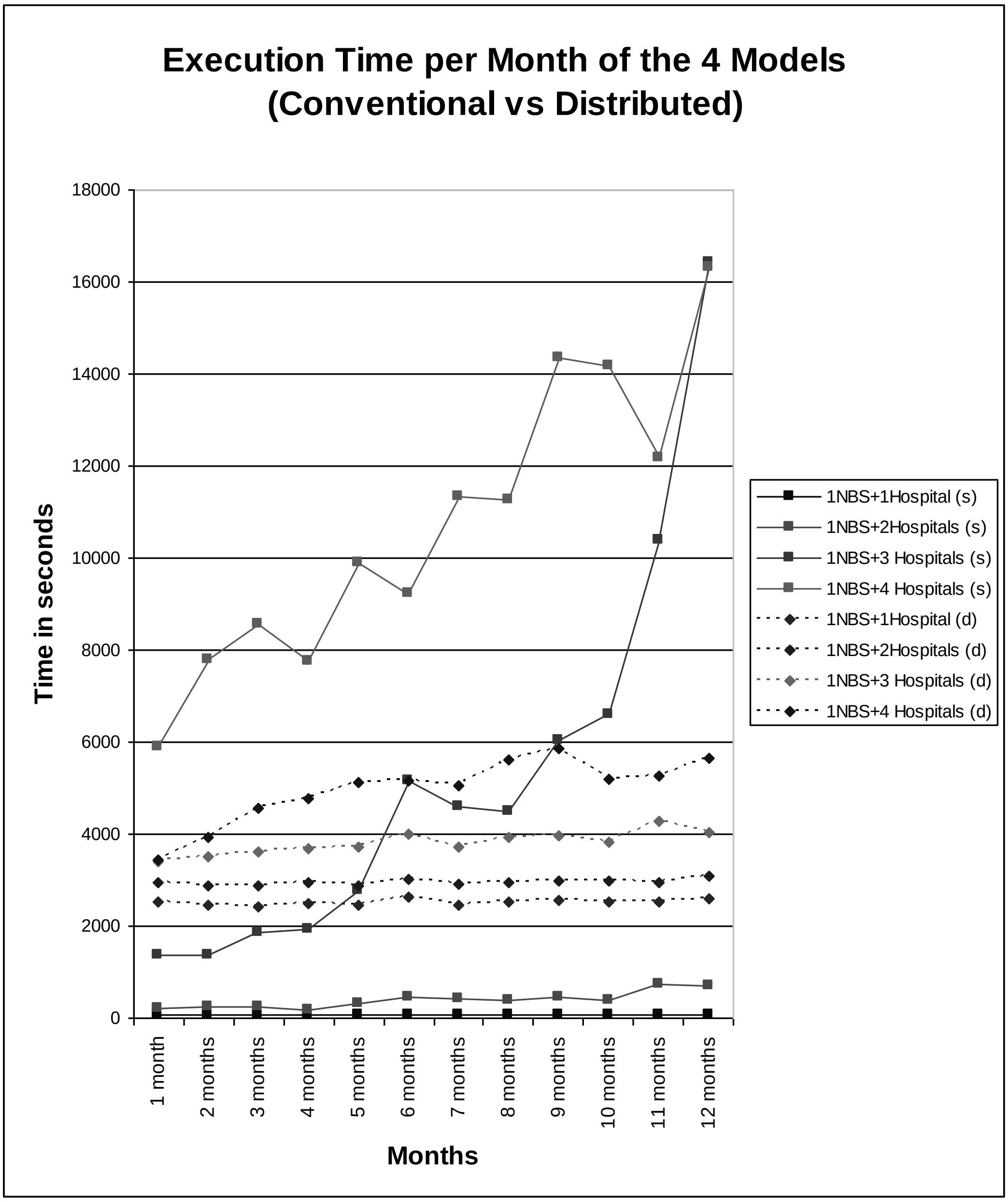

Figure 5 Execution time per month of the four models (conventional and distributed).

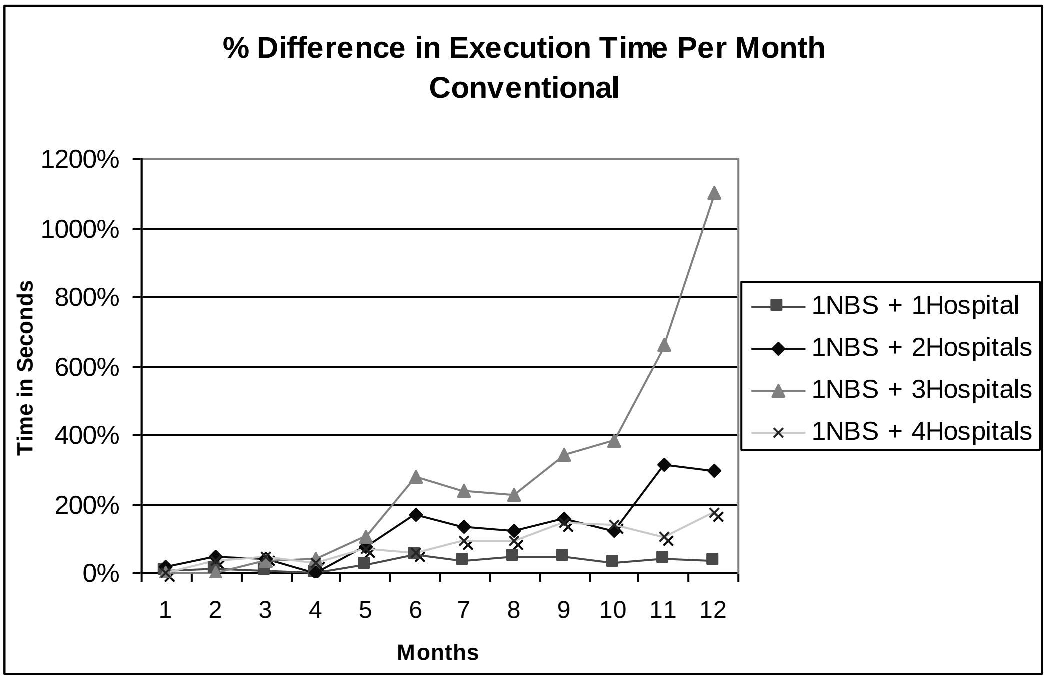

Figure 6 Execution time per month of the four models for the conventional method.

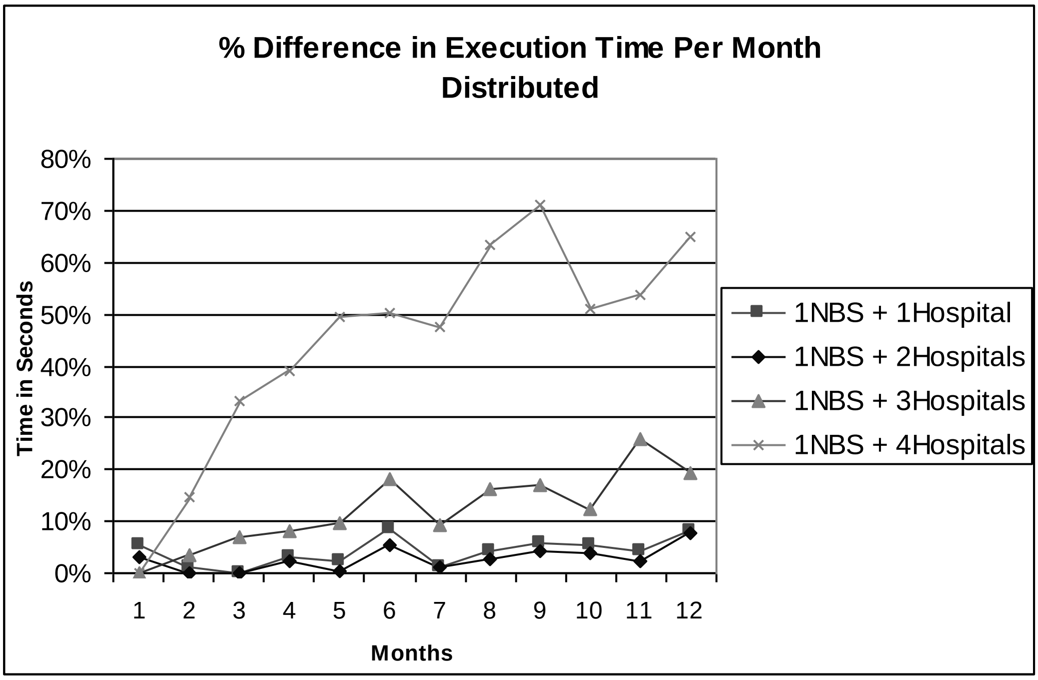

Figure 7 Execution time per month of the four models for the distributed method.

Figure 1 Screenshot of a simplified version of the Simul8 model showing one hospital.

Figure 2 Conventional simulation approach.

Figure 3 Distributed simulation approach.

Figure 4 Runtimes of conventional and distributed method for one NBS Centre with one to four hospitals.

Figure 5 Execution time per month of the four models (conventional and distributed).

Figure 6 Execution time per month of the four models for the conventional method.

Figure 7 Execution time per month of the four models for the distributed method.