PROXEL-BASED SIMULATION OF QUEUING SYSTEMS WITH ATTRIBUTED CUSTOMERS

Claudia Krull[(a)], Wenjing Xu[(b)], Graham Horton[(c)]

[(a)(b)(c)]University of Magdeburg, Department of Simulation and Graphics

[(a)]claudia@sim-md.de, [(b)]wenjing_de@yahoo.com, [(c)]graham@sim-md.de

ABSTRACT

This paper describes a state space-based simulation method for queuing systems with attributed customers. The approach extends a previous version, which was designed for only one customer class, enabling the simulation of a larger group of queuing models. The work is motivated by the need for exact solutions for queuing systems where no analytical solution is available. In this paper, the original Proxel-based queuing simulation method is extended to incorporate attributed customers, concentrating on efficient coding and storage strategies to dampen the state space explosion. The attribute classes priority, deadline and processing time are implemented. Experimental results indicate the maximum number of attribute values that is still feasible. Some interesting statistics are presented, which would be hard to obtain using traditional simulation methods. The presented method can yield deterministic results for a larger number of queuing systems that cannot be easily solved analytically.

Keywords: state space-based simulation, multiclass queuing systems, attributed customers.

1. INTRODUCTION

Queues are a part of our everyday life. Besides the obvious queues of human customers, they are widely used in telecommunications and computer systems. Therefore, queuing analysis is a well-researched area of mathematical modeling. The usual approach to a queuing problem is to devise an appropriate formal representation, find a match among the already known classes of queuing systems or create a new one and derive measures for the queuing system's performance analytically.

For a number of classes of queuing systems, no analytical expression for performance measures like throughput or server utilization is available. In such cases, simulation is the preferred way to tackle the problem. However, discrete event-based simulation may need to perform many replications to obtain estimates for the results of one single specified queuing system.

The recently developed method of Proxel-based simulation (Horton 2002; Lazarova-Molnar 2005) can feasibly conduct state-space based analysis of discrete stochastic systems and yield accurate deterministic results with reasonable effort. The Proxel results can compete with typical analytical ones, and the method is not inherently limited in the type of system.

Proxels have already been successfully applied to the simulation of queuing systems with one type of customer (Krull and Horton 2007). However, many queuing problems implement attributed customers to enable more advanced queuing strategies than FIFO.

The object of this paper is to introduce attributes into a Proxel-based queuing system simulation, test the resulting performance and show the suitability of the approach. If successful, more classes of queuing problems can be tackled using Proxel-based simulation. We believe that this could provide useful help for queuing analysts. Proxel-based simulation can provide deterministic solutions for a specific queuing systems performance measures, when no analytical solution is available. One can obtain rough estimates at low cost or results of arbitrary accuracy at higher computation cost. Furthermore, the method yields a transient solution of the system, which can for example help detect the possibility of infinite postponement when implementing SJF (*shortest job first*) as a queuing strategy.

2. STATE OF THE ART

2.1. Queuing System Analysis

Classical queuing analysis takes a real queuing system and derives a queuing model by identifying the arrival process, the service process and other system specifications. The performance measures for the queuing model are then calculated using known formulae for the given class of queuing system. The performance measures of a queuing system are usually expressed as scalar measures such as the system throughput. (Bolch et. al 2006, Gross and Harris 1998)

Queuing systems that contain generally distributed processes are often hard to handle analytically. For most of these classes no general solution exists. Discrete event-based simulation can be used to obtain stochastic estimates for these systems performance measures. However, unless a lot of computation cost is invested, the accuracy of these results is not comparably to analytical solutions.

An important application area of queuing analysis is the modeling of data traffic in networks, for example

in the internet. The models consist of multiple queuing systems that are connected to form queuing networks. As stated in Cremonesi, Schweitzer and Serazzi (2002) the traffic consists of customers with very different resource requirements (attributes), resulting in so-called multiclass queuing networks. The paper introduces a modeling framework for approximate solutions, because it is not feasible to derive exact solutions for problems with large state spaces. Exact solutions to multiclass queuing networks are only possible for special classes (Casale 2006). These two examples show the importance of attributed customers, but also their impact on the complexity of exact solution methods.

One approach towards the analysis of queuing networks is the decomposition into single queuing systems and combination of their solutions. In Heindl (2001) one such approach is described for a special class of queuing systems. A decomposition approach for multiclass queuing networks is described in Whitt (1994). The restriction of the decomposition approaches lies in the goal to describe the performance measure results using mathematical analysis. This makes special solutions necessary for the many classes of queuing systems. However, to obtain exact solutions decomposition is the only feasible way to reduce the models' state space to controllable size. Therefore, a generally applicable and exact solution method for single queuing systems is still of interest.

2.2. Proxel-based Queuing System Simulation

Discrete event-based simulation is one way to derive estimates for the scalar performance measures, if no analytical solution exists. However, the results are of stochastic nature and only applicable to the single specified queuing system investigated. The simulation itself can get very expensive for stiff queuing models.

In (Hasslinger and Kempken 2006) an approach is presented for the transient analysis of a queuing system. Transition equations for discrete time points are derived and only the relevant state transitions at arrival and service instances are considered. The approach seems promising, and according to the authors can be extended to a number of queuing problems. However, the example in the paper is small, considering the two discretization points for the arrival and service distributions. The transition equations and the system state space can get complex when the problem gets larger, since all possible combinations of the arrival and service of a customer have to be considered.

The recently proposed Proxel-based simulation of queuing systems has several advantages. It can yield deterministic results for the performance measures of any queuing system. The accuracy of the results can be controlled. In addition to the steady state performance measures, the Proxel method also yields a transient solution, containing the probability of every possible system state at every investigated simulation time step. This cannot be easily obtained using common simulation techniques.

Proxel-based simulation is a state space-based simulation method that scans the possible system development paths in discrete time steps. (Horton 2002, Lazarova-Molnar 2005) In contrast to DES it follows all possible development paths, tracking their respective probabilities and building up a discrete-time Markov chain. This method ensures the discovery of rare development paths. The disadvantage of Proxels is, that through expanding the discrete system states by supplementary age variables, the resulting Markov chain can be of immense size. This so-called state-space explosion limits the applicability of Proxels to models with few discrete states, which is usually the case for queuing models.

The current implementation of the Proxel-based queuing system simulator is restricted to customers without attributes. This limits the possible queuing strategies to FIFO and other simple queuing strategies. However, many real systems use more sophisticated strategies for ordering their customers. To enable the Proxel-based simulation of more realistic queuing models, attributed customers are essential.

The basic idea of the extension to attributed customers and one example attribute implementation have been described in Wenjing Xu (2008).

3. ADDING ATTRIBUTED CUSTOMERS

This section describes the steps which are necessary to include attributed customers in the Proxel-based simulation of queuing systems.

We selected three static attributes without preemption policy for implementation. Adding variable attributes or preemption would further enlarge the state space. We also decided to limit the approach to ordering strategies with only one attribute, again to dampen state space explosion.

3.1. Attribute Choice

Common queuing strategies often sort the arriving customers by intrinsic priorities or some given time restrictions. Therefore, the following three attributes were chosen for the implementation and test of the method: *priority*, *processing time* and *deadline*.

A customer's priority is a measure of urgency or importance. It is given by an integer, where a smaller number implies a higher priority.

A customer's processing time holds the information how long the servicing of the customer will take. The processing time is usually given by a real value, but it can be converted into an integer by discretization using the Proxel simulation step Δt. This reduces memory cost.

The attribute deadline is a measure of urgency, which is again given by a real value. The deadline can also be expressed as the difference between the current simulation time and the customer's deadline. Doing this limits the attribute's value set and thereby reduces the system state space. The 'time to deadline' can also be converted to an integer by discretizing this distance using the discrete simulation time step Δt.

3.2. Coding and Storage Strategies

All three attributes can be expressed as integer values. The processing time and time to deadline can have possibly large value sets depending on the discretization time step Δt and the maximum support of the distribution.

By adding attributes to the customers in the queuing system, the discrete state space of the model will be increased significantly. This has two effects. First of all, the memory requirement of the algorithm will grow, because the number of Proxels is increased with a greater state space and because the size of a Proxel grows when one replaces a scalar queue length by an integer array holding the queued customers attributes. Secondly, the computational bottleneck of the Proxel-based method is the retrieval of already created Proxels. A newly created Proxel has to be compared to a possibly large number of existing Proxels in the storage container, requiring a comparison of all elements of these Proxels including discrete state space and the age vector. Therefore, a clever storage of the Proxels and the connected discrete system states is necessary to counter these effects.

The attribute of a customer in service can be stored in an array the size of the number of servers. The attributes of the customers in the queue are also stored in an integer array with a maximum length given by the program, that will most likely not be reached. After testing several options, this has shown to be the most efficient storage strategy for saving memory and retrieval time. The queue array contains the attributes of the customers in the order that they have in the queue.

The customer priority can be coded more efficiently than by the enumeration of all customers priorities in the queue. By listing the number of customers of each priority class in the array instead, the size of the array can be limited to the number of priority classes, which is most likely less than the maximum queue length. The example queue ‘111222233555’ could be expressed by ‘34203’. This saves memory and also shortens the time needed to compare two Proxels in the retrieval bottleneck.

Processing time and time to deadline can be enumerated for all queued customers, since there will not be as many customers with the same values.

3.3. Adapting Algorithm

The Proxel-based queuing system simulation algorithm also needs to be adapted slightly.

The attributes of the customers in the queuing system are fixed upon their arrival. Therefore, the formerly single arrival Proxel will now be split into as many Proxels as possible values of the chosen attribute (e.g. number of priority classes, or each possible discretized service time). In each of these Proxels, the queue has to be reordered, meaning the attribute of the arriving customer has to inserted into the queue array at the appropriate position.

The time to deadline and priority attribute have no influence on the processing of the customer. Fixing the processing time of a customer upon arrival turns the actual service into a deterministic process. This reduces the number of possible service time development paths to one, reducing the overall system state space. This decrease might compensate some of the increase caused by splitting each arrival Proxel.

3.4. Adapting Interface

The interface of the Proxel-based queuing simulation tool has to be changed to enable the selection and specification of customer attributes. In the adapted interface the user can choose to include customer attributes in the queuing system. A separate dialog enables a selection of the attributes. The number of priority levels and a probability distribution for the time to deadline can also be specified.

The output part of the interface was not adapted to include statistics of attributed customers. It only holds the overall system performance measures. Due to complexity, the measures calculated with regard to the attributes (see next subsection) are output into files.

3.5. Calculating Performance Measures

The inclusion of customer attributes into the queuing system specification enables the calculation of more sophisticated performance measures. Examples of newly possible measures are the following:

- Priority - average and extreme waiting times per priority class
- Deadline - the number of late customers and cumulative lateness
- Processing time - minimum and maximum waiting time for each possible processing time

All of these performance measures can be calculated on the fly or using the steady state result of the Proxel algorithm. Their calculation does not influence the algorithms performance significantly.

4. EXPERIMENTAL RESULTS

This section describes some experiments conducted to show that the inclusion of attributes into Proxel-based queuing system simulation is possible, and to test the effect of adding attributes on the algorithms performance. Furthermore we want to show some interesting results that can be obtained only by including attributes in the Proxel-based simulation.

The first experiment simulates a queuing system using the SJF (shortest job first) scheduling strategy. We obtain a distribution of the waiting time over the possible processing time values. This result is not easily computable using existing queuing analysis and simulation methods. The investigated queuing system has a Markovian arrival process with a rate of one customer per minute and a normally distributed service process with the parameters $\mu=0.5$ and $\sigma=0.1$. Figure 1 shows the average waiting time for the different

discretized values of the service time distribution. The waiting time of customers with short processing times is small, as expected and the waiting time of customers with larger processing times is longer. The steeper increase around the expected value of the service time is due to the higher number of customers created. These customers with equal discrete processing time delay each other when they are ordered according to FIFO.

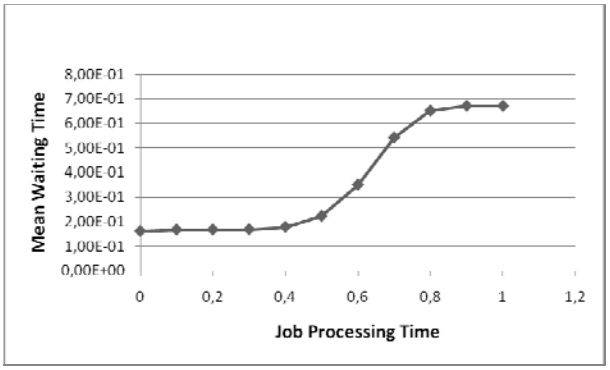

Figure 1: Waiting Time Distribution over Processing Time

The second experiment investigates the effect of adding different processing time distributions on the performance of the Proxel-based simulation method. The arrival process is set to be Markovian with a rate of one customer per minute. The service process is normally distributed with a mean of 0.5 and the standard deviation varied from 0.01 to 0.08. This results in 2 to 12 discretization time steps of the distributions as possible attribute values. Proxel simulations are performed using the same processing time distribution with and without attributes. Figure 2 shows the development of the runtime for the different service time distributions. The runtime of the algorithm without attributed tokens (grey line) is hardly affected by the change in service time distribution. The runtime of the simulation incorporating attributed customers (black line) rises from two seconds to several hundred. The increasing number of discretization time steps of the service time distribution increases the attributes value space and thereby increases the system state space.

This experiment shows, that adding the attribute service time to the Proxel simulation increases the complexity drastically. The combinatorial effect of enumerating every possible combination of customers waiting in the queue blows up the state space by several orders of magnitude. The effect is more severe, when more different attribute values are possible.

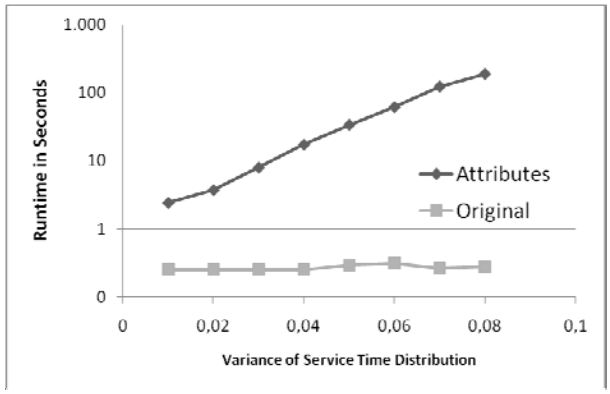

Figure 2: Runtime over Service Time Distribution with and without Attribute

The third experiment investigates the effect on computation cost when adding the attribute priority and increasing the number of priority classes. The tested queuing system has again a Markovian arrival process with an arrival rate of one customer per minute and one server. The service time distribution was chosen to have a rate of two customers per minute. Two different distributions were tested for the service time: Exp(λ=2) and N(μ=0.5, σ=0.1). We want to investigate the effect of the attributes on different queuing systems and compare the storage strategies enumeration of priority classes and count of elements per class. The resulting number of concurrent Proxel elements (as a measure of memory requirement) for both experiments were the same. The overall storage space needed could be reduced to about one half by using the counting strategy, since the storage space needed by one Proxel was reduced.

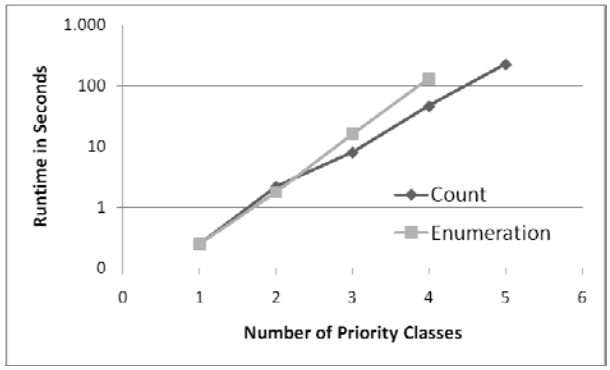

Figure 3: Runtime for Exponential Service Time Distribution and Different Numbers of Priority Classes

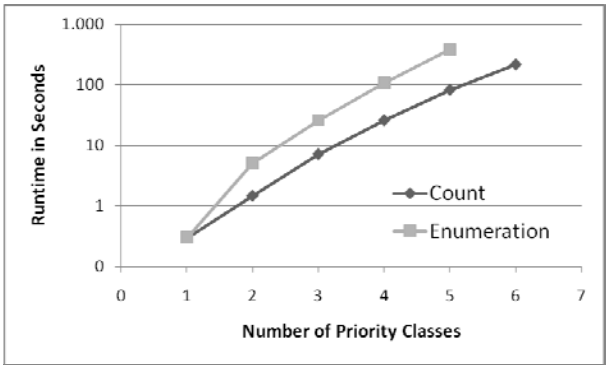

Figure 4: Runtime for Normal Service Time Distribution and Different Numbers of Priority Classes

The development of the algorithm runtime for the M/M/1 queuing system with a growing number of priority classes can be seen in Figure 3. The grey line shows the experiment using enumeration of all queue elements and the black line the development for only counting the elements per priority class. The drastic effect of this small change in the storage scheme is surprising. This shows that the handling of the Proxels and especially the comparison of two individual probability elements is one major bottleneck of the algorithm. When the attributes of all queue elements are stored, these also need to be compared individually when two Proxels are compared. This overhead can be reduced when only the number of elements per priority class need to be compared. The same effect is also visible in Figure 4, where the runtime for the normal service time distribution is depicted.

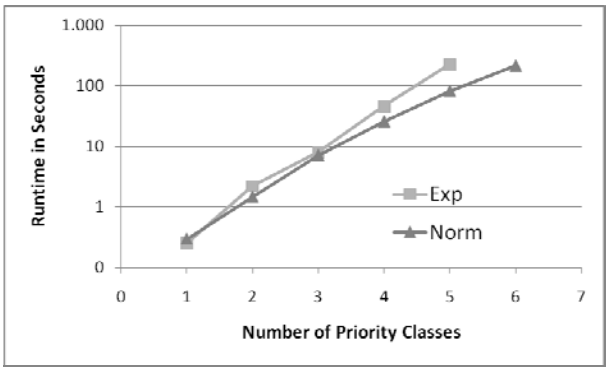

Figure 5: Algorithm Runtime for for Different Servicec Time Distributions with Priority Attribute

In Figure 5 one can see that using a normally distributed service times, the runtime is faster than when using exponential service times. This behavior is quite opposite to that of the original Proxel algorithm, where exponential distributions do not expand the state space due to their constant rate function. The higher runtime costs for the exponential service time distribution are caused by a larger state space, due to longer possible queue length when having a service time distribution with a greater variance.

This experiment shows, that the increase in state space by adding age variables for a non-Markovian distribution is less severe than the combinatorial explosion caused by the greater queue length.

Therefore, the effect of adding attributes to a queuing system is not so much dependent on the original state space of the system, but rather on the possible maximum number of customers in the system, which is causes the number of possible combinations of customers attributes

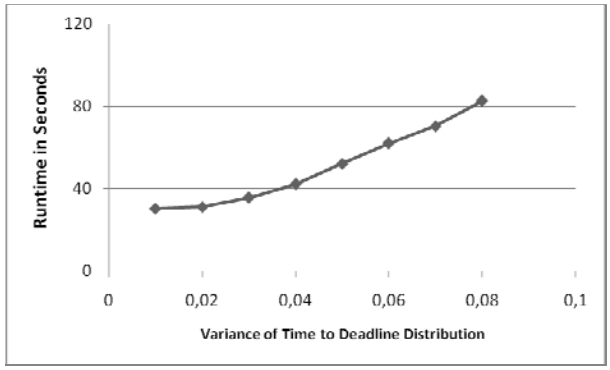

Figure 6: Runtime for Different Deadline Distributions

The last experiment shows the effect of different deadline distributions on the computation cost of the method. The queuing system used has normally distributed interarrival times (~N(0.7,0.2)) and normally distributed service times (~N(0.5,0.1)). Figure 6 shows the development of the algorithm runtime using different deadline distributions with a mean of 0.5 and with the number of possible time-to-deadline values varying from 2 to 12.

The initial simulation runtime of 30 seconds is already large considering a maximum possible queue length of 5, but the runtime increase for supposedly larger value set is not that severe. This happens because the value of the attribute time to deadline is not static. In the course of the simulation the value is updated by decreasing it in every time step, so that it reaches zero when the deadline approaches, afterwards it becomes negative. Therefore the actual value space of the attribute deadline ranges from the maximum distribution value to the longest delay and is larger than the initial number of discretized distribution values.

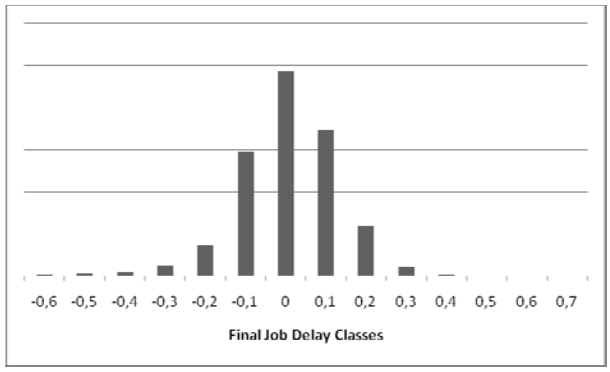

Figure 7: Throughput of Customers with Different Delays

In our experiment, the maximum delay was 1.7 minutes, which increases the maximum number of attribute values to almost 30 in the worst case. Figure 7 shows the throughput for the different final delays. The most frequent attribute values can be found around zero, as is expected when the service time and deadline distribution have the same mean. All other classes between -1.7 and 1.1 have much lower frequencies.

All experiments show a significant increase in memory and runtime cost due to adding attributes, as can be expected. However, clever storage and coding strategies help reduce these costs. The number of different attribute values that can be added feasibly differs between the attributes. The maximum number of priority classes is about 10, also depending on the maximum possible number of customers in the system. The discrete time steps of a distribution can be up to 20 or more, when the extreme values are less frequent than the average ones, such as when using a normal distribution. This happens because the simulation algorithm truncates Proxels below a certain probability threshold to limit the computation complexity.

Overall, the inclusion of attributes into Proxel-based queuing system simulation is feasible for attributes with a small value set or a small number of frequent attribute values.

5. CONCLUSION AND OUTLOOK

The paper described how Proxel-based queuing system simulation can be improved by adding attributes to the customers. This increases the number of possible applications and realism of the simulated queuing system classes. Limiting the approach to only one static attribute was necessary to keep the model's size in check. Furthermore, efficient storage strategies are essential for handling of the state space explosion.

Adding attributes leads to an increase in computation cost. This currently limits the applicability to queuing systems with general distribution functions with small variances or a low mean value. Adding one attribute with a small value set is possible.

The developed approach can be useful to queuing analysts, if no analytical solution is available for a queuing system. Especially rough estimates of the performance measures can be obtained very fast. Interesting results can be obtained such as the distribution of waiting time over the possible values of a customers processing time or the probability for infinite postponement.

Since we were able to implement the attribute processing time at a low cost, one future task is to implement round-robin as a queuing strategy. Implementing multiple attributes per customer is possible, but will only be feasible in conjunction with even more efficient storage strategies.

REFERENCES

Bolch, G., Greiner, S., de Meer, H., Trivedi, K.S., 2006. *Queueing Networks and Markov Chains*. John Wiley & Sons, Hoboken, New York, 2nd edition.

Casale, G., 2006. An efficient algorithm for the exact analysis of multiclass queueing networks with large population sizes. In *Performance Evaluation Review*, Vol. 34 (1), pp 169-180.

Cremonesi, P., Schweitzer, P. J., Serazzi, G., 2002. A Unifying Framework for the Approximate Solution of Closed Multiclass Queuing Networks. In *IEEE Transactions on Computers*, Vol. 51 (12), pp. 1423 - 1434.

Gross, D., Harris, C. M., 1998. *Fundamentals of Queueing Theory*. John Wiley & Sons, New York, 3rd edition.

Hasslinger, G., Kempken, S., 2006. Transient analysis of a single server system in a compact state space. In *Proceedings of 13th International Conference on Analytical and Stochastic Modelling Techniques and Applications*. European Council for Modelling and Simulation.

Heindl, A., 2001. Decomposition of General Tandem Queueing Networks with MMPP Input. In Performance Evaluation, vol. 44 (1-4), pp 5-23.Krull, C., Horton, G., 2007. Application of Proxels to Queuing Simulation. In *Proceedings of Simulation and Visualization 2007*, Magdeburg, Germany.

Horton, G., 2002. A new paradigm for the numerical simulation of stochastic petri nets with general firing times. In *Proceedings of the European Simulation Symposium 2002*. SCS European Publishing House.

Lazarova-Molnar, S. 2005. *The Proxel-Based Method: Formalisation, Analysis and Applications*. PhD thesis, Otto-von-Guericke-University Magdeburg. Germany.

Whitt, W., 1994. Towards better multi-class parametric-decomposition approximations for open queueing networks. In the *Annals of Operations Research*, Springer Netherlands, Vol. 48 (3), pp. 221-248.

Xu, W. 2008. *Application of Proxels to Queuing Simulation with Attributed Jobs*. Masters thesis, Computer Science Department, Otto-von-Guericke-University Magdeburg, Germany.

AUTHORS BIOGRAPHY

CLAUDIA KRULL studied Computer Science at the Otto-von-Guericke-University Magdeburg, obtaining her Diploma in 2003. She spent an exchange year at the University of Wisconsin, Stevens Point, where she graduated in 2002. In April 2008 she successfully defended her PhD thesis at the Otto-von-Guericke-University Magdeburg.

GRAHAM HORTON studied Computer Science at the University of Erlangen, obtaining his Masters degree ("Diplom") in 1989. He obtained his PhD in Computer Science in 1991 and his "Habilitation" in 1998 at the same university, in the field of simulation. Since 2001, he is Professor for Simulation and Modelling at the Computer Science department of the University of Magdeburg.