SIMULATION OF SEARCH AND DISTRIBUTION METHODS FOR JOBS IN A WEB OPERATING SYSTEM (*WOS*™)

Herwig Unger[*], Peter Kropf[†], Gilbert Babin[†], and Thomas Böhme[‡]

[*] Fachbereich Informatik
Universität Rostock
D-18051 Rostock, Germany
hunger@informatik.uni-rostock.de

[†] Département d'informatique
Université Laval
Québec, Canada G1K 7P4
{kropf,babin}@acm.org

[‡] Mathematisches Institut
TU Ilmenau
D-98684 Ilmenau, Germany
boehme@theoinf.tu-ilmenau.de

KEY WORDS

Distributed Systems, Communication Simulation, Search Methods, World Wide Web, WOS™(Web Operating System), Performance Estimation, Protocols.

ABSTRACT

Internet or WEB computing concepts, such as the Web Operating System (WOS™), require suitable methods for finding the most effective hardware and software environment for an application execution on the network. We present an efficient search strategy and show that the resulting communication load penalty is well acceptable while preventing the network from being flooded by administrative overhead messages. Using stochastic time estimation, we demonstrate the favorable response time behavior and performance of the search algorithm.

1 INTRODUCTION

With the rapid development of new forms and concepts of networked and mobile computing, it is increasingly clear that operating systems must evolve so that all machines in a given network can appear to be controlled by the same operating system. As a result, the world-wide interconnected networks, commonly called the Internet or the Web, could potentially be supported and managed by a huge virtual operating system (Reynolds 1996). The transparent use of heterogeneous networks of computers has been partially addressed in work on *metacomputing*, whose objectives are to transform a network into one single computer system. However, the Web is more than just a metacomputer because there is no complete catalogue of all available resources and above all, such a catalogue is infeasible, given the highly dynamic and distributed nature of the Web.

Being able to use the global network for parallel/distributed execution of a program is clearly promising. For this to be realistic, mechanisms are required to distribute the work, collect the results and coordinate the participating processes or agents. In particular, before any task can be scheduled to any suitable computer or sub-architecture, a (nonempty) set of machines must be found, where

- the necessary software is installed in the right version,
- enough free resources can be found to execute the job,
- the job can be finished in the time required and
- the machine is available for remote execution of the necessary program(s).

It is easily seen that in the huge network of the Internet a number of communication is necessary to complete all the above tasks while executing any service. In (Ben Lamine et al. 1997) and (Unger et al. 1998) a number of problems was described resulting from the special situation in the Web where no actual catalogues of resources are available.

2 CONCEPTS OF A WEB OPERATING SYSTEM (WOS)

The *Web Operating System* (WOS™) (Ben Lamine et al. 1997) is a virtual operating system that supports and manages distributed/parallel processing on the Internet. The WOS is a versioned system, in which different versions not capable of dealing with a particular request for service, then pass it on to another version, as currently done for packet routing. Generalized software configuration techniques, based on a demand driven technique called *eduction* (Plaice and Ben Lamine 1997) are being developed, that can be used to define versions of a WOS

to be built in an incremental manner. Software and hardware (description) repositories, or warehouses, will provide the necessary components for fulfilling a service requested. The kernel of a WOS is a general eductive engine responding to requests from users or other eductive engines and fulfill these requests using its warehouses.

The WOS then works in the following manner. A request may be placed by a user to run a particular program or to initiate some service. The programs or services might be located at different sites of the network. The eductive engine may then decide whether it is capable of dealing with the request or whether it will pass it over to some other eductive engine until finally one engine accepts responsibility for the request. Once all the resources (programs, services, hardware) become available, then the program is run and the requested service fulfilled.

There are numerous projects currently under way dealing with similar problems as the WOS. Computer network exploitation approaches, such as PVM, MPI, CORBA (Mowbray and Zahavi 1995), Legion (Grimshaw et al. 1994), require login privileges and a global catalog of resources, as opposed to the WOS which uses distributed warehouses. Other approaches, including : ParaWeb (Brecht et al. 1996), Atlas (Baldeschwieler et al. 1996), and Charlotte (Baratloo et al. 1996), create a *metacomputing* infrastructure for large networks to enable parallel execution of applications. The approach proposed by WebFlow (Bhatia et al. 1996) exploits the Internet for parallel execution of programs.

The approach closest to the WOS is SuperWeb (Alexandrov et al. 1996). Here, clients submit requests to service brokers which will allocate servers to perform the request. The model proposed includes security aspects as well as a simple economic model of resource management. Contrarily to the WOS, servers and clients must register with the broker. This implies that the broker manages a complete catalog of resources. This approach may be interesting for a small network, but would prove impractical over a large global network.

3 PRINCIPLES OF SERVICE EXECUTION IN THE WOS

As mentioned above, no global catalogues of resources are available in the Web. Thus all methods used by the WOS must be based on the existence of decentralized information warehouses as introduced in (Ben Lamine et al. 1997). Furthermore, as described in (Unger et al. 1998), we suppose that there is a non-fixed number of such warehouses containing references to machines which potentially can execute a service in a specified area of the Web. Therefore, searching for any information implies contacting nearby warehouses. If the necessary information cannot be found there, other warehouses further away must be contacted. At least one warehouse must give a positive answer to terminate the search and initiate the service requested (see Fig. 1, 2 and 3).

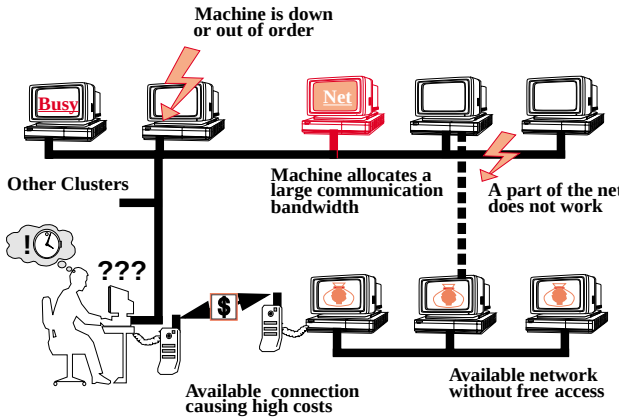


Figure 1: Example architecture showing some of the new problems in a WOS

4 MODELING AND ESTIMATION OF THE PRINCIPAL SEARCH ALGORITHMS

4.1 Basics

Two main strategies can be considered for the communication with the warehouses during the search process.

1. **The Broadcast Request Strategy**
   In this case the requesting machine broadcasts the request to a set of a warehouses. These a warehouses can almost work in parallel such that answers will quickly be available on the requesting machine. The problem is that in this case $2a$ messages will be generated; the broadcast implementation must be realized as multiple unicasts, hence delaying data transmission. This strategy implies high network load.

2. **The Serial Request Strategy**
   If a serial request is used, only one message will

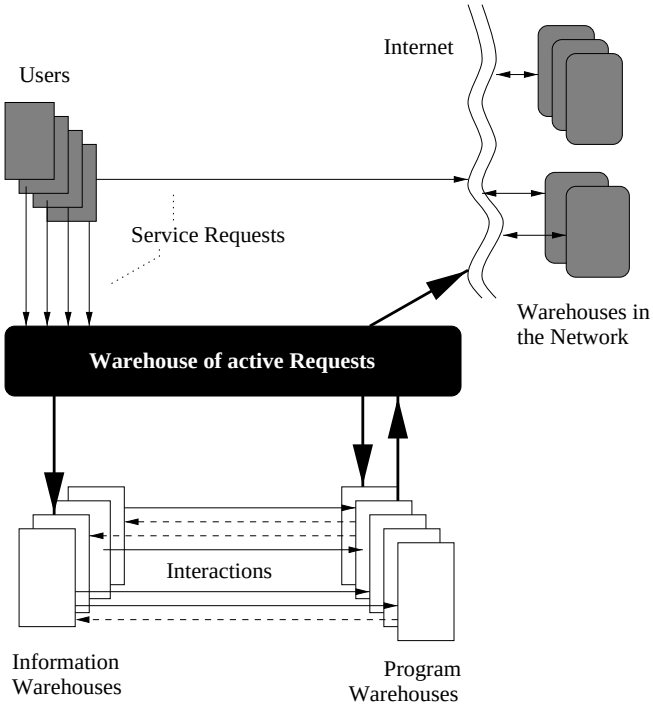


Figure 2: Introducing the idea of Warehouses

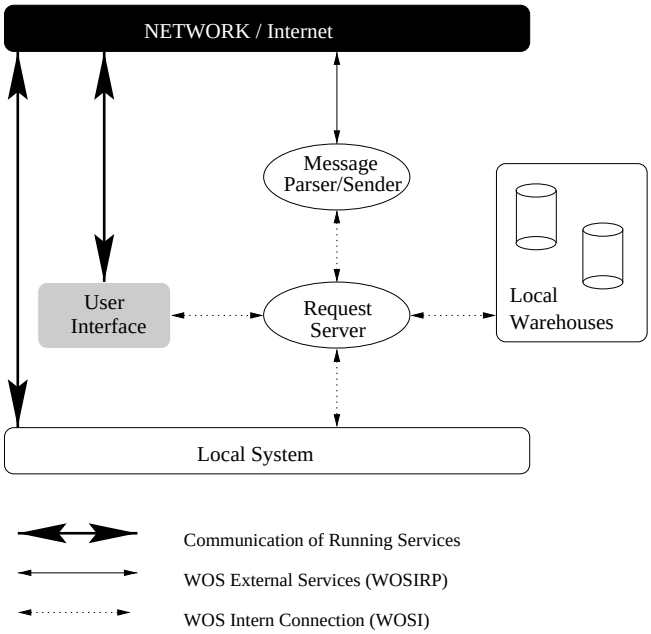


Figure 3: Necessary Protocols in a WOS

be sent by the requesting machine to one warehouse. If the necessary service is available in the area of this warehouse, a positive message is directly sent back to the requesting machine. Otherwise the request will automatically be directed to the next possible warehouse taken from a list of warehouses included in the request (until there is no warehouse left in the list). If there are b warehouses in the list, $b+1$ messages will be generated and sent through the communication network. The number of messages needed is much less than in the first case. On the other hand, the response time of such a request chain is much higher than in the first case and any communication problems or long transfer times directly influence the response time until a suitable machine is found.

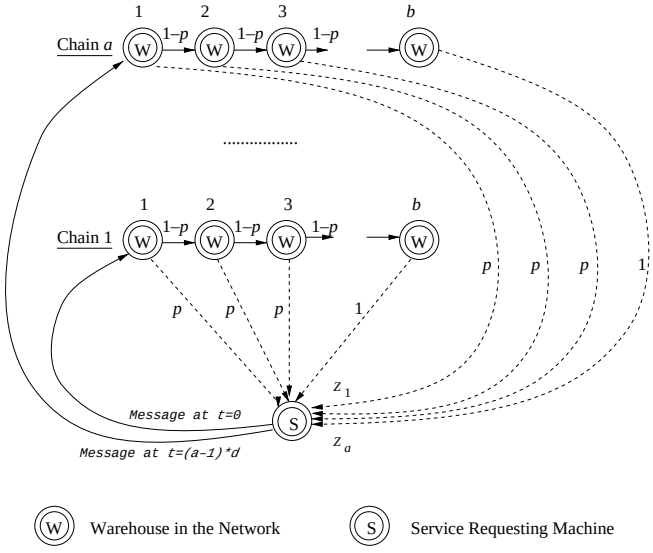
It is clear that the integration of both strategies can improve the relations by combining the main advantages and avoiding the described disadvantages. We can generate a serial chains, each chain containing b warehouses. For $b = 1$, we use the Broadcast Request Strategy. For $a = 1$, we use the Serial Request Strategy. For a first theoretical estimation, other parameters need to be introduced (see Fig. 4) :

- t_t, the average transmission time of a message from one machine to another one;
- t_v, the average processing time of each warehouse, to search whether a service can be executed in its area;
- d, the time difference between the transmission of two broadcast messages within the first common shared subnetwork;
- p_i, the probability that the requested service can be executed in the area of the considered i-th warehouse (we suppose that this is a constant for each warehouse and denote this probability by p);
- q, the probability that the service is found in the first search period (a message chains of b machines).

Furthermore, to calculate the expected response time of a system of a chains of b warehouses, Z_i shall denote the response time of the i-th chain. The response time can be determined in case of a successful search from any machine or for a negative answer after b machines. We are now ready to consider the behavior of our system.

4.2 Stochastic Time Estimation

From the above, it is clear that Z_i denotes a probabilistic value, which is identically distributed for all a chains. If $P(Z_i = x)$ is the probability that chain



Distribution of the Answer Time Z_i of Chain i after its start

Z_i	$t_v + 2t_t$ ...	$jt_v + (j+1)t_t$	...	$bt_v + (b+1)t_t$
Probability	p ...	$p\,(1-p)^{(j-1)}$	...	$(1-p)^{(b-1)}$

Figure 4: Message Transmission and their Probabilities in the used Model

i answers within x time units, the distribution function for Z_i is defined by

$$F_{Z_i}(x) = P(Z_i \leq x)$$

For chain i, F can be calculated by

$$F_{Z_i}(x) = \sum_{j=1}^{b-1}(p(1-p)^{(j-1)}\sigma(x-(jt_v+(j+1)t_t)))+R(x)$$

$$\text{with } \sigma(r) = \{1 : r \geq 0; 0 : r < 0\} \text{ and}$$

$$R(x) = (1-p)^{(b-1)}\sigma(x-(bt_v+(b+1)t_t))$$

This includes the case where a negative answer is provided after b warehouses have been visited.

All the chains work independently, but will be started by broadcast messages issued from the same requesting machine. Because of the transmission mechanism described above, the start of the work of the i-th chain will be delayed by $(i-1)\cdot d$. Therefore the distribution function of the response times must be modified as follows :

$$\tilde{F}_{Z_i}(x,i) = F_{Z_i}(x-(i-1)\cdot d)$$

The response time Z for all chains is the minimum value of all Z_i. The distribution function for all chains $F_Z(x)$ must be determined. Because $F_{Z_i}(x) = P(Z_i \leq x) = 1 - P(Z_i > x)$, $F_Z(x)$ is given by

$$F_Z(x) = 1 - \prod_{i=1}^{a} P(Z_i > x) = 1 - \prod_{i=1}^{a}(1 - \tilde{F}_{Z_i}(x,i))$$

Finally it is known from probability theory that the expected overall response time Z of the system can be calculated by

$$E(Z) = \sum_{x=1}^{\infty} x(F_Z(x) - F_Z(x-1))$$

From the above, it is easy to see that the requested service will be found with probability q on at least one of $n_q = a \cdot b$ machines during the first attempt, if

$$\sum_{i=1}^{n_q-1} p(1-p)^{(i-1)} < q \leq \sum_{i=1}^{n_q} p(1-p)^{(i-1)}$$

Therefore n_q can be determined by

$$n_q = \frac{ln(1-q)}{ln(1-p)}$$

4.3 Results and Consequences

Figures 5, 6 and 7 show the results graphically. Figure 5 contains a linear correction, for the case where too many chains are used and the requesting machine is blocked for a longer time.

In light of the theoretical results for the expected response time (see Fig. 5 and 7) we should consider the following :

- A number of about 6-10 chains working in parallel results in good response times for all n_q's while avoiding a too large load of the networks.
- Z mostly depends on the probability p that a service is available in the warehouses; if n_q is big enough, it does not influence the expectation very much. Hence, warehouses with the highest success probability p_i should be placed at the beginning of chains.
- The necessary n_q drastically grows, as q exceeds the 80 percent border. This is the reason why q shall be selected around 80 percent in order to find a suitable n_q.
- Warehouses being searched should have a similar distance from the requesting machine and/or from each machine in a chain to the next one for achieving similar upper response time limits for all chains.

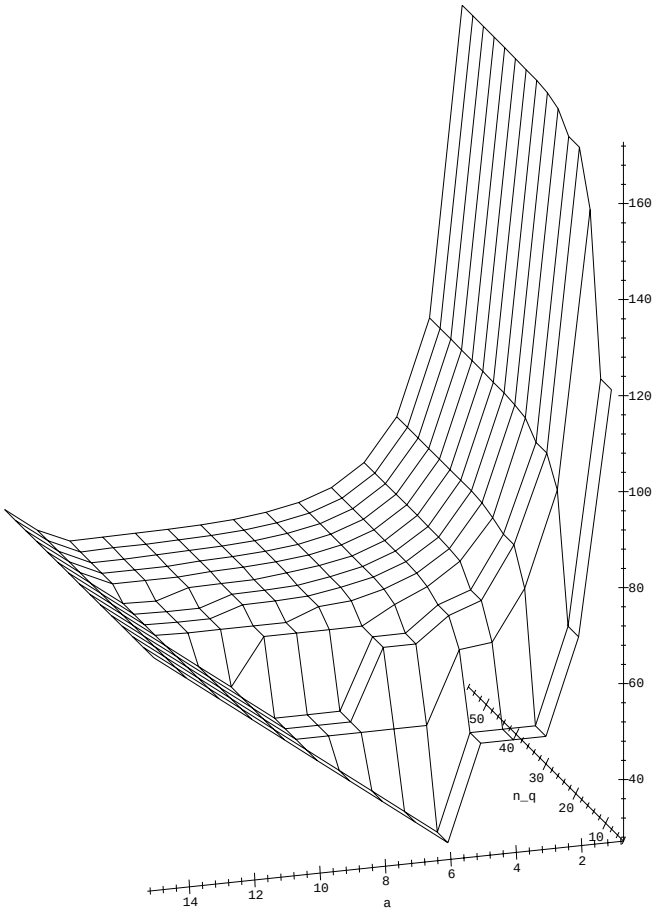


Figure 5: Response time Z as a function of n_q and a

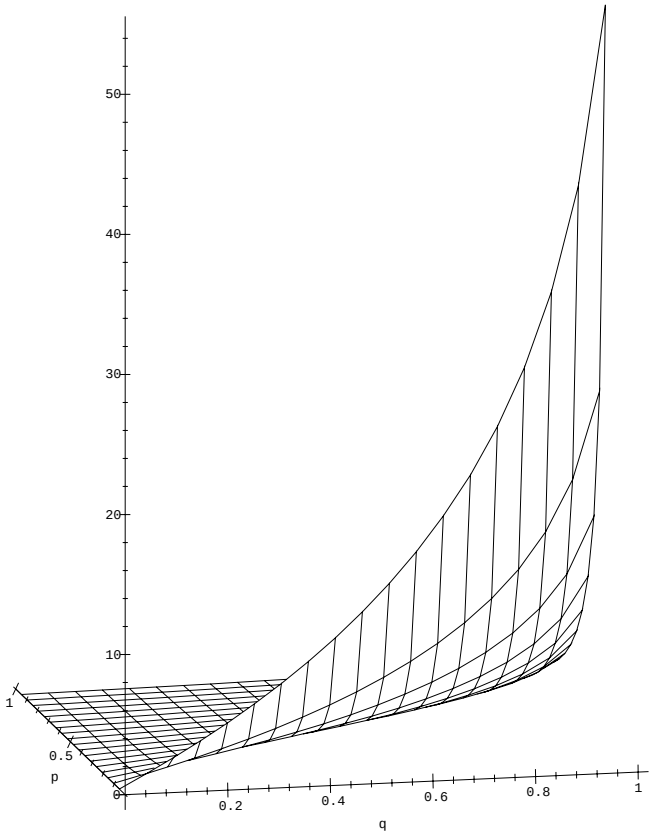


Figure 6: n_q as a function of q and p

- The communication time between any two machines and the response time of each chain must be restricted to tolerate machine faults or a drastic increase of the communication time in a network structure. Therefore, two timeouts must be introduced: the first for limiting the waiting time for a result from a chain, the second to prevent search messages from running endlessly trough the system. Such search messages must be rendered invalid and of course be removed from the system.
- All these characteristics mentioned above recursively apply, if a a warehouse itself starts any further search actions or if a search must be repeated.

5 CONCLUSIONS AND FUTURE WORK

A major goal of the WOS system concept is to provide to a user the most effective hardware and software environment. In particular, this means providing the fastest possible application execution, respecting any possible constraints such as price/performance demands, security issues or response time requirements. To achieve these aims, it is crucial that (1) the most suitable resources are found (if available at all) within the desired time, and (2) the communication network is prevented from being flooded by (valid or invalid) search messages. The proposed search strategy based on the combination of broadcast and serial requests has been shown to meet these requirements.

Future work includes exhaustive practical testing of the theoretical results presented in this paper. First results obtained using a set of machines spread over the Internet indicate that the search strategy behaves according to the predictions.

References

Alexandrov, A.; M. Ibel; K. Schauser; and C. Scheimann. 1996. “SuperWeb: Research issues in Java-based global computing.” In *Workshop on Java for computational science and engineering workshop* (Syracuse University, Dec.).

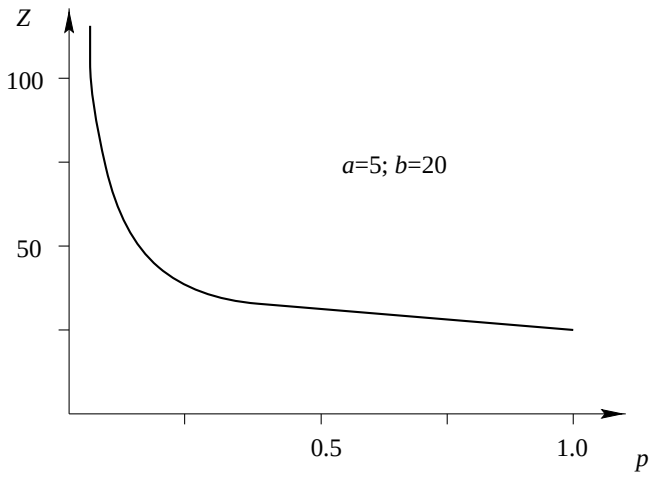


Figure 7: Response time Z as a function of p

Baldeschwieler, J.; R. Blumofe; and E. Brewer. 1996. "Atlas: An infrastructure for global computing." In *Seventh ACM SIGOPS european workshop on system support for worldwide applications*.

Baratloo, A.; M. Karaul; Z. Kedem; and P. Wykoff. 1996. "Charlotte: Metacomputing on the Web." In *9th conference on parallel and distributed systems*.

Ben Lamine, S.; J. Plaice; and P. Kropf. 1997. "Problems of computing on the Web." In *High performance computing 1997* (Atlanta, Ga.), Tentner, A., ed. SCS, 296–301.

Bhatia, D.; V. Burzevski; M. Camuseva; G. Fox; W. Furmanski; and G. Premchandran. 1996. "Webflow." In *Workshop on Java for computational science and engineering workshop* (Syracuse University, Dec.).

Brecht, T.; H. Sandhu; M. Shan; and J. Talbot. 1996. "Towards world-wide supercomputing." In *Seventh ACM SIGOPS european workshop on system support for worldwide applications*.

Grimshaw, A.; W. Wulf; J. French; A. Weaver; and P. Reynolds. 1994. "A synopsis of the Legion project." Technical Report CS-94-20, University of Virginia (Jun.).

Mowbray, T. and R. Zahavi. 1995. *The essential CORBA: Systems integration using distributed objects*. John Wiley & Sons, New York, NY, USA.

Plaice, J. and S. Ben Lamine. 1997. "Eduction: a general model for computing." In *Intensional programming II*. World Scientific, Singapore.

Reynolds, F. 1996. "Evolving an operating system for the Web." *IEEE Computer 29*, no. 9: 90–92.

Unger, H., P. Kropf and J. Plaice. 1998. "Load and resource sharing in a Web based computing system." In *IASTED International Conference on Networks and Communication Systems* (Pittsburgh, PA, May).

BIOGRAPHIES

Herwig Unger was born in 1966 in Halle. He got his M.Sc. from the Technical University of Ilmenau in 1991. In 1994 he received the Ph.D. from the TU Ilmenau; his thesis work was on transformation and implementation methods of Petri-Nets used as models of distributed programs. He has a permanent position at the Computer Science Department of University of Rostock since 1996, where he had his first appointment as an Assistant Professor. His reserach interests include resource scheduling, operating systems and formal methods for distributed computation and simulation.

Peter Kropf received the M.Sc. and Ph.D. degrees from University of Bern (Switzerland). Subsequently, he was a research scientist at the University of Bern for several years. Since 1994, he has been an Assistant Professor at Université Laval. He has been carrying out research in parallel computing for over ten years, particularly in the field of mapping and load balancing. Current projects especially include Internet computing and the WEB, and the efficient use of parallel computing in real world applications (radiotherapy, biomechanics, radio-reconnaissance, distributed command & control etc.). His research interests include Internet computing, parallel/distributed computing tools, networking and simulation.

Gilbert Babin received the B.Sc and M.Sc. from Université de Montréal (Canada), and the Ph.D. from Rensselaer Polytechnic Institute. Since graduating in 1993, he has been a faculty member at Université Laval (Canada). He has been working on the integration of heterogeneous, distributed databases using reactive agents and a central knowledge repository. In addition, his current interests include the formalization of the requirement engineering process, the Web Operating System (WOS^{TM}), and activity report generation from activity databases.

Thomas Böhme received the doctors degree (Dr. rer. nat.) in mathematics from the Technical University of Ilmenau in 1988. Since then he holds an appointment as an assitant professor at the Technical University of Ilmenau (Germany). His current research interests include graph theory, Petri-Nets and theoretical aspects of distributed operating systems.