

# How to Efficiently Deploy Mobile Agents for an Integrated Management

Steffen Lipperts

Aachen University of Technology, Department of Computer Science (i4)  
D-52056 Aachen, Germany  
steffen@lipperts.de

**Abstract.** Along with today's networks, which are gradually converging and setting the field for universal service markets offering value added services to users on a global scale, their requirements are changing fundamentally. It is becoming increasingly important to provide an enabling technology which allows to meet these new requirements. Mobile agents are being discussed as comprising the potential to be such a technology. In order to use mobile agents for an integrated management, however, two problems need to be solved, namely the problem of non-determinism and the problem of competing goals. This paper closely examines performance aspects of mobile agents and evaluates a solution for the two problems which is based on utility theory.

## 1 Introduction

The years have passed where different kinds of networks, especially in the internet domain, the media, and the telecommunication domains, merely coexisted. Having reached a global scale, networks today are converging in order to provide a uniform platform for all kinds of service provisioning. The management of the individual networks has always been a key issue due to their growing complexity and scale. Now it becomes necessary to provide suitable means for ensuring the availability and reliability of upcoming and converging service markets. The diminishing suitability of current approaches to distributed system and network management with regard to scale and flexibility is becoming more and more apparent and is widely discussed in the literature [1,2]. Accordingly, in October 1999, the Internet Engineering Task Force (IETF) has issued an internet draft (as an update to RFC 2257 [3]), in which it is stated that "This very real need to dynamically extend the management objects..." exists and that "It can become quite cumbersome to configure subagents ... on a particular managed node.". Several approaches have been made or are currently being examined to shape management according to the new requirements.

Some of these approaches aim at extending SNMP to overcome the problems of centralisation and inflexibility, such as the introduction of hierarchies in SNMPv2, Remote Monitoring (RMON), and AgentX. More powerful approaches appeared with the introduction of mobile code, first management by delegation [4], then various

approaches which deployed mobile agent technology. Several projects are active in this field, either evaluating the suitability of mobile agent for management in general [1,2], or focusing on different aspects, e.g. intelligence [5] and organisation [6] of mobile agents. The main reason for mobile agent deployment is their ability to migrate, which promises to offer added value for distributed systems and service markets through asynchronous, autonomous actions, usually entitled as “disconnected operations”, and through the “benefit of locality”, i.e. the advantage of a much more efficient local communication, reduced execution times, and reduced network traffic. This, however, is not necessarily always the case, as an overhead is introduced by the migration process. Therefore, by studying the applicability of mobile agents depending on the prerequisites set by the distributed system itself and by the nature of the agents’ tasks, this paper addresses a key issue for enabling the deployment of mobile agents in future distributed systems.

The paper is structured as follows. In the next section, two key problems of mobile agent deployment, which so far have been neglected, are pointed out and discussed in detail, together with performance studies which supply evidence for both the existence and the gravity of these problems. The third section illuminates how these problems affect the way mobile agents are to be used – not only for an integrated management – and evaluates a solution for these problems.

## 2 Performance of Migration

The key feature of mobile agents is their ability to migrate. While this promises several benefits as shown in the introduction, the migration process with its serialisation, transfer, and de-serialisation also introduces an overhead which makes these improvements situation dependent. In order to assess this in detail, measurements have been made to compare the migration process with remote communications. The results are shown in table 1.

The measurements have been made in a local area network. A single agent migration was to be compared to remote CORBA communications, as these are the standard for distributed systems. Two cases were to be distinguished, a migration including a classloading process and another one excluding it. The former is required if an agent arrives at a host it has never visited before and which does not hold the agent’s classes prior to its arrival. The latter is given if the classes are already present upon arrival.

**Table 1.** RPC vs. Migration

1a)

	Time in ms	one migration equals
migration with classloading	1430.1	-
CORBA RPC	0.9	1557
CORBA RPC (100 byte)	1.2	1210
CORBA RPC (1 Kbyte)	3.8	372
CORBA RPC (10 Kbyte)	23.1	62

1b)

	Time in ms	one migration equals
migration without classloading	19.4	-
CORBA RPC	0.9	21
CORBA RPC (100 byte)	1.2	16
CORBA RPC (1 Kbyte)	3.8	5
CORBA RPC (10 Kbyte)	23.1	0.6

As a first result, it can be stated that a migration including classloading is an expensive process, as in the meantime, an awful lot of remote communication can take place, even if the size of the parameters is reasonably large. Yet, one must not forget that in slower (class 2) media, this relation is less extreme, as can be deduced from the results shown in figure 1. Here, it can be seen that a far bigger percentage of a remote communication is affected by the underlying network (marked grey) than with a migration. Consequently, remote communication will perform worse on slow media. Things look even better for the migration process, if no classloading is required (see table 1b). Here, agent migration suddenly becomes very efficient, as in the meantime, only few remote communication can occur and following the migration, the agent will be able to use local communication, which only takes a fraction of a millisecond.

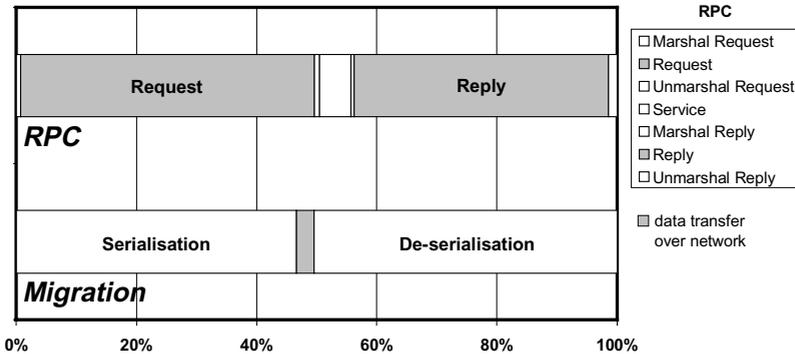


Fig. 1. Percentage of Data Transfer in Entire Process

Taking these results into account, the assumption can be verified that it is situation dependent whether or not an agent migration can provide a performance improvement. Given the results presented above, it can be decided to replace a certain number of remote communications by a migration, taking into account parameters such as

- How many remote communications will be replaced?
- What is the size of their parameters?
- What is the size of the agent?
- Which kind of network is given?

That is, a break-even point can be determined given this information where the migration overhead is compensated by the following local communications. Yet, rather than having reached a solution, here is where the problems start. In most cases, the required

information listed above cannot be obtained. Take for example operation that are to complex to calculate their communication costs in reasonable time. Even worse, take non-deterministic operations such as sequential searches or negotiations. In any of these cases, how are the parameters to be determined?

Even under the condition that the required information is given, a decision on the applicability of a migration would be far from perfect if merely the aspect of data transfer and execution time was taken into account. As described in the introduction, mobile agent deployment can offer other benefits which might be just as valuable as a performance improvement, such as improved robustness and availability, reduced error rates, or even required battery power of mobile systems. Picture a case, where not enough communication occurs in order to justify a migration. The link which is used for these communications, however, is not reliable, e.g. a GSM connection in a moving vehicle. Depending on the importance of the process to be executed, improved reliability might be the main target. Moving a mobile agent to the wired network thus might be reasonable after all.

### 3 A Solution Based on Utility Theory

So far, the necessity for new approaches to management has been pointed out and both the potential and the major problems of mobile agent technology have been discussed, non-determinism and competing goals. In [7], we have presented the outline of a solution for these problems which is based on utility functions. These assign numbers to states in order to express their desirability, thus allowing a ranking of these states. Let  $U(S)$  denote such a utility of a state  $S$ . Now, let  $C$  be the current state an agent is in and  $A$  a non-deterministic action with  $i$  possible results  $Result_i(A)$ , i.e. a transition from  $C$  to any state in  $Result_i(A)$ . If for  $C$  and for all  $A$ , the probabilities  $P$  of the results  $Result_i(A)$  are given, i.e.  $P(Result_i(A)|A, C)$  given for all  $A$  and for all  $i$ , then for all  $A$ , the expected utility EU can be calculated:

$$EU(A|C) = \sum_i P(Result_i(A)|A, C) \cdot U(Result_i(A)) \quad (1)$$

In utility theory, the principle of maximum expected utility (MEU) has been established which denotes that a rational agent should choose an action which maximises the agent's expected utility, i.e. the agent is to maximise the average utility [8]. The algorithm to be followed by a mobile agent thus is as follows (for simplification in pseudo-code):

```
void next_step(){
    determine current status C;
    for all actions Ai {
        for each possible outcome Resultj(Ai) {
            obtain current probability P(Resultj(Ai)|Ai, C);
        }
    }
    maxEU = 0;
```

```

for all possible actions  $A_i$  in status  $C$  {
    eu = calculate  $EU(A_i|C)$ ;
    if(maxEU < eu)
        maxEU = eu;
    }
execute action  $A_i$  with maxEU;
}
    
```

Whilst determining the current status is fairly straightforward and can be done by the agent internally, the probability densities of the possible goal states of all actions  $A_i$  depend on current system states and thus need to be retrieved from the system, i.e. from suitable system components. A trader or an equivalent component in a distributed platform which is responsible for service procurement can generate this information. Then, given the current state  $C$ , the possible actions  $A_i$ , and the probabilities of their outcomes, the expected utilities can be calculated and maximised.

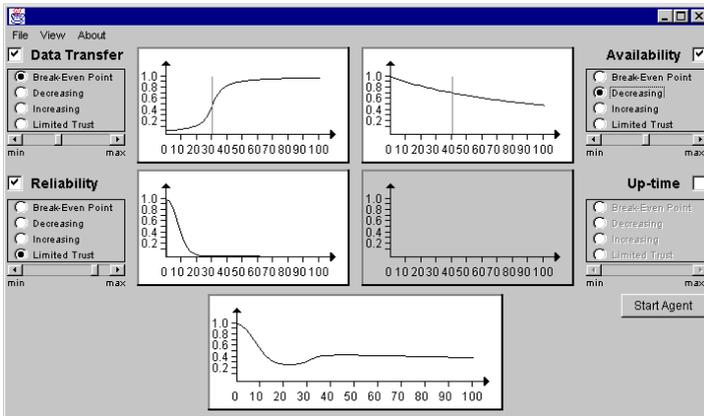


Fig. 2. Tool based Determination of the Utility Function

That is, once the utility functions are given. In [7], we have shown how multiattributive utility functions can be composed by individual utility functions, if a multiplicative form can be determined. Figure 2 shows a tool based definition of such a multiattributive utility function, where the functions for the individual attributes and their respective weights can be specified by the user, leading to an automated composition of the multiattributive utility function.

In order to assess the suitability of the solution presented in this paper, smart mobile agents deploying the utility based behaviour have been implemented and evaluated. A direct comparison has been made with stationary agents and other mobile agents running in the identical environment. The focus has been on the execution times, as these allow a quantitative evaluation of the suitability of the new approach.

Despite the overhead through data transfer, additional calculations, and additional size due to the utility functions carried on their itinerary, the smart agents were able to

reduce the execution times as shown in figure 3, always choosing the most effective paradigm, either migration (phase 2) or remote communication (phase 3).

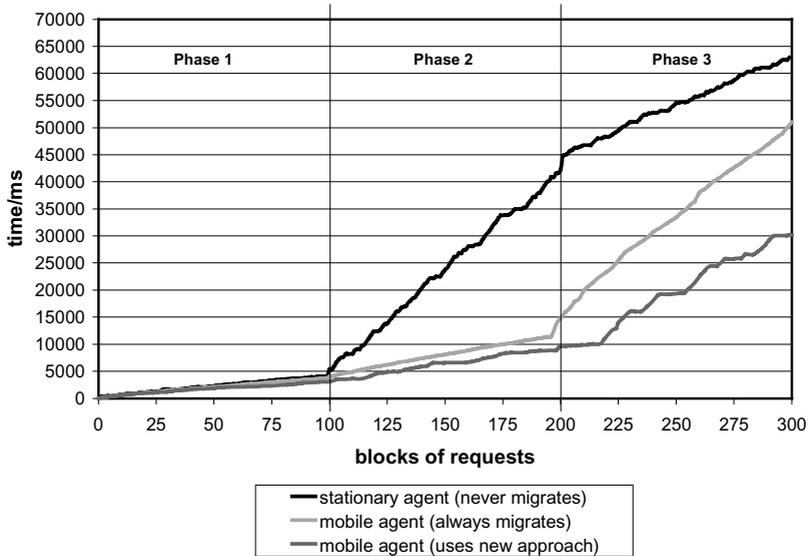


Fig. 3. Comparison of the Different Paradigms

## References

1. Bieszczad, A.; Pagurek, B.; White, T.: Mobile Agents for Network Management. In: IEEE Communications Surveys, September 1998
2. Puliafito, A.; Tomarchio, O.: Advanced Network Management Functionalities through the Use of Mobile Software Agents. In: Proceedings of the Third International Workshop on Intelligent Agents for Telecommunication Applications, Stockholm, Sweden, August 1999
3. IETF: Agent Extensibility (AgentX) Protocol Version 1. RFC 2257, January 1998
4. Goldszmidt, G.; Yemini, Y.: Delegated Agents for Network Management. In: IEEE Communications: Management of Heterogeneous Networks, Vol.36, No.3, March 1998
5. Sahai, A.; Morin, C.; Billiard, S.: Intelligent Agents for a Mobile Network Manager (MNM). In: IFIP International Conference on Intelligent Networks and Intelligence in Networks, Chapman & Hall, Paris, France, September 1997
6. White, T.; Bieszczad, A.; Pagurek, B.: Distributed Fault Location in Networks Using Mobile Agents. In: Proceedings of the 3rd International Workshop on Agents in Telecommunications Applications, AgentWorld'98, Paris, France, July 1998
7. Lipperts, S.: On the Efficient Deployment of Mobility in Distributed System Management. In: 3rd International Workshop "Mobility in Databases & Distributed Systems", Greenwich, UK, September 2000
8. Russel, S.; Norvig, P.: Artificial Intelligence – A Modern Approach. Prentice Hall, 1995

This work has been funded by the German Research Council (DFG) under SFB 476 IMPROVE and GRK 185/1-00.