Int J Adv Manuf Technol (2007) 32: 822–833
DOI 10.1007/s00170-005-0378-1

ORIGINAL ARTICLE

Chi-Yu Huang · Kai Cheng · Alan Holt

An integrated manufacturing network management framework by using mobile agent

Received: 14 June 2005 / Accepted: 2 December 2005 / Published online: 22 March 2006

Abstract The growth in the development of distributed manufacturing systems necessitates increases in efficient technologies in order to facilitate management of the networked resources. Systems such as manufacturing automation protocol (MAP) already exist which were specifically designed for managing networked manufacturing devices. However, MAP has been criticised for being over-complex and expensive for small–medium sized enterprises. The simple network management protocol (SNMP) was designed for managing TCP/IP networks, and we argue that it can also be used for managing networked manufacturing devices. However, SNMP, like MAP, is based upon a client/server architecture, and both have scalability issues. We also investigate the use of mobile agent (MA) technology as a means of managing a domain of manufacturing devices. An integrated network management framework based on MA technology is developed to direct the research into the investigation of aligning network management paradigm to the strategic management decision.

Keywords Mobile agent · SNMP · Clustering · Mobile agents platform—MaP · Virtual physical domain agent platform

C.-Y. Huang (✉)
DL Consulting Ltd.,
Hamilton, New Zealand
e-mail: chi@dlconsulting.co.nz

K. Cheng
Faculty of Information and Engineering Systems,
Leeds Metropolitan University,
Calverley Street,
Leeds, UK
e-mail: K.Cheng@leedsmet.ac.uk

A. Holt
Department of Computer Science and Mathematics,
University of Waikato,
Hamilton, New Zealand
e-mail: aholt@cs.waikato.ac.nz

Nomenclature

CIM	Computer integrated manufacturing
CoD	Code on demand
DDR	Distributed database repository
HMS	Holonic manufacturing system
ISO	International standards organisation
IT	Information technology
KICAD	Knowledge infrastructure of collaborative and agent-based design
LAN	Local area network
MA	Mobile agent
MaP	Mobile agents platform
MAP	Manufacturing automation protocol
MbD	Management by delegation
MEI	Manufacturing enterprise information layer
MIB	Management information base
MMS	Manufacturing message specification
OSI	Open systems interconnection
REV	Remote evaluation
RMON	Remote network monitoring
SNMP	Simple network management protocol
TCP/IP	Transmission control protocol/internet protocol
TOP	Technical office protocol
VPDAP	Virtual physical domain agent platform
WAN	Wide area network
D_{MA}	Total MA delay
D_{SNMP}	Total SNMP delay
L_{clus}	LAN MA delay under clustering
L_{MA}	MA delay in a LAN environment
ma_{Δ}	MA size increase
ma_{INIT}	Initial MA size
$\vec{O}_{MA}$	MA itinerary
ϕ_{LAN}	LAN propagation delay
ϕ_{WAN}	WAN propagation delay
s_{LAN}	LAN speed
s_{WAN}	WAN speed
ξ_{clus}	End-to-end MA delay under clustering
ξ_{MA}	End-to-end MA delay (over WAN)
ξ_{SNMP}	End-to-end SNMP delay (over WAN)
ξ_{VPDAP}	End-to-end MA delay using VPDAP

1 Introduction

Due to the development of information technology (IT), manufacturing enterprises are facing increasingly competitive challenges in improving their capability to respond efficiently to rapidly changing customer needs. This encourages a manufacturing enterprise to integrate relevant activities with their partners and, occasionally, with their competitors. Such a movement is likely to lead to an increase in the use of heterogeneous networking products. Networks, along with the appropriate software, give manufacturing devices the ability to communicate, which in turn enables them to be managed from a remote location. As with any distributed system, networks introduce delays (transmission delays, propagation delays, etc.) that can directly affect any management system's ability to exercise effective control of a manufacturing environment.

The manufacturing automation protocol (MAP) is an established protocol for managing distributed manufacturing environments [4, 15]. MAP, like simple network management protocol (SNMP), functions within a client/server paradigm and is, therefore, subject to issues of scalability and flexibility. This could inhibit these protocols from being able to adequately manage large domains of devices in widely distributed environments. While we wish to retain the simplicity of SNMP, we also recognise the need to overcome the performance limitations of the client/server static agent approach.

Mobile agent (MA) technology represents a new climate for the future of distributed manufacturing enterprises. In a distributed environment, the network infrastructure can impose significant delays on management functions and can therefore adversely impact the ability to effectively control the manufacturing system. In this research, we propose the use of mobile agent technology as a means of improving the efficiency of managing a distributed manufacturing enterprise environment.

A number of distributed agent approaches to concurrent design and manufacturing have been proposed, such as holonic manufacturing system (HMS), knowledge infrastructure for collaborative and agent-based design (KICAD), Aglet from IBM, and Condoria from Mitsubishi [7, 9, 13]. However, most of the available commercial applications focus on the alignment between IT systems and the high-level manufacturing process flow they support. There has been little research into proposing an integrated framework to incorporate high-level company decisions with real-time shop-floor level information.

A framework is proposed in this paper based upon MA technology as a means of managing distributed manufacturing environments. The intention is to develop a strategic and flexible manufacturing enterprise network framework. The structure of this paper is outlined as follows. In Section 2, we examine the structure of MAP framework and explore the factors which must be addressed by the manager in order to maintain high network performance. Section 3 will present the concepts and characteristics of MA. In Section 4, an integration of MA technique is proposed to develop a strategic and flexible framework of managing a distributed manufacturing enterprise system. In Section 5 analytical models of network delays for both MA and SNMP are developed. These are used to compare the performance of both MA and SNMP in a distributed manufacturing management environment. The models are also used to show how our framework can optimise MA performance. Finally, the conclusions will be drawn.

2 Network management standards in the manufacturing system

2.1 The manufacturing automation protocol-MAP

In a computer integrated manufacturing (CIM) environment, various computerised tools and pieces of equipment are used in order to support the overall manufacturing plan. Communication networks are utilised in many aspects of the manufacturing environment, ranging from production planning, scheduling, and control at shop-floor level. During actual operations, however, networks for manufacturing automation experience considerable fluctuations in traffic volume. The traffic volume may change due to common machine failure or production scheduling changes. This can impact adversely on the network performance and the performance of the connected devices.

Efforts have been made to specify a communication system which provides for data connectivity between heterogeneous computer and control systems [15, 22]. Examples include:

– General Motors' manufacturing automation protocol (MAP) specified for open communication in manufacturing automation
– Boeing's technical office protocol (TOP) for office communication

Both MAP/TOP models adopt the OSI (open system interconnection) model as a general framework and are intended to provide a cost-effective implementation of a communication link between numerous manufacturing and business components for further improvements in productivity and company performance. The architecture of MAP network management, based on the OSI network management model, provides fault, performance, configuration management and event processing services. The purpose of this is to control and monitor the configuration, performance and failures of manufacturing devices, area/cell controllers and unscheduled events [10]. The actual functions of the application process are performed by agents and managers in a manner similar to the OSI network management model. Agents and managers may operate in each layer of the MAP node to collect the required data from the manufacturing resources.

2.2 Limitations of the MAP architecture

Though the MAP structure is well defined [11], research has shown that the MAP-MMS (manufacturing message specification) has not been completely successful due to its complexity, poor performance, and the high cost of implementing ISO protocol stacks [3, 12, 21]. Furthermore, based on the traditional client/server model, the protocol is limited to simple packet data exchange for control and command processes, and so is not widely used. Although the MAP was accepted as an international standard in the CIM, Internet protocols, albeit designed for generic applications, have become the dominant networking standard. Many computer systems are sold with TCP/IP as part of their specification.

Critics of the MAP-MMS have carried out much research in this field. Some researchers proposed the creation of applications of MAP-MMS in a heterogeneous manufacturing LAN environment. For example, Soudani et al. [18] evaluate the structure of the MMS in order to integrate multimedia services such as video and digital cameras. It focuses on the real time transport protocols requested by the distributed multimedia applications at shop floor level. There has been little research into producing an integrated framework to incorporate high-level company decisions with real-time shop-floor level information.

3 Mobile agent paradigm

3.1 The evolution of code mobility

The aim of using mobile code is to overcome the limitations of the traditional client/server model. The mobile code paradigm does not bind the code performing services statically; instead, it allows the migration of the code among different hosts. Applying this approach to network management reverses the logic according to which the data produced by network devices are periodically transferred to the central management station.

The key benefits of code mobility include a reduction in network traffic (under certain circumstances), the efficient utilization of computational resources, support for heterogeneous environments, and increased flexibility. Nevertheless, the use of mobile code does bring several problems. In particular, code migration introduces additional traffic into the network and requires considerable resources from the agent hosts, depending on the agent configuration and functionality [1]. The overheads caused by code migration can outweigh the benefits and make this approach inefficient. It is thus important to identify various aspects of code mobility and relate them to the network management paradigm in order to identify their potential benefits. Several models are designed to support the concept of code mobility. These are:

- Management by delegation (MbD)
- Code on demand (CoD)
- Remote evaluation (REV)
- Mobile agents (MAs)

The concept of MAs allows a program to travel between machines for remote execution, particularly in heterogeneous networks. In this research, we adopt the definition introduced by Sugauchi et al. [20]:

> A mobile agent can execute some management procedures. Moves to a network element, accesses local resource, and sends the result(s) necessary for fulfilling the network management task.

We also conclude that MAs do have certain properties, such as autonomy, learning, mobility, intelligence and co-operation. These features are important in the design and implementation of complex distributed applications.

3.2 Traveling behaviour of MAs

An MA can migrate from device to device in a heterogeneous network. It is capable of interacting with servers or other agents where services are provided, moving to another machine while carrying the intermediate results, and resuming execution when it reaches its destination.

We consider a mobile agent that performs a (trivial) management function, that is, it is despatched from the management station S and visits each managed device r_i in turn, where $1<i<k$ and retrieves a set of managed objects. After visiting the kth managed device it returns to S. The notation below describes the traveling behaviour of the mobile agent. Let O_i be the set of managed objects on each managed device r_i, such that,

$$O_i = \{o_{i1}, o_{i2}, \ldots, o_{iv}\} \tag{1}$$

where v is the number of managed objects on the device. The retrieval of the set of objects O_i from r_i, is denoted by the network transaction $\{r_{i-1}, r_i: O_i\}$. This term represents the migration of the MA from managed device r_{i-1} to r_i in order to effect the retrieval of the set of managed objects O_i from r_i. The MA itinerary for the retrieval of managed objects from the domain of managed devices is expressed as,

$$\overrightarrow{O}_{MA} = \{\{S, r_1 : O_1\}, \{r_1, r_2 : O_2\}, \ldots, \{r_{k-1}, r_k : O_k\}, \{r_k, S : O\}\} \tag{2}$$

where $\{S, r_1: O_1\}$ represents transmission of the initial mobile agent from the management station to the first managed object (S could be considered to be r_0) and $\{r_k, S: O\}$ represents the return of the MA (with all $k \times v$ lots of managed objects) from the last managed device r_i to the management station S.

It is important to note that the events denoted by the terms in $\overrightarrow{O}_{MA}$ occur sequentially, that is, the event $\{r_i, r_{i+1}: O_{i+1}\}$ does not start until the event $\{r_i, r_{i-1}, r_i: O_i\}$ has completed.

By moving to the location of an information resource, the agent can search the resource locally, which eliminates the need for the transfer of intermediate results across the network and therefore reduces network delays. Furthermore, it is assumed that the agent can choose different migration strategies depending on its task and the current network conditions, and then change its strategies as network conditions change. Hence, what is required is a set of travelling algorithms that allow an agent or a small group of cooperating agents to identify the best migration path in order to minimise the total expected time of searching for the desired information.

Literature has revealed several variations and fundamental network issues affecting both mobile agents and overall network performance, such as [2, 16]:

- The number of MAs
- The size of MAs
- The network latencies
- Probabilities of success or failure
- The task computation time at each machine
- Locality
- Concurrency

In this research, some of these factors will be taken into consideration while evaluating the performance of the MA.

4 A proposed framework

This research addresses some of the problems encountered in a practical CIM (computer integrated manufacturing) field regarding network management issues. Devising an integrated system would have the potential to overcome networking complexity in distributed manufacturing environments, which would promote an increase in productivity and a reduction in overall costs.

Through networking and telecommunication technology, manufacturing enterprises have created an integrated virtual environment where groups of companies are forming short-term relationships to collaborate on certain projects. In this virtual environment, the system design should support highly specialised, concurrent and distributed task-planning and decision-making as well as integrating business functions.

The main purpose of this research is to develop a novel infrastructure with enhanced support from the network management perspective in the area of manufacturing systems. In order to enhance the management functionality, the research proposes a new management framework for the evaluation of enterprise-oriented management across networking applications and distributed systems.

4.1 An investigation of the framework

The evaluation criteria of this research focus on management structure and human user factors to cater for cooperative enterprise processes in a manufacturing enterprise. In the manufacturing enterprise environment, the managed physical layer comprises managed components such as robots, cell controllers, numerical controllers, programmable logic controllers, and machining drilling, all of which can be interconnected through computer networks. Therefore, the intention of this research is to investigate how collaborative high-level decision-making procedures can be optimized by considering the real-time information retrieved at shop-floor level.

Similar work was conducted by Raibulet and Demartini [14]. Their research presents an industrial example by introducing a distributed database repository (DDR) model. The DDR model is constructed by a two-tier hierarchical structure and employs MA techniques to monitor and control manufacturing resources according to their physical domain classification. However, this model did not consider the associations between those managed resources and company policies or decisions at a higher management level. Therefore, without planning ahead with regard to management strategy, this model appears inefficient and time-consuming due to its linear search behaviour.

Thus, this research intends to design a framework which is more scalable and flexible, and can be easily upgraded and customised. A proposed framework is shown in Fig. 1.

4.2 Functionalities of the proposed framework

4.2.1 Manufacturing enterprise information layer

The proposed architecture adopts a multiple-manager structure in order to yield a high degree of scalability and customisation. At the MEI (manufacturing enterprise information) layer, an enterprise manager, a number of

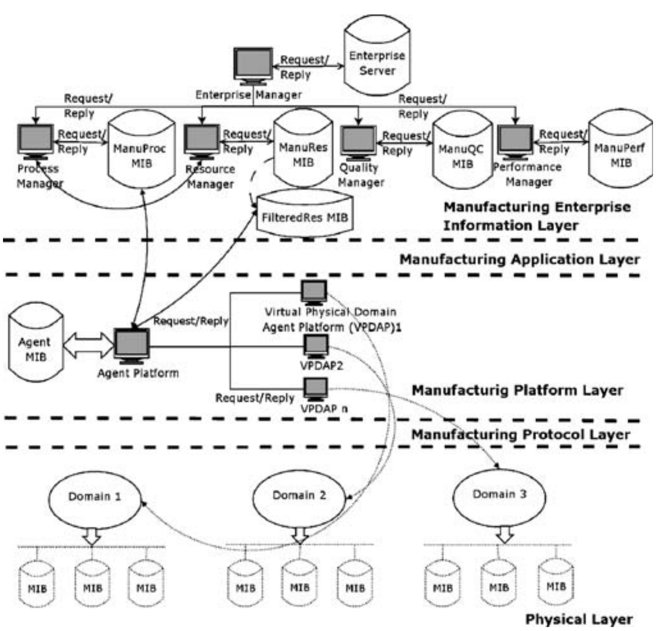

Fig. 1 The view of integrated manufacturing management framework

functional managers and several domain managers are designed to achieve the flexibility and efficiency of the system. Each manager offers a set of services (management primitives) that can be used both by applications and other managers. The major concern is to provide a strategic perspective applying to the area of business and manufacturing processes. It covers all the aspects of a manufacturing system. The basic components of the MEI layer are:

- Enterprise manager: An abstraction of the various entities that interact with various functional processes. By sending a high-level request to the respective functional manager, the enterprise manager can access manufacturing plans and information prior to planning a change. This paradigm is based on the master/slave paradigm in which control information flows from the manager station (master) to agents (slaves), while status information flows from agents to stations.
- Enterprise server: The main function of an enterprise server is to maintain resources at enterprise level including a process repository, machines and tools, and human resources. The enterprise server can also function as a communication facilitator to provide resource access information and to keep logs of all communications.
- Functional managers: Functional (process, resource, quality, performance manager, etc.) managers play an intermediate role to integrate the essential information into the applications in order to provide the enterprise manager with an overview of overall manufacturing planning. A functional manager is a form of agent that deals with requests from the enterprise manager. The functional manager also requests essential information based on these requests from the enterprise manager.
- Associated information base servers (for example, ManuProc MIB, ManuRes MIB, ManuQC MIB, and ManuPerf MIB): These are respective information bases to store general information related to all proven processes and their base descriptions and features; installed resources and services which are used to support the overall manufacturing system; quality and fault control issues and performance measurement for selected devices and services. More details in the functional descriptions for these information bases can be found in [6].
- FilteredRes MIB: This MIB is used to store relevant information reflecting the request of the enterprise manager. Associating with the ManuProc MIB and ManuRes MIB, the filtered information will then be stored in the FilteredRes MIB and used as a directory service in the implementation of the management platform layer. The information contained should include the physical location of selected resources and the degree of dependency for each individual selected resource related to the specific process, etc.

4.2.2 Management platform and physical layer

In the management platform layer, an agent-based platform architecture is presented. MaP (mobile agents platform) is a software package or platform for the development and management of (MAs) and gives all the primitives needed for their creation, execution, communication, migration and even security concerns. In addition, the MaP is concerned mainly with the processing of management data and information collection, providing key management services like monitoring and controlling. Figure 2 demonstrates a detailed MaP and its key management functions/modules are described below:

- Application dependency analysis. This dependency application mainly provides details of relevant resources accompanied by their degree of dependencies to the specific process.
- Agent manager (master). The agent manager provides the communications infrastructure that allows agents to be transmitted from and received by nodes on the network. The manager acts as the master and can launch mobile agents (slaves), identify the travelling itinerary of MAs, monitor and control the network situation, and display their results. The travelling itinerary is pre-determined according to the information retrieved from the FilteredRes MIB.
- Mobile agents (slave agents). From our perspective, MAs are software program objects with a unique ID, capable of migrating between hosts where they execute as separate threads and perform their specific management tasks. Theoretically, a MA should be designed with an itinerary table, a data folder where collected management information is stored and several methods to control interaction with polled devices.
- Directory service. A directory is essentially a specialised, server-based database and provides a lookup

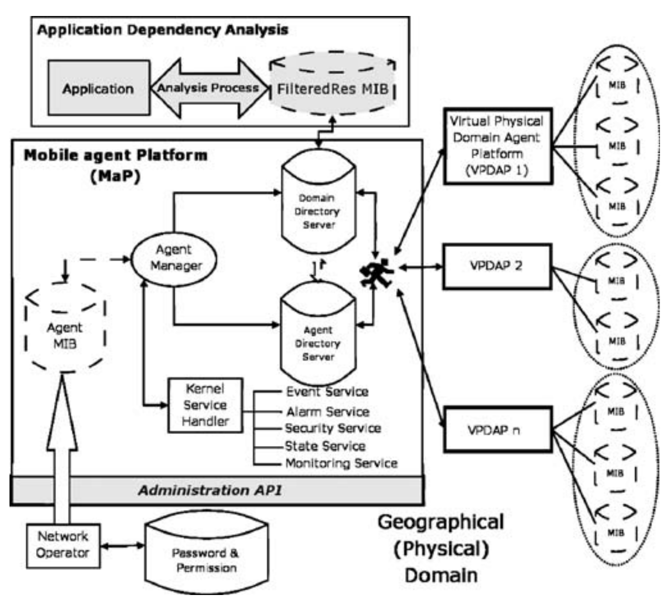

Fig. 2 Management of the mobile agent platform MaP

function. In our implementation, the directory service is categorised into two modes. One is used to store the domain information retrieved from the FilteredRes MIB, whereas another stores dispatched agents' details. The domain directory service provides the agent manager with a guideline for dispatching a number of slave agents to each different domain. As it cannot be predicted when an agent might appear, disappear or migrate unpredictably, this may cause a known agent to become unreliable for future interactions. Thus, the agent directory service records every detail of assigned agents and provides a means to control their uncertainty.

– Kernel service handler. The kernel services implement "house-keeping" operations and management services. The services include log, alarm, security, state monitoring, and event/trap services. The key functions and responsibilities of these services are described in [19].
– Virtual physical domain agent platform (VPDAP). At each physical domain, a VPDAP is designed for handling certain key management services such as agents' authentication and issues concerning the processing of the management data and information collection, monitoring and controlling, etc. Then, a physical domain manager will be responsible for providing detailed and summary information to the agent manager.
– Local MIB. Every resource (device) in the system has an associated local MIB in the managed physical layer. The local MIB contains all the detailed information that concerns the respective resource (for example, effective values of the resource parameters, resource status, resource capacity, dependency, usage) in the manufacturing system.

5 Evaluation of the proposed framework

5.1 Message exchange

In this section, the research will look at the message exchange behaviour for each of the layers. In the MEI layer, a hierarchical query model in showing the message communications is developed, as shown in Fig. 3. The service starts with an enterprise manager (human manager) invoking an operation at a user interface, resulting in a sequence of messages between the different functional managers and the associated database servers. High-level requests are sent in order to investigate the current situation of manufacturing processes before making any decisions or changes.

A filtered resource database is created after the information retrieval at the MEI layer. This database server will then turn into a directory guideline service of the management platform layer. A dedicated server is created as a domain directory server to store information about the pre-defined agent itinerary according to the information retrieved from the higher-layer. A detailed message exchange model for agents' communication at the management platform is presented in Fig. 4.

In this model, the agent manager sends out a request to the directory server and retrieves the relevant domain information. Once the related domain information is captured, several MAs will be sent out to each VPDAP after completing their registration. The VPDAP can then decide and control the number of agents being sent to each individual managed resource within the domain. In theory, the VPDAP can send out a number of agents which are equivalent to the number of required destination devices in parallel or send out an agent with a risk of overloading the traffic. Therefore, an investigation of the number of managed devices to be visited by each agent will be conducted in order to yield the optimal performance of the MA. This will be conducted below with an analysis of domain clustering.

5.2 Research analysis

An analysis of the performance of the MA travel itinerary within a distributed network environment is presented in this section. Analytical models of MA and SNMP network delays are developed based upon the network delay models in [5]. We compare the analytical results of MA performance with a client/server environment like SNMP. The function $d_{\mathrm{MA}}(\phi, s, i)$ is the network delay incurred by

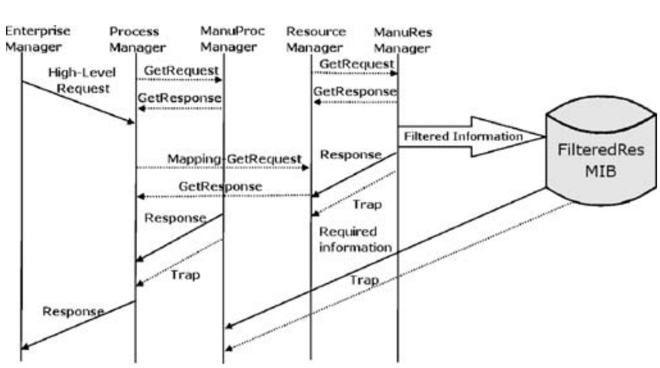

Fig. 3 A hierarchical management query model in the MEI layer

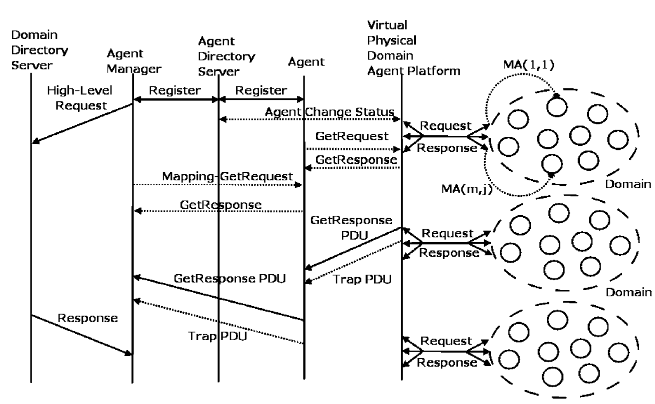

Fig. 4 A query model in the MaP and managed physical layer

an MA traveling from the ith device to the i+1th device (in order to retrieve the data from v managed objects stored on device i).

$$d_{MA}(\phi, s, i) = \phi + \frac{ma_{INIT} + ma_{\Delta}(i)}{s} \tag{3}$$

The term ma_{INIT} is the initial size of the MA dispatched from the management station. In this particular scenario the MA consists of code only and (as yet) no data. We choose a value ma_{INIT}=5 kbytes, which is based on values used in previous MA performance analysis research [8]. The function $ma_{\Delta}(i)$ returns the amount of data the MA has collected up to and including the ith device and is given by the expression:

$$ma_{\Delta}(i) = 4vi \tag{4}$$

The function $ma_{\Delta}(i)$ is implemented specifically to our application in that v managed objects are retrieved by the MA. The number of managed objects is arbitrarily set to v=10 and each object is assumed to be a 4 byte integer. The parameters ϕ and s are the network propagation delay and network speed, respectively. The expression is implemented this way so that it can be used generically to model both local area (LAN) and wide area (WAN) network environments. For example, if we say that $\phi=\phi_{LAN}$ and $s=s_{LAN}$, where ϕ_{LAN} and s_{LAN} are the LAN propagation delay and the LAN speed, respectively, then $d_{MA}(\phi_{LAN}, s_{LAN}, i)$ yields the network delay for an MA being transmitted from device i to device i+1 over a local area network. Similarly, $d_{MA}(\phi_{WAN}, s_{WAN}, i)$ represents the WAN case. The total time incurred from when the management station dispatches the initial mobile agent to reception of the MA from the last device in the domain k is given by the function:

$$D_{MA}(\phi, s, j, k) = \sum_{i=j}^{i=k} d_{MA}(\phi, s, i) \tag{5}$$

The parameter k is usually set to k=K, where K is the number of managed devices in the domain. The parameter j is typically set to j=0. However, for analysis later on, it is convenient to have the flexibility to set j and k arbitrarily. The total delay to fetch v managed objects using SNMP is given by:

$$D_{SNMP}(\phi, s, k) = 2vk\delta_{SNMP}/s + k\phi \tag{6}$$

where δ_{SNMP} is the size (in bytes) of the SNMP request and response. SNMP requests and responses are typically small. We choose δ_{SNMP}=100 bytes as a reasonable request/response size. The expression reflects our assumption that SNMP operates in synchronous mode, in that the retrieval of a managed object is not initiated before any previous managed object has been received. Furthermore, managed devices are queried sequentially.

5.3 Analysis of LAN delays

An analysis of MAs delays in a LAN environment are carried out in this subsection. The results are compared to the SNMP delays. The graph in Fig. 5 shows the results for the MA delay:

$$L_{MA}(K) = D_{MA}(\phi_{LAN}, s_{LAN}, 0, K) \tag{7}$$

The graph in Fig. 5 also shows the SNMP delay:

$$L_{SNMP}(K) = D_{SNMP}(\phi_{LAN}, s_{LAN}, K) \tag{8}$$

where $1 \le K \le 200$ is the domain size. The value used for the LAN propagation delay is ϕ_{LAN}=17 µs [17].

In our delay model we assume TCP (over IP) is the transport protocol that MA uses. However we do not explicitly consider protocol overhead, such as packetisation, connection setup and termination, flow-control or congestion control. We use a LAN throughput value of s_{LAN}=1 Mbyte/s, this being a very approximate byte rate estimate over a 10 Mb/s Ethernet, loosely taking TCP/IP protocol overhead into account. SNMP delays are dominated by propagation delays. MA is more efficient than SNMP in terms of propagation delay. MA incurs only one ϕ_{LAN} for each network transaction, whereas SNMP incurs a whole round trip time, that is $2\phi_{LAN}$ per transaction. However, in a LAN environment, propagations delays are small and SNMP delay increases linearly with domain size. MA delays on the other hand are dominated by the speed of the network. While LAN speeds are high, the amount of data that is transmitted over the network is much greater for MA (than SNMP) and increases

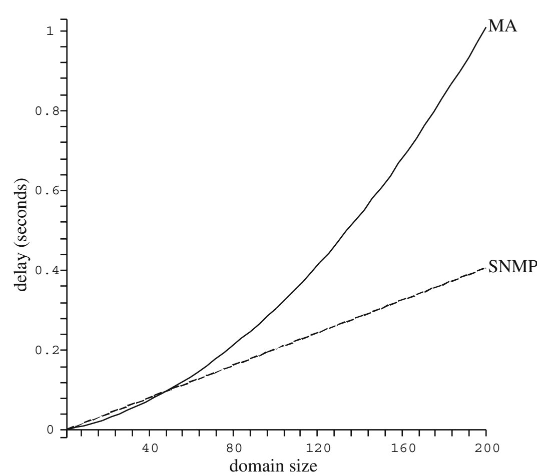

Fig. 5 MA versus SNMP delay on a local area network

exponentially with the size of domain. The results show that MA delay is lower than for SNMP only when the domain size is small, $K \leq 49$ (and even then only slightly better). The graph clearly shows the linear increase in SNMP delay versus the exponential increase in MA delay.

5.4 Clustering in a LAN environment

Given that the dominant factor of MA delay is network speed and the exponential growth of MA data transmitted over the network, we investigate *clustering* as a strategy for capping MA sizes. A (large) domain is divided into a number of clusters. An MA is dispatched (sequentially) for each cluster in the domain. While it is necessary for the management station to despatch multiple MAs (instead of just one), each cluster contains only a fraction of the managed devices in the whole domain, which will limit the growth of the data transmitted over the network. In order to analyse this, we need to consider two cases. The first is where the size of the domain k is exactly divisible by number of clusters u where, for the size of the domain k, (k mod u=0). In this case we have to dispatch u MAs for each cluster containing k/u managed devices.

$$l_{\text{clus}}(j, k, u) = uD_{\text{MA}}(\phi_{\text{LAN}}, s_{\text{LAN}}, j, k/u) \tag{9}$$

The second case is when the division k by u does not result in an integer (k mod u>0). We then must dispatch an MA for each of the u clusters containing $\lfloor k/u \rfloor$ managed devices, plus one MA for a runt cluster containing k mod u managed devices.

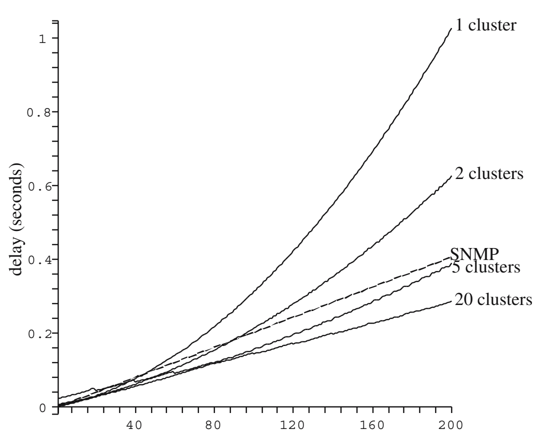

Fig. 6 MA versus SNMP LAN delays. The MA domain is clustered where U=1, 2 and 8

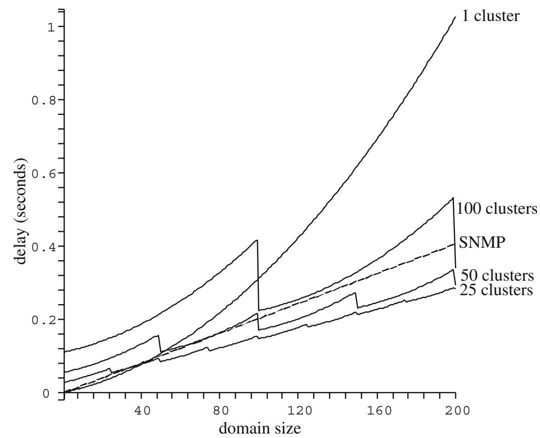

Fig. 7 MA LAN delays in a clustered environment; number of clusters U=25, 50 and 100

$$\begin{aligned}\bar{l}_{\text{clus}}&(j, k, u)\\ &= uD_{\text{MA}}(\phi_{\text{LAN}}, s_{\text{LAN}}, j, \lfloor k/u \rfloor) + D_{\text{MA}}\\ &\quad(\phi_{\text{LAN}}, s_{\text{LAN}}, j, k \, mod \, u)\end{aligned} \tag{10}$$

Therefore the delay for MAs in a clustered LAN environment is given by:

$$L_{\text{clus}}(j, k, u) = \begin{cases} l_{\text{clus}}(j, k, u) \text{ if } k \bmod u \\ \bar{l}_{\text{clus}}(j, k, u) \text{ if } k \bmod u > 0 \end{cases} \tag{11}$$

The graph in Fig. 6 shows the MA delay results for clustering (L_{clus}) where $1 \leq K \leq 200$ and U=1, 2, 5 and 20 (clusters). The result for SNMP (L_{SNMP}) is also repeated in the graph.

The graph shows that clustering reduces MA delays. Dispatching multiple MAs to small groups of managed devices overcomes the dominance of transmission times of (potentially) large MAs. Note that the 1 cluster case is essentially the no cluster case given in Fig. 5. The graph in Fig. 7 shows the results for higher numbers of clusters (U=25, 50 and 100). Here the propagation delays begin to dominate and the overall MA delays increase (above those of SNMP).

This suggests that, for any given management domain size there is an optimum number of clusters which brings about the best performance in terms of MA delay. The graph in Fig. 8 shows the MA delays for three different domain sizes K=100, 150 and 200 over a range of clusters $1 \leq U \leq K/2$.

5.5 WAN delay analysis

In this section we investigate the performance of mobile agents (compared to SNMP) when the management station and the managed devices are in separate locations.

In this scenario, the management station is attached to a local LAN, while the managed devices are attached to a

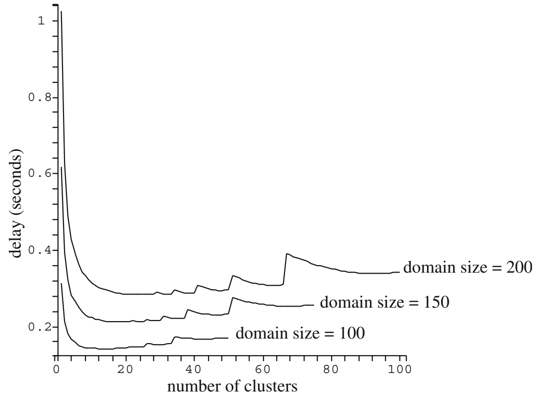

Fig. 8 MA LAN delay in a clustered environment for domain sizes K=100, 150 and 200

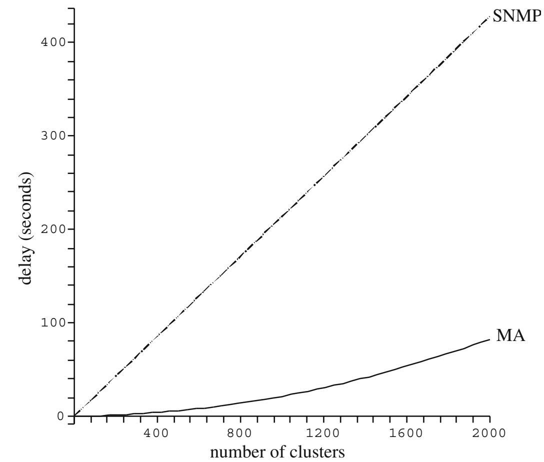

Fig. 9 MA versus SNMP delays when the management station and the domain of managed devices are separated by large geographical distances

remote LAN. Both LANs are connected via a wide area network using two (IP) routers.

When the management station dispatches the initial MA, it is transmitted across the local LAN (to which the management station is attached) to the first managed device on the remote LAN.

After visiting the first management device, the MA visits the remaining devices in the management domain. All transmissions between managed devices are entirely contained within the remote LAN environment. Upon visiting the last device k=K, the MA is transmitted back to the management station on the local LAN (across the WAN).

When an MA is transmitted from the management station to a manage device (and vice versa), it incurs the aggregate of two LAN propagation delays (one for the local and one for the remote) and one WAN propagation delay. So the total end-to-end propagation delay is given by:

$$\phi_{\mathrm{end}} = 2\phi_{\mathrm{LAN}} + \phi_{\mathrm{WAN}} \tag{12}$$

Communication between management station on the local LAN and managed device (first and last) on the remote LAN is rate limited by the speed of the WAN s_{WAN}, so the delay incurred sending the initial MA is given by $d_{\mathrm{MA}}\left(\phi_{\xi}, s_{\mathrm{WAN}}, 0\right)$. Similarly, the time taken to send the last MA is $d_{\mathrm{MA}}\left(\phi_{\xi}, s_{\mathrm{WAN}}, k\right)$. Therefore the total MA delay, including the time to send the MAs to the intermediate managed devices on the remote LAN is given by:

$$\begin{aligned}\xi_{\mathrm{MA}}(k) &= d_{\mathrm{MA}}(\phi_{\mathrm{end}}, s_{\mathrm{WAN}}, 0) + d_{\mathrm{MA}}(\phi_{\mathrm{end}}, s_{\mathrm{WAN}}, k) \\ &\quad + D_{\mathrm{MA}}(\phi_{LAN}, s_{\mathrm{LAN}}, 1, k-1)\end{aligned} \tag{13}$$

The corresponding end-to-end delay for SNMP is given by.

$$\begin{aligned}\xi_{\mathrm{SNMP}}(k) &= 2D_{\mathrm{SNMP}}(\phi_{\mathrm{LAN}}, s_{\mathrm{LAN}}, k) + D_{\mathrm{SNMP}} \\ &\quad (\phi_{\mathrm{WAN}}, s_{\mathrm{WAN}}, k)\end{aligned} \tag{14}$$

Our model assumes that router delay is negligible. The graph in Fig. 9 shows the end-to-end delay for MA and SNMP. The WAN propagation delays is s_{WAN}=100 ms and the WAN speed is s_{WAN}=200 kbytes/s (approximate speed in bytes/s of a 2.048 Mb/s E1 link). We continue to use ϕ_{LAN}=17 µs and s_{LAN}=1 Mbyte/s as the LAN propagation delay.

The graph in Fig. 9 shows the MA and SNMP delay results. In this scenario, the propagation delays of the WAN dominate the SNNP delay. The speed of the WAN is much slower than the LAN but this only affects the initial and last MA, the intermediate MAs are sent at LAN speeds.

In this analysis we consider larger domain sizes k=K where 1≤K≤2000. Even for these larger domain size, MA outperforms SNMP. However, the exponential growth of MA delays is still evident. For extremely large domain sizes (K of the order of tens of thousands of managed devices), MA delays would approach (and then exceed) those of SNMP.

This necessitates an analysis of clustering in the WAN domain also.

5.5.1 Clustering in a WAN environment

As with the LAN case, we can derive the MA delay under clustering when the size of the domain k is exactly divisible by the number of cluster u, so for k mod u=0:

$$\in_{\mathrm{clus}}(k, u) = u\xi_{\mathrm{MA}}(k/u) \tag{15}$$

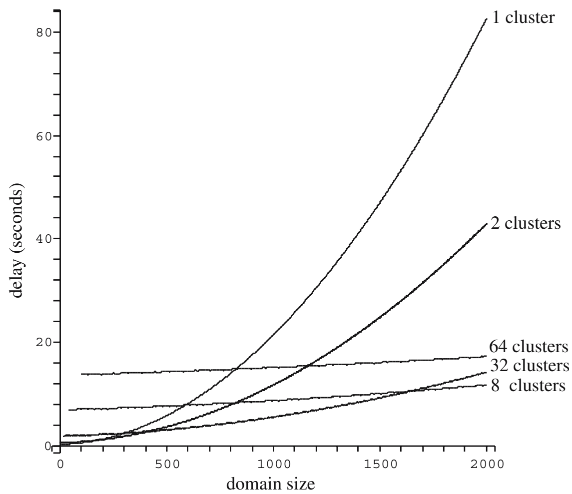

Fig. 10 MA delays in a clustered WAN environment; U=1, 2, 8, 32 and 64 clusters

and when k mod u>0:

$$\overline{\in}_{\mathrm{clus}}(k,u)=u\xi_{\mathrm{MA}}(\lfloor k/u\rfloor)+\xi_{\mathrm{MA}}(k \bmod u) \tag{16}$$

Thus ξ_{clus}, the MA delay in a cluster WAN environment, is given by:

$$\xi_{\mathrm{clus}}(k,u)=\begin{cases}\in_{\mathrm{clus}}(k,u)\text{ if } k \bmod u\\ \overline{\in}_{\mathrm{clus}}(k,u)\text{ if } k \bmod u>0\end{cases} \tag{17}$$

The graph in Fig. 10 shows the MA delay results for various cluster sizes U. Again, the no cluster case is when U=1. Initially, MA delays drop as the number of cluster is increased (U=2 and 8) but begin to increase with continued increases in U (U=32 and 64).

The graph in Fig. 11 shows MA delays over a range of clusters 1≤U≤40. It shows that for small domains (K in the low hundreds) clustering is not particularly beneficial. For larger domain sizes (K over 1,000), clustering reduces delays but only when the number of clusters is small relative to the domain size ($U/K\approx 0.05$).

5.6 VPDAP analysis

In a WAN environment, a small number of clusters will bring about an improvement in performance. However, as WAN propagation delays are much higher, performance begins to degrade as the number of clusters increases because more MAs are sent over the WAN. Clustering in a WAN environment is a trade-off between capping MA sizes (thus reducing transmission times) and propagations delays incurred by MAs being transmitted across the WAN.

Here we examine how the VPDAP can be utilised to overcome this tradeoff. If the VPDAP is co-located with managed devices on the remote LAN, then it can implement the clustering strategy on behalf of the management station. The VPDAP can be thought of as equivalent to an RMON (remote network monitoring) probe in an SNMP environment.

The management station sends the initial MA to the VPDAP (across the local LAN, the WAN and then the remote LAN). The VPDAP then dispatches an MA for each cluster of managed devices. MA delays using VPDAP to implement the clustering strategy on the remote LAN is

$$\begin{aligned}\xi_{\mathrm{VPDAP}}(k,u)\\ &=d_{\mathrm{MA}}(\phi_{\mathrm{end}},s_{\mathrm{WAN}},0)+d_{\mathrm{MA}}(\phi_{\mathrm{end}},s_{\mathrm{WAN}},k)\\ &\quad+L_{\mathrm{clus}}(0,k,u)\end{aligned} \tag{18}$$

The graph in Fig. 12 shows the MA delays ξ_{VDPAP} using the VPDAP for K=200, 1000 and 2000, and 1≤U≤40.

Comparing the results in Fig. 12 with those in Fig. 11, it can be seen from the graph MA delays flatten out as the number of clusters increases.

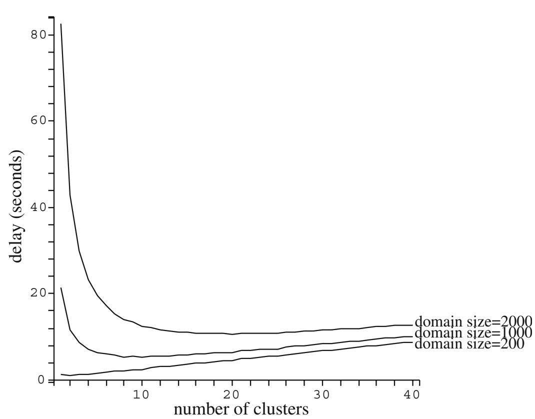

Fig. 11 MA delays in a clustered WAN environment for domain sizes K=200,1000 and 2000

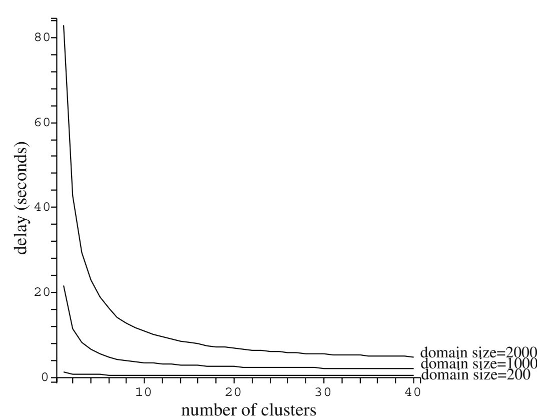

Fig. 12 MA delays using the VPDAP in WAN environment for domain sizes K=200,1000 and 2000

5.7 Discussion

In this analysis we have considered only a trivial network management function, that is, the retrieval of managed objects from a domain of managed devices. However, MAs are capable of more than just fetching data for the management station. MAs could be programmed to carry out some of the management functions. For instance, suppose that the management station computes, for each device, some metric from the set of managed objects collected by the MA. Instead, the MA could perform the metric calculation on behalf of the management station *remotely* on the managed device. This would obviate the need to return managed object data back to the management station. Instead, only the (computed) metric data is returned.

It is clear that MA technology is capable of even greater network efficiencies than we have conveyed in this paper.

6 Conclusions

In this paper a hierarchical integrated framework is proposed based upon MA technology as a means of managing distributed manufacturing environments. It is intended to resolve problems of scalability and management efficiency in large, multiple networked manufacturing domains. It is shown that using MAs overcomes the scalability and flexibility limitations of client/server protocols, such as SNMP and MAP. The proposed framework brings about network efficiencies and improvements in delay performance.

The results of the analysis in this paper show that in a distributed manufacturing management environment MA generally brings about improvements to network performance compared to a client/server technology such as SNMP. In a LAN environment MAs delays are less than SNMP but only if domain sizes are small. MA delays grow exponentially with domain size. However our analysis shows that clustering is an effective method of reducing MA delays.

MAs are particularly effective in the WAN environments. High WAN propagation delays have a significant impact on SNMP performance. Clustering in a WAN environment can bring about further improvement to MA delays. However, the number of clusters should be kept small, otherwise WAN propagation delays start to have an adverse effect on MA delay too.

The virtual physical domain agent platform (VPDAP), which is part of the proposed enterprise and manufacturing network management platform framework, is co-located with the managed devices on the remote LAN. The management station delegates the clustering functionality to the VPDAP. Therefore, the management station only has to dispatch one MA to the VPDAP (and only receive one MA from the VPDAP), thus optimising MA performance when using clustering over the WAN.

The benefit of the proposed hierarchical framework is that the high-level manager does not have to store a huge amount of information that can, over time, get proportionally larger with the development of the distributed system. Moreover, the updates or changes of the configuration information carried out over time from the manufacturing plants may be varied. This hierarchical architecture helps keep management activities more efficient and effective.

7 Future work

This research has examined how the concepts of network management and mobile agent can be utilised to improve decision making efficiency in a distributed manufacturing environment. We assume that the mobile agent can choose different migration strategies depending on its task and current network conditions, and then modify its strategies as network conditions change. One of the criteria affecting performance is the time spent locating the target managed objects, as well as the network overhead generated by the algorithm. Therefore, to examine a suitable strategy for locating the agent to the required managed devices effectively will consequently consume less search time and network overhead than normal search strategies.

References

1. Bohoris C, Pavlous G, Cruickshank H (2000) Using mobile agents for network performance. In: Proceeding of the 2000 IEEE/IFIP network operations and management symposium—the networked planet: management beyond 2000 (NOMS 2000), IEEE, Piscataway, NJ, pp 637–652
2. Brewington B, Gray R, Moisumi K (1999) Mobile agents in distributed information retrieval. In: Klusch M (ed) Intelligent information agents. Springer, Berlin Heidelberg New York, pp 335–395
3. Guyonnet G, Gressier-Soudan E, Weis F (1997) COOL-MMS: a CORBA approach for ISO-MMS. In: Proceedings of the 11th European conference on object-oriented programming ECOOP'97, Jyvskyl, Finland
4. Heuser R, Stockett M, Chen R (1993) An OSI based architecture for tracking station automation. http://lamspeople.epfl.ch/kirrmann/mms/Contacts/ RandyHeuser/tda 2/tda 2.htm. Accessed 13 June 2005
5. Holt A (2003) Theoretical performance analysis of monitoring world wide web sites. IEE Proc Commun 150(4):228–232
6. Huang C (2003) A strategic management framework for improving performance in a distributed manufacturing enterprise environment. PhD Thesis, Leeds Metropolitan University, UK
7. IBM (2005) Aglets software development kit. http://www.trl.ibm.com/aglets/. Accessed 20 June 2005
8. Adhicandra I, Pattinson C, Shaghouei EM (2003) Using mobile agents to improve performance of network management operations. Proceeding of 4th annual symposium of postgraduate networking conference (PGNET03), EPSRC, John Moores University, Liverpool, UK, 16–17 June 2003
9. Shen W, Norrie DH, Barthes AJ-P (2001) Multi-agent systems for concurrent intelligent design and manufacturing. Taylor and Francis, London
10. MAP/TOP Users Group (1987) Manufacturing automation protocol specification-version 3.0, implementation release. MAP/TOP Users Group
11. SISCO (1995) Overview and introduction to the manufacturing message specification (MMS), rev 2. SISCO, Inc., USA, pp 1–47

12. Mackiewicz R (2003) An overview to the manufacturing message specification. http://icawww.epfl.ch/MMS/. Accessed 03 September 2003
13. Mitsubshi (2005) Mitsubishi electric information technology center America upgrades CONCORDIA mobile agent framework. http://www.mitsubishielectric.com/news/1998/ 061598a.html. Accessed 13 June 2005
14. Raibulet C, Demartini C (2000) Mobile agent technology for the management of distributed systems-a case study. Int J Comp Telecomm Netw 34(6):823–830
15. Rembold U, Nnaji BO, Storr A (1994) Computer integrated manufacturing and engineering. Addison-Wesley, Reading, USA
16. Rubinstein MG, Duarte OCMB (1999) Evaluating trade-offs of mobile agents in network management. Netw Inf Syst 99:1–16
17. Partridge C (1994) Gigabit networking. Addison-Wesley, Reading, USA
18. Soudani A, Nasri S, Divoux T (2002) QoS and network resources management for communication in distributed manufacturing processes. Comput Ind 1607:1–15
19. Stamatelopoulos F, Chiotis T, Maglaris B (1995) A scalable, platform-based architecture for multiple domain network management. In: Proceeding of the IEEE international conference on communications (ICC'95), Seattle, WA, USA
20. Sugauchi K, Miyazaki S, Covaci S, Zhang T (1999) Efficiency evaluation of a mobile agent based network management system. In: Proceeding of the IS and N '99-6th international conference in intelligence in services and networks, Barcelona, pp 527–535
21. Tuttle H (2003) EMGT 862 manufacturing systems integration. Lecture notes, University of KU, USA
22. Umar A (1993) Distributed computing and client-server systems. PTR Prentice-Hall, New Jersey, USA