

DMTF DMI 2.0

PC Server

(金 鍾瑞)

1998

DMTF DMI 2.0

PC Server

Design and Implementation of a PC
Server Management System
Using DMTF DMI 2.0

Design and Implementation of a PC Server Management System Using DMTF DMI 2.0

by

Jong-Seo Kim

Department of Computer and Communication Engineering
POSTECH Graduate School of Information Technology

A thesis submitted to the faculty of POSTECH Graduate
School of Information Technology in partial of the requirement
for the degree of Master of Engineering in the Department of
Computer and Communication Engineering

Pohang, Korea
November 27, 1997
Approved by

Major Advisor

DMTF DMI 2.0

PC Server

1997 11 27

()

()

()

MCC
9551M09 , Jong-Seo Kim, Design and
Implementation of a PC Server Management
System Using DMTF DMI 2.0, DMTF DMI 2.0
PC Server
, (), 1998, 54P,
Advisor. Won-Ki Hong, Text in Korean.

ABSTRACT

Recently, networked PC desktops and servers have become very complex and difficult to manage. Therefore, the need for managing information, computing resources and networked systems has never been greater. Without solutions that are reliable, sophisticated, interoperable and easy to use, managing them becomes almost impossible. Management is critical for the next generation of computing to emerge.

Recently, many attempts have been made to enable management for networked PCs. Some work is in progress to enable management for networked PCs using DMTF DMI. DMI is complementary to existing network management standards, such as SNMP, which already have been widely used for managing networking devices. DMI maps to SNMP-based consoles to access the desktop in a uniform manner. Both can work together as an integrated, cohesive solution to deliver information to the LAN administrator, and also to help individual users provide accurate diagnostic information to remote technicians. DMI was designed as an industry standard for PC management by DMTF which is an industry-wide consortium committed to making PCs easier to use, understand, configure and manage.

In this thesis, we present the design and implementation of a DMI-based PC server management system which can manage NT-based systems easily. Our prototype implementation has been carried out using Microsoft Windows and database which have been developed by us. We also present performance evaluation of the developed management system

ABSTRACT	I
.....	II
.....	III
.....	IV
1.	1
1.1	1
1.2	2
2.	3
2.1 (System Management)	3
2.1.1	3
2.1.2	4
2.2 DMI (Desktop Management Interface)	7
2.2.1	7
2.2.2	7
3.	13
3.1	13
3.2	15
4.	17
4.1	17
4.1.1 Service Provider (SP)	17
4.1.2 MA (Management Application)	19
4.2	22
4.2.1	22
4.2.2 Service Provider	23
4.2.3 Management Application (MA)	25
5.	27
5.1	27
5.2 Service Provider (SP).....	27
5.2.1 MI Module	27

5.2.2 CI Module	29
5.2.3 Core Module	30
5.3 Management Application.....	34
5.3.1	34
5.3.2 (User Interface).....	34
5.3.3 Management Application Functions (MAF).....	35
5.4	41
5.4.1 Auto-discovery Ping	41
5.4.2 Security	41
5.4.3 Event Mail Delivery	42
5.4.4 Event	44
6.	45
[].....	47
[: PC].....	49
1. (SMFA).....	13
2. (SMF).....	16
3. MANAGEMENT INTERFACE	16

1.		4
2.	SNMP	5
3.	DMI	5
4.	SNMP DMI	6
5.	SNMP DMI	6
6.		8
7.	MIF	9
8.	SERVICE PROVIDER	10
9.	MANAGEMENT INTERFACE	11
10.	COMPONENT INTERFACE	12
11.	OSI	14
12.		15
13.		22
14.	SERVICE PROVIDER	23
15.	MANAGEMENT APPLICATION	25
16.	MANAGEMENT INTERFACE	26
17.	RPC SERVER & CLIENT APPLICATION	28
18.	MI	28
19.	MI	29
20.	CI	29
21.	SP CORE MODULE	30
22.	DB	31
23.	MEMORY	32
24.	SERIALIZATION	33
25.	MANAGEMENT APPLICATION	34
26.		35
27.	MAF	36
28.	PING AUTO-DISCOVERY	42
29.	DMIREGISTER DMICHECKREGISTER	43
30.	EVENT	44
31.	DMTF	45
32.	가	46
33.	WBEM	46
34.	(TESTBED)	49
35.	AUTO-DISCOVERY	50
36.	SP CI-CODE	50
37.		51
38.	EVENT VIEWER (EV)	52
39.	CONSOLE EVENT LOG	52

1.

1.1

가 . , , PC

가

Local Area Networks (LAN)

가 .

PC (server),

(desktop) (portable) .

3

가 가

가 가 .

Internet Engineering Task Force (IETF) Simple Network
Management Protocol (SNMP) [Wil93, Wil96] , .

SNMP (host
system), (router), (hub), (gateway)

(network infrastructure device)

PC PC

Desktop Management Task Force (DMTF) Desktop

Management Interface (DMI) , Ver2.0 [Dmt96a, Dmt96b].

PC (client) PC (domain) 가 . , PC .

1.2

가 , PC PC .

가 . DMTF DMI ver2.0 PC MS-windows .

.1 DMI . 2 3

. 4 , 5 4 6 .

2.

DMI

2.1 (System Management)

(System Management)

2.1.1

(network management)

(servers, desktops, and portables) (operating systems and applications)

(physical infrastructure)

(hubs),

(switch),

(router)

(gateways)

가

(management application)

(managed systems)

(management console)

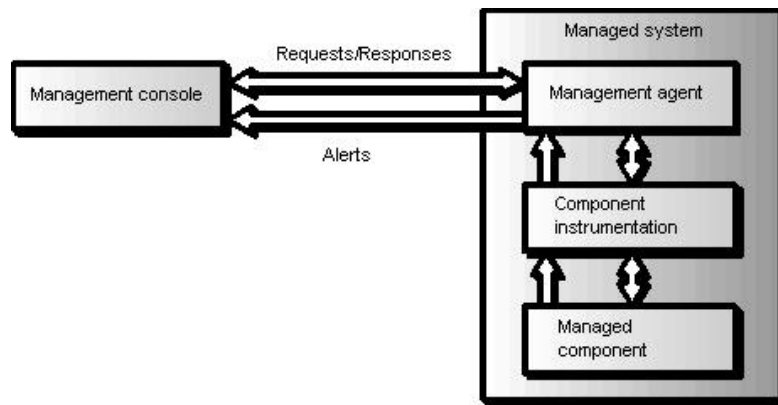
(request)

(response)

가

console

3 가
management agent, component instrumentation managed component
. Management agent console
console request component component
instrumentation response event console
. component instrumentation console request
managed component
가 . Managed component
component 가 . 1 .



1.

2.1.2

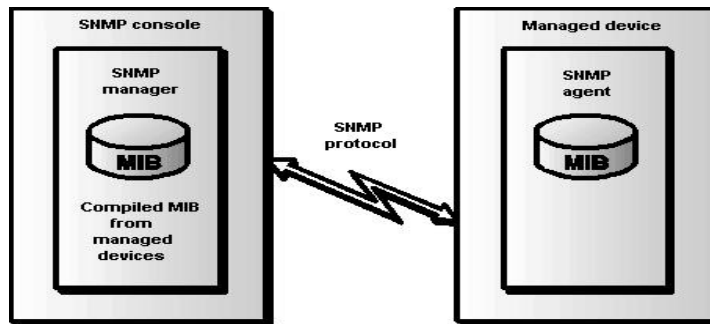
가 가
가
가
가 가
가 가

SNMP [Wi193, Wi196] DMTF DMI [Dmt96a, Dmt96b]가

SNMP 1988 IETF

. 1988 Ver 1.0

1993 Ver 2.0



2. SNMP

SNMP 가 (host)

(, ,

PC

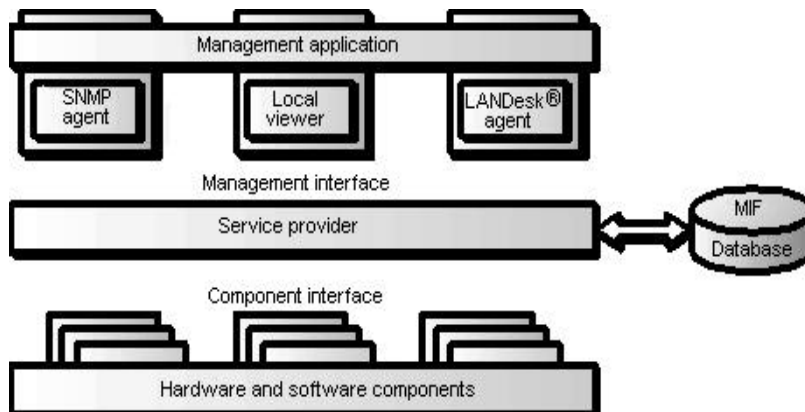
PC

DMTF 1996 1 DMI Ver 1.1

1996

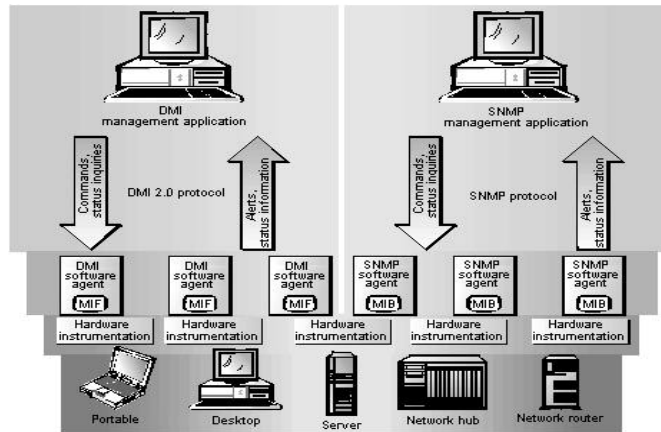
3 Ver 2.0

3



3. DMI

SNMP DMI 가 Console managed device 가 , managed device event console . 4 , managed device hardware instrumentation, software agent management database (MIFs or MIBs) .



4. SNMP DMI

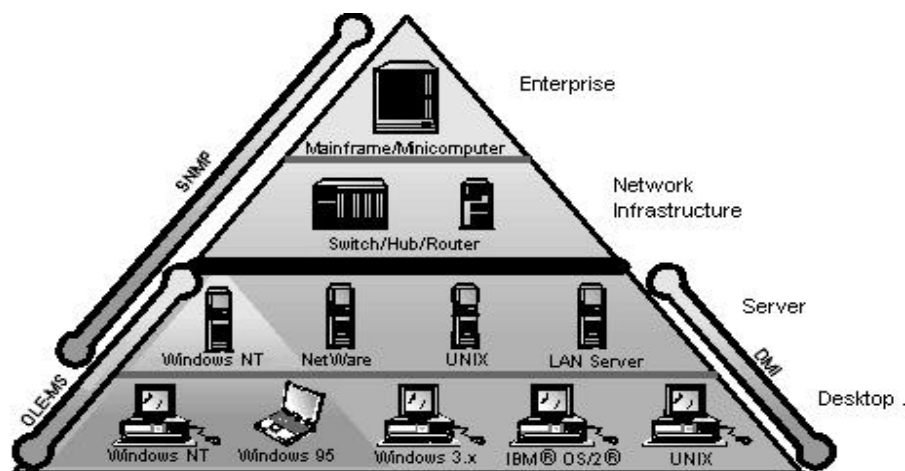
, SNMP

DMI

PC

, SNMP DMI

5



5. SNMP DMI

2.2 Desktop Management Interface (DMI)

DMTF

DMI

2.2.1

PC () ,
PC (install), (uninstall),

PC

DMTF DMI

DMTF PC

(consortium)

가

1996 1 DMI Ver 1.1

(SDK)

, 1996 3 DMI

Ver 2.0

DMI Ver 2.0

DMI Ver 2.0

2.2.2

DMI 4 가

[Dmt96b].

?

(format)

Management Information

Format (MIF)

, . SNMP

Management

Information Base (MIB)

? Console

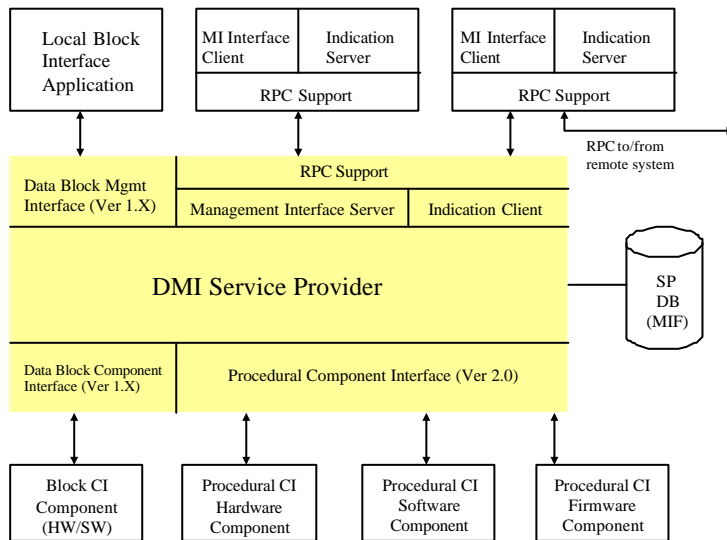
Service Provider 가
agent

? 가 가 ,
- Service Provider console Management Interface (MI)
- Service Provider (Component) Component Interface (CI)

? (Remote Communication) 가 가
DMI Ver 2.0 Remote Procedure Call (RPC)

4

6

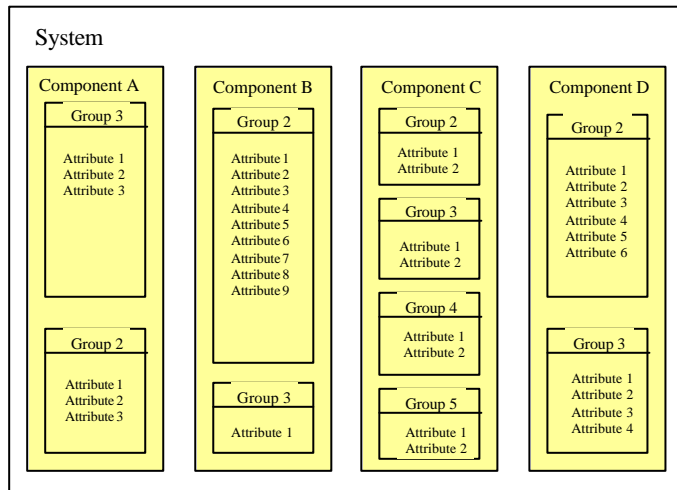


6.

6 component service provider
MIF database , service provider
component , management application service
provider RPC service provider 가
remote ,
management application local remote service provider

2.2.2.1 Management Information Format (MIF)

component 가 ,
component group 가 , group
attribute 가 . 7
, group table attribute
key 가 row column



7. MIF

2.2.2.2 Service Provider

Service provider (SP) management application (MA) component
 component instrumentation code (CI-code) request
 event . , CI-code component
 component
 . SP 가

? MA CI-code (install) (uninstall)

? MIF data

? MA CI-code request event

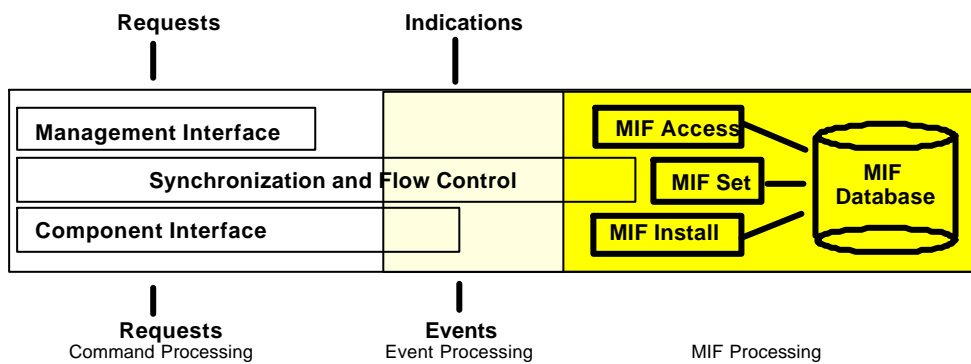
? CI-code event filtering

? MA event MA event

? MA component ID 1

? unicode

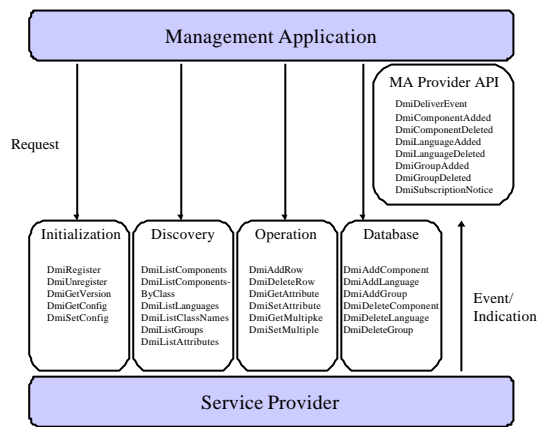
8 SP



8. SERVICE PROVIDER

2.2.2.3 Management Interface (MI)

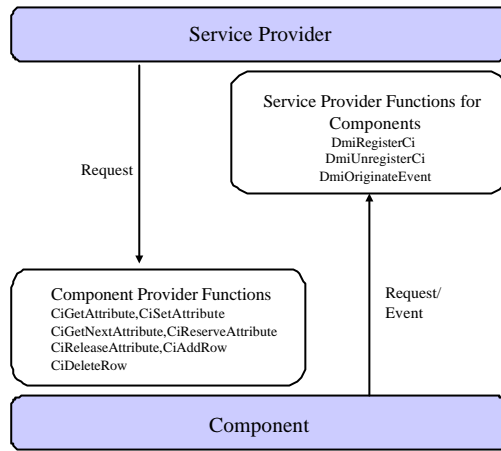
Management Interface (MI) SP MA SP
 가 MA 가 . SP 가
 MA (request/response)
 , MA 가 component event 가
 SP MA indication
 9



9. MANAGEMENT INTERFACE

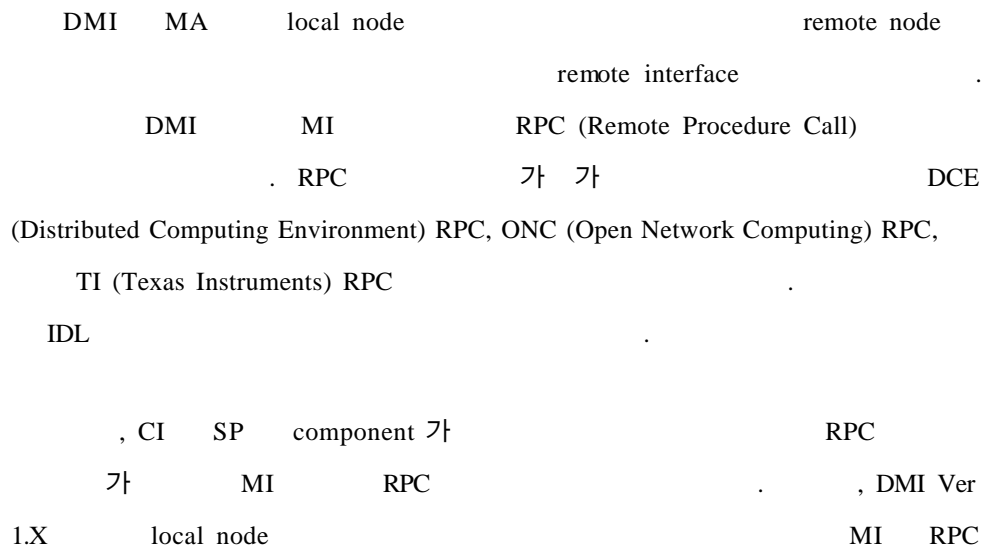
2.2.2.4 Component Interface (CI)

Component Interface (CI) SP component
 SP 가 component 가
 SP 가 component event 가
 SP CI-code SP
 component 가 SP 가
 MA (request) component
 10



10. COMPONENT INTERFACE

2.2.2.5 Remote Interface



3.

3.1

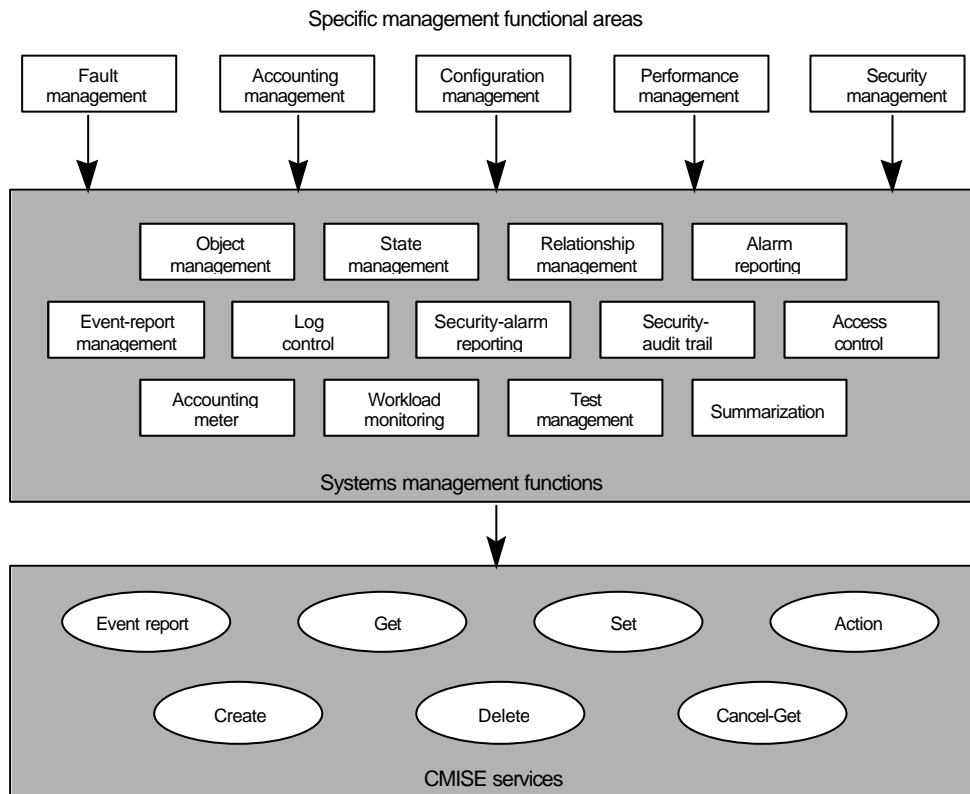
가 . Open Systems Interconnection (OSI)
 가 (System Management Functional Area: SMFA)
 [1].

SMFA	
Fault management ()	(detection), (isolation), (correction) (alarm collection) (analysis), (testing), (locating)
Accounting management ()	(tariffing function) (billing function)
Configuration management ()	,
Performance management ()	, 가
Security management ()	,

1. (SMFA)

SMFA (System Management Functions:
 SMF) , SMF 가
 SMFA . SMF

SMFA
(event-report-management function) SMFA
SMFA SMF SMF
SMF , SMF CMIS (Common
management information service) [Wil93]
(mapping) 11
SMF SMFA
SMF
(object-management function)
, SMF CMIS
[Wil93, Cct92b]

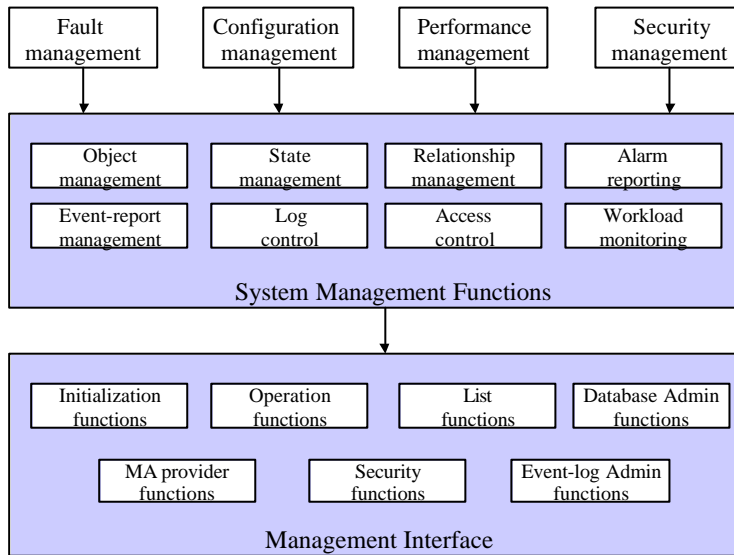


11. OSI

3.2

OSI

DMI Spec MI CI 가 3 가
 가
 가
 MI 가
 12



12.

12 4
 8 OSI
 가
 MI MI
 2 3

Object management ()	
State management ()	
Relationship management ()	
Alarm reporting ()	
Event-report management ()	가
Log control ()	
Access control ()	(management information) (management operation)
Workload monitoring ()	

2. (SMF)

MI	
Initialization functions	
Operation functions	
List functions	
Database Admin functions	
MA provider functions	event indication
Security functions	
Event-log Admin functions	event indication

3. Management Interface

4.

4.1

Service Provider (SP) Management Application (MA)

4.1.1 Service Provider (SP)

SP components
DMI Spec
MI for MA
CI for
DMI Spec
Security , Ping , Event Log

4.1.1.1 MI (Management Interface) for MA (Management Application)

MI DMI Ver 2.0 interface , 2
DMI Spec [Dmt96b]

? Initialize Functions: MA SP 가
SP
SP 가 가 (version,) 가

? Listing Functions: MA SP component 가
component group attribute
가 , SP component
list group, attribute list

? Operation Functions: attribute 가
 , SNMP Get/Set
 , attribute table table 가,

? Database Administration Function: SP component group
 가 , SP component
 component group attribute database

4.1.1.2 CI(Component Interface) for Components

CI DMI Ver 2.0 interface , 2
 DMI Spec [Dmt96b]

? Initialize Function: CI-code (Component Instrumentation Code)가 SP CI-code
 가 component .

? Event Function: Component SP
 event .

4.1.1.3 (Security)

DMI Spec MA 가 SP attribute 가
 SP MA 가 .
 SP SP 가
 가 가 , MA 가 SP
 가 .

4.1.1.4 Ping

MA . MA SP 가
SP 가

4.1.1.5 Event Event (log)

SP component event 가 , MA
가 . , event SP
database log MA

4.1.2 MA (Management Application)

MA SP attribute
Console SP event ,
event Event

4.1.2.1 Console

SP component component group, attribute
가 .
Get/Set/List , SNMP Get/Set/GetNext
.
, MA security SP 가
auto-discovery .

4.1.2.1.1 (Get)

SP attribute 가
attribute attribute 가
attribute table keylist keylist
attribute 가

4.1.2.1.2 (Set)

Get SP attribute
, attribute table keylist
attribute

4.1.2.1.3 (List)

SP install component list component
group, attribute list 가 MA
SP component list

4.1.2.1.4

가 MA , 가
, 가 super-user
가, 가 가
password

4.1.2.1.5 (Auto-Discovery)

MA SP 가
MA 가 (domain)
SP 가

SP 가

4.1.2.2 Event

SP event SP
database (DB) event
SNMP Trap

4.1.2.2.1 Event (Alarm)

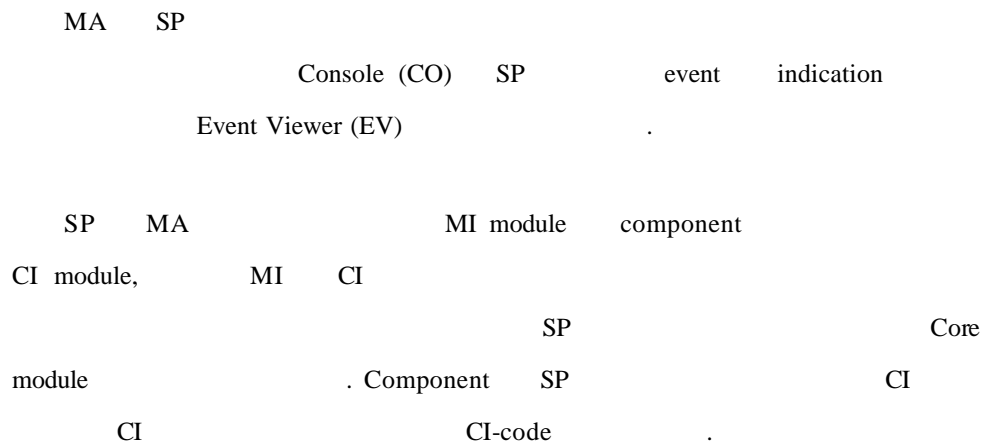
component event 가 component SP
SP event DB
MA MA event
가

4.1.2.2.2 Event (Get)

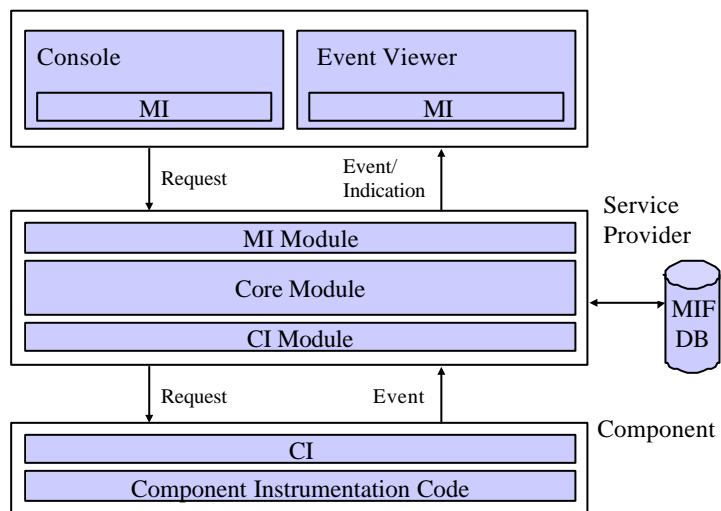
Component event SP DB
가 가
가 가
(filtering)

4.2

4.2.1



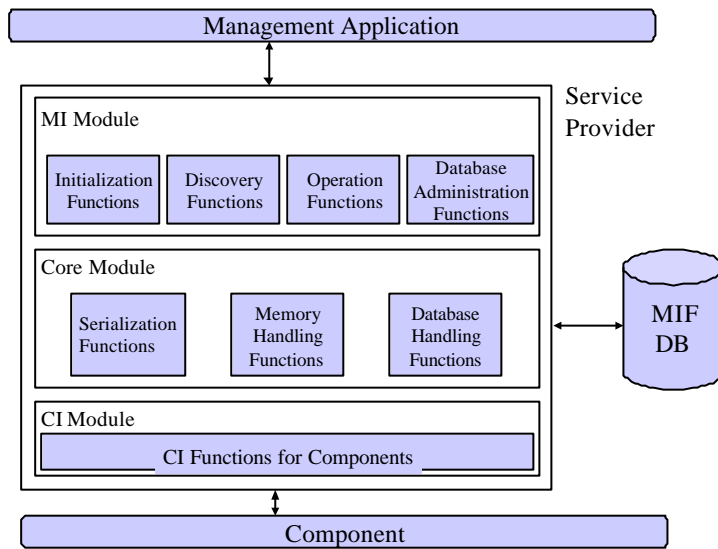
13



13.

4.2.2 Service Provider

3 가 module core module SP
 (serialization, DB handling, memory handling)
 SP . MI module MA interface
 CI module Component interface . 14



14. SERVICE PROVIDER

4.2.2.1 Core Module

Core module SP 가
 ? Serialization Functions: SP MA
 component , MA
 component event 가
 request event (serialization)

? DB Handling Functions: component MIF
MIF SP 가 MIF DB . MIF
DB MIF , MIF
read/write , MIF DB

? Memory Handling Functions:

DMI Spec .
(malloc) (free) 가
library

4.2.2.2 MI Module

SP MA interface DMI Spec .
가 가
, .
MI MA MA MI 가
. server
client

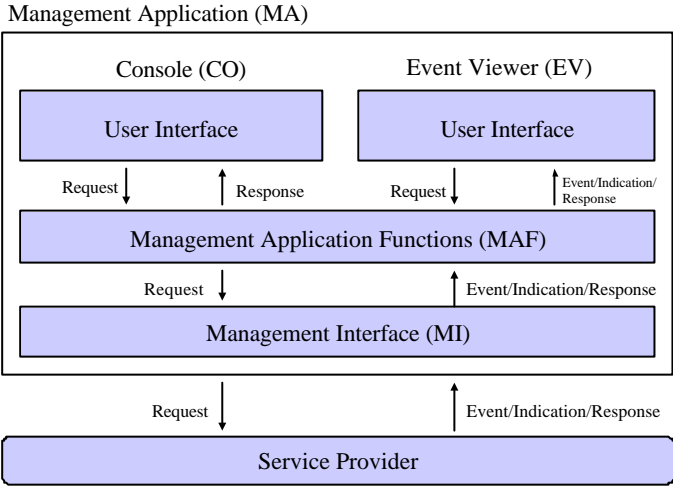
13 .

4.2.2.3 CI Module

SP component interface DMI Spec
core module . SP CI
CI-code . , CI-code SP 가
CI .

4.2.3 Management Application (MA)

MA SP attribute
 Console (CO) SP event
 event log Event Viewer (EV)
 15



15. MANAGEMENT APPLICATION

CO EV
 User Interface (UI)
 , Management Application Functions
 (MAF) MI UI 가 , UI
 DMI
 MI , MI DMI UI
 UI ,

4.2.3.1 User Interface (UI)

Graphical User Interface (GUI)

MS-windows

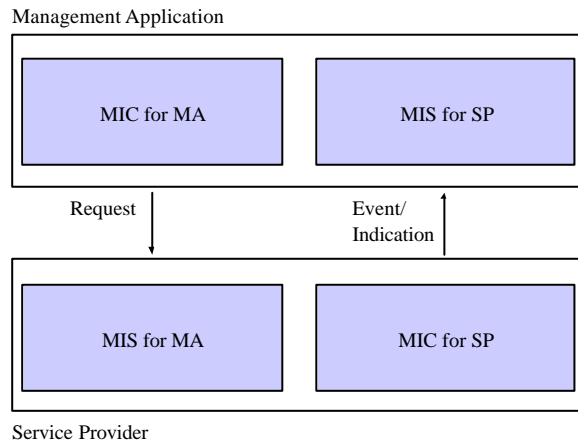
4.2.3.2 Management Application Functions (MAF)

UI, MI
CO EV 가
C++ (class)

4.2.3.3 Management Interface (MI)

SP UI SP
SP event indication UI
2 Management Interface Client (MIC) for MA
Management Interface Server (MIS) for SP 가 . MIC for MA SP
SP MA
MIS for SP SP event 가 MA
SP

16



16. MANAGEMENT INTERFACE

5.

5.1

PC . PC windows
NT windows
OS MS-Windows
Microsoft Visual C++ [Mic95, Mic97]
PC windows 95

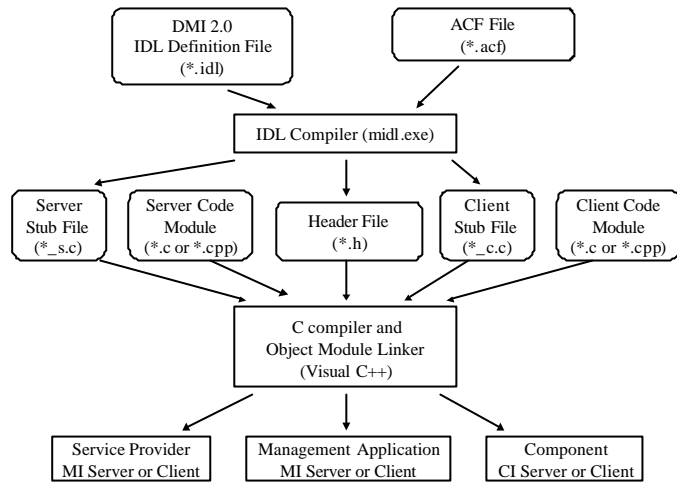
5.2 Service Provider (SP)

SP core module, MI module, CI module
module

5.2.1 MI Module

MI DMI RPC [War93]
RPC DCE RPC, ONC RPC, TI RPC 가
DCE RPC MS (Microsoft) RPC
RPC Interface Definition Language (IDL)
DMI DCE RPC IDL

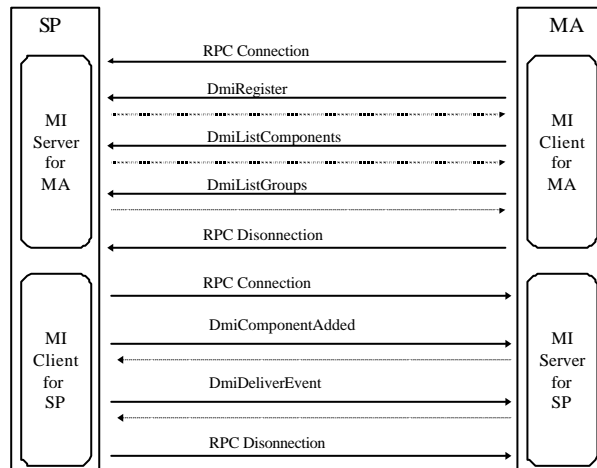
MI RPC stub
RPC server server , RPC client client



17. RPC SERVER & CLIENT APPLICATION

MI server MI 가 RPC server
 MI client 가

18

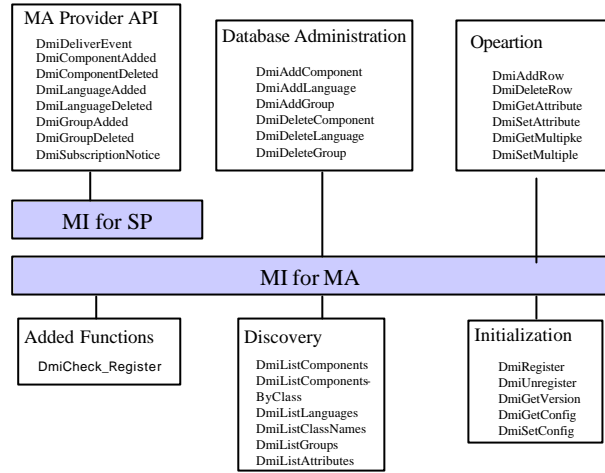


18. MI

MI DMI Spec

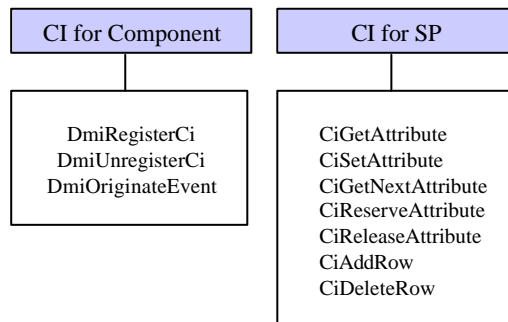
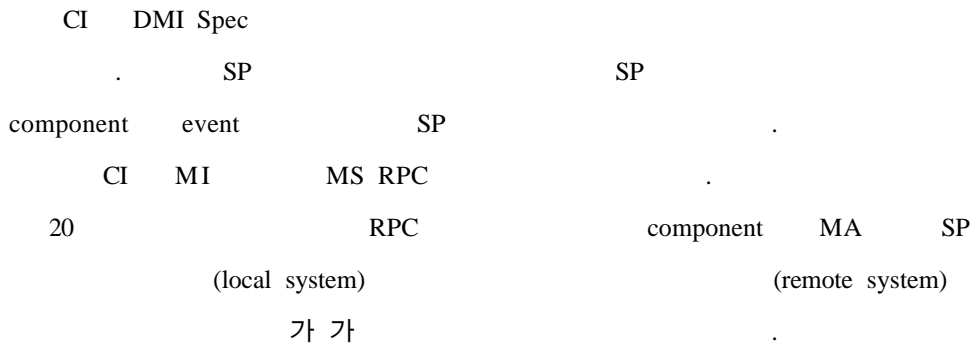
security

가 가 [19].



19. MI

5.2.2 CI Module



20. CI

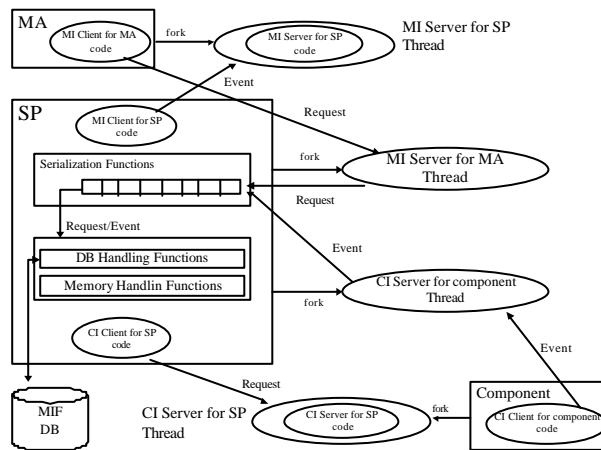
5.2.3 Core Module

SP serialization DB handling ,
 memory handling RPC server
 (thread) .

5.2.3.1

SP . MA
 MI , component CI
 가 (main) 가 [21].

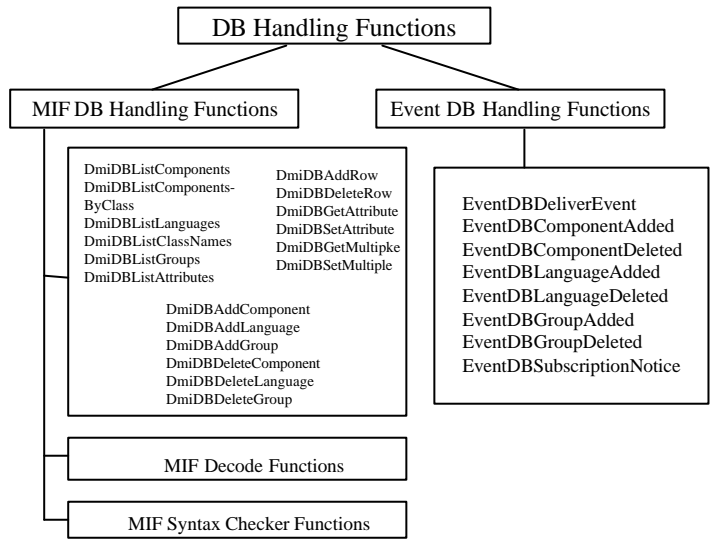
SP 가 module 가
 DB memory,
 , MI-server-thread CI-server-thread MA
 component (background)
 child-thread 가
 가
 (integrity) 가 가
 serialization .



21. SP CORE MODULE

5.2.3.2 DB Handling Functions

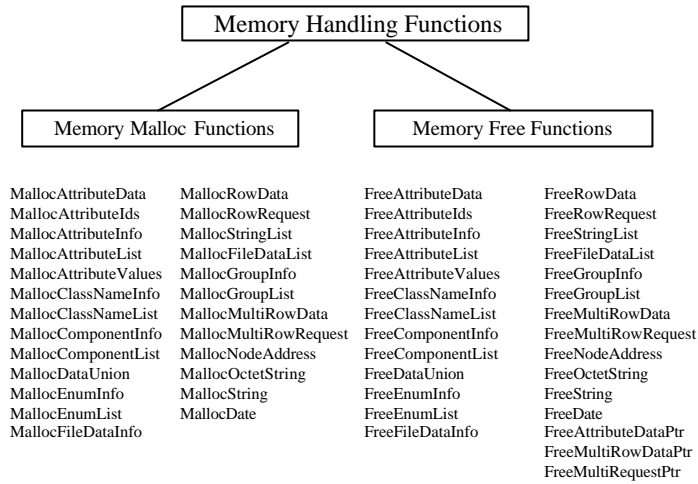
SP 2 가 DB .
 component MIF component event SP
 indication DB .
 DB 2 가
 22 .
 SP DB General Management
 Information Repository Service (GMIRS) DB .



22. DB

5.2.3.3 Memory Handling Functions

SP DMI Spec .
 Spec (malloc) (free)
 (type)
 C malloc ,
 C free



23. MEMORY

5.2.3.4 Serialization Functions

SP MA component
event

가

single-writer/multiple-reader guard (SWMRG)

[Jef96].

SWMRG C - /

(single-writer/multiple-reader)

MS-Window MUTEX (Mutual Exclusion), Semaphore, Event

SWMRG

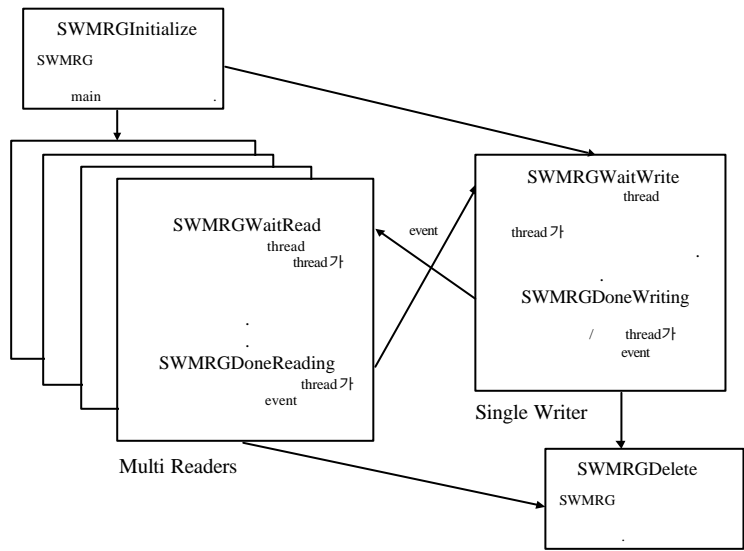

```
// The single -writer/multiple -reader guard
// compound synchronization object
typedef struct SingleWriterMultiReaderGuard {

    //          (Mutex)          가
    //          (access)          .
    //          가
    HANDLE hMutexNoWriter;

    //          (event)          가          가
    HANDLE hEventNoReaders;

    //          (semaphore)          가
    //
    //
    HANDLE hSemNumReaders;

} SWMRG, *PSWMRG;
```



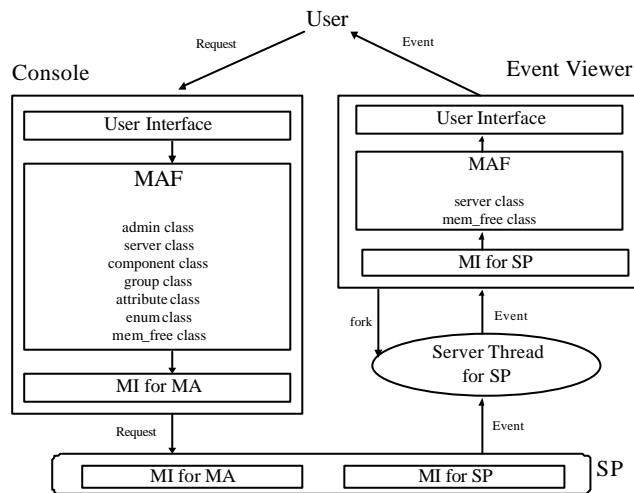
24. SERIALIZATION

5.3 Management Application

MA

5.3.1

Console (CO) UI MAF DMI
 MI SP ,
 UI , Event Viewer (EV) EV가
 SP event event 가
 MAF UI
 event
 25 .



25. MANAGEMENT APPLICATION

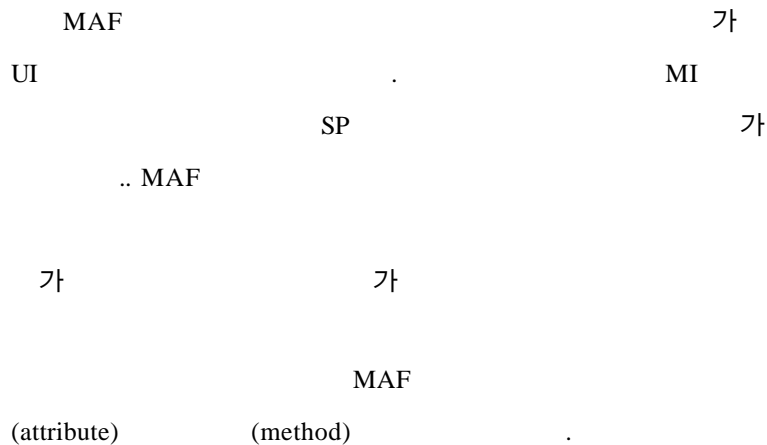
5.3.2 (User Interface)

Microsoft Microsoft Foundation Class
 (MFC)
 (dialog box)
 CTreeView [Mic97a] CListView [Mic97a] , MS-windows
 [26] .

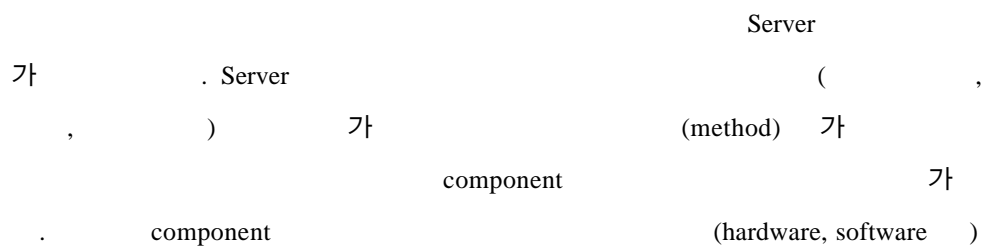


26.

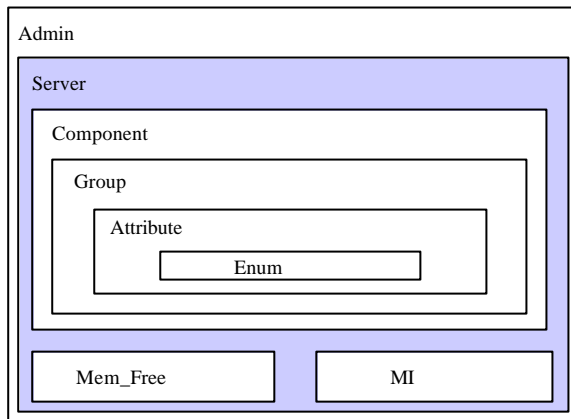
5.3.3 Management Application Functions (MAF)



5.3.3.1



component component 가 .
 component 가 group
 가 group
 attribute 가 가
 가 MAF 27



27. MAF

5.3.3.2 Admin

Console
 (domain) SP 가
 server

? Auto-discovery: (B class) SP 가

? Ping: SP 가

? Add/Delete Server:
 가

```

class CAdmin : public CWnd
{
// Construction
public:
    CAdmin();
    CAdmin(char* dom, int bFrom, int bTo, int cFrom, int cTo);
    virtual ~CAdmin();

// Attributes
protected:
    char        servers [MAX_SERVERS][32];
    int         serverNo;
    int         bFromIp, bToIp, cFromIp, cToIp;
    char        domain [32];

// Operations
public:
    void        Set_Parameter (char* dom, int bFrom, int bTo,
                               int cFrom, int cTo);
    int         Search_Servers ();
    int         Ping_Server (char* addr);
    int         Get_Server_No ();
    int         Get_Servers (Managed_Agent_t      *srv);

    int         Add_Server (char*      addr);
    int         Delete_Server (int      serverId);
    int         Exist_Server (char*      addr);

    int         Search_Proc ();
    int         createSenderServer ();
    int         freeSenderServer ();
    int         createReceiverServer ();
    int         freeReceiverServer ();
};

```

5.3.3.3 Server

```

(      , IP      ,
)      가      component      가      MI
      SP      가      component
      , Mem_free

```

```

class CServer
{
public:
    CServer();
    ~CServer();
    CComponent*      pComponent;
    DMI_Mem_Free     Dmi_Free;
    int              Server_Number_Of_Components;
protected:
    unsigned char*   pszNetworkAddress;
    char*            DMI_Server_Description;
public:
    char*            Get_Server_Description ();
    int              Disconnect_From_Server ();
    int              Connect_To_Server(char *addr);

    int Verify_DMI_Agent();
    int Get_Component();
    int Get_Group(int compld);
    int Get_Attribute_Structure(int compld,int grpId);
    int Get_Table_Attributes(int compld, int grpId);
    int Set_Table_Attributes(int compld, int grpId, int row);
    int Change_Password(char* id, char* old_pass, char* new_pass);
    int Check_Password_Register (char* id, char* passwd);

    int OnCListAttr(int compld, int grpId, DmiAttributeList_t **reply);
    int OnCListComp(DmiComponentList_t** reply);
    int OnCListGroup(int compld, DmiGroupList_t** reply);

    int OnDAddComp(char *name, char *body, int type, int *cid);
    int OnDAddGroup(char *name, char *body, int type, int cid, int* gid);
    int OnDDelComp(int cid);
    int OnDDelGroup(int cid, int gid);
    int OnOAddRow(int cid, int gid, DmiRowData_t* rowData);
    int OnODelRow(int cid, int gid, DmiRowData_t* rowData);

    int OnOGetAttr(int cid, int gid, int aid);
    int OnOGetMulti(DmiMultiRowRequest_t* request,
                   DmiMultiRowData_t** rowData);
    int OnOSetAttr(int cid, int gid, int aid);
    int OnOSetMulti(DmiMultiRowData_t* rowData);
};

```

5.3.3.4 Component, Group, Attribute

	SP	component, group	attribute
		가 . 가	server
MI	SP	가	server

```

class CAttribute
{
public:
    CAttribute();
    ~CAttribute();
    void DMI_Attribute_Set_Parameter( ... );
    void DMI_Attribute_Set_Data_Type( ... );
    void DMI_Attribute_Set_Data_Value( ... );

    unsigned long    Attribute_ID;
    char*            Attribute_Name;
    char*            Attribute_Pragma;
    char*            Attribute_Description;
    unsigned int     Attribute_Type;
    void*            value;
};

class CDMI_Group
{
public:
    CGroup();
    ~CGroup();
    void DMI_Group_Set_Parameter( ... );
    void DMI_Group_Set_Parameter( ... );
    void DMI_Group_Malloc_Attribute(int Number_Of_Attributes);
    void DMI_Group_Malloc_Table(int Number_Of_Rows);

    CAttribute*      pAttribute;
    CAttribute_Array* pTable;

    int              Group_Number_Of_Attributes;
    int              Group_Number_Of_Rows;
    int              Group_Number_Of_Columns;
    unsigned long    Group_ID;
    unsigned long    Group_KeyNo;
    char*            Group_Name;
};

```

```

class CComponent
{
public:
    CComponent();
    ~CComponent();
    void DMI_Component_Set_Parameter( ... );
    void DMI_Component_Set_Parameter( ... );
    void DMI_Component_Malloc_Group(int Number_Of_Groups);

public:
    CGroup*          pGroup;
    int              Component_Number_Of_Groups;
    unsigned long    Component_ID;
    bool             Component_ExactMatch;
    char*            Component_Name;
    char*            Component_Pragma;
    char*            Component_Description;
};

```

5.3.3.5 Mem_Free

DMI

NULL . ,

```

class Mem_Free
{
public:
    Mem_Free();
    virtual ~Mem_Free();

public:
    void DMI_Structure_Free (DmiString_t*          str);
    void DMI_Structure_Free (DmiOctetString_t*     str);
    void DMI_Structure_Free (DmiTimestamp_t*       date);
    void DMI_Structure_Free (DmiFileTypeList_t*    fileType);
    void DMI_Structure_Free (DmiComponentList_t*   list);
    void DMI_Structure_Free (DmiAttributeList_t*   list);
    void DMI_Structure_Free (DmiAttributeInfo_t    info);
    void DMI_Structure_Free (DmiGroupList_t*       list);
    void DMI_Structure_Free (DmiGroupInfo_t        info);
    void DMI_Structure_Free (DmiRowRequest_t*      request);
    void DMI_Structure_Free (DmiAttributeValues_t* values);
    void DMI_Structure_Free (DmiAttributeData_t*   data);
    void DMI_Structure_Free (DmiAttributeIds_t*    ids);
    void DMI_Structure_Free (DmiRowData_t*         data);
};

```


5.4

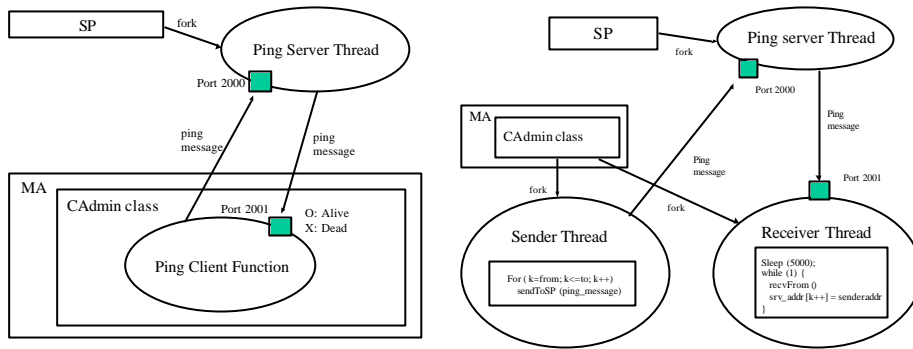
DMI Spec

5.4.1 Auto-discovery Ping

MA 가 SP 가
SP 가
SP ping
port
MA ping 가
port SP port
alive, dead
MA auto-discovery ping
ping
MA
port ping
ping 가 port
ping
[28].

5.4.2 Security

DMI Spec
MI MI
DmiRegister
가



28. PING AUTO-DISCOVERY

```
DmiErrorStatus_t DmiRegister(
  /* [in] */ handle_t IDL_handle,
  /* [out] */ DmiHandle_t __RPC_FAR *handle);
```

```
DmiErrorStatus_t DmiCheck_Register(
  /* [in] */ handle_t IDL_handle,
  /* [in] */ DmiString_t __RPC_FAR *login,
  /* [in] */ DmiString_t __RPC_FAR *password,
  /* [out] */ DmiHandle_t __RPC_FAR *handle);
```

, DmiRegister SP
 (handle) . MA 가 MI
 . SP MI

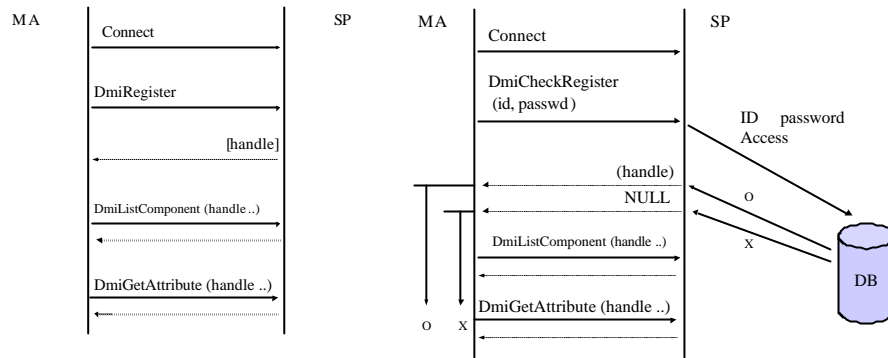
MA 가 SP

. DmiRegister DmiCheck_Register
 . MA SP MA SP
 가

[29].

5.4.3 Event Mail Delivery

Component Event 가 SP EV event
 event DB
 event



29. DMIREGISTER DMICHECKREGISTER

event indication

SP Simple Mail Transfer Protocol (SMTP) 가 (port) 25

```

WSAStartup (0x0101, &WsaData);
sid = socket (AF_INET, SOCK_STREAM, 0);
bind (sid, (LPSOCKADDR)&addr, sizeof (addr));

addrTo.sin_family = AF_INET;
addrTo.sin_port = htons (25); // SMTP reserved port number
addrTo.sin_addr.s_addr = inet_addr (mailServer); // SMTP Server
connect (sid, (LPSOCKADDR)&addrTo, sizeof (addrTo));

nBufLen = sprintf (buf, "%s", "HELO ");
nBufLen += sprintf (buf+nBufLen, "%s%s", host, CRLF); // local IP address
nBufLen += sprintf (buf+nBufLen, "%s", "MAIL FROM:");
nBufLen += sprintf (buf+nBufLen, "%s%s", fromAddr, CRLF); //
nBufLen += sprintf (buf+nBufLen, "%s", "RCPT TO:");
nBufLen += sprintf (buf+nBufLen, "%s%s", toAddr, CRLF); //
nBufLen += sprintf (buf+nBufLen, "%s%s", "DATA", CRLF);

nBufLen += sprintf (buf+nBufLen, "%s", CRLF);
nBufLen += sprintf (buf+nBufLen, "%s", contents); //
nBufLen += sprintf (buf+nBufLen, "%s", CRLF);
nBufLen += sprintf (buf+nBufLen, "%s", ".");
nBufLen += sprintf (buf+nBufLen, "%s", CRLF);

nByteSent = send (sid, buf, nBufLen, 0); //
closesocket (sid);

```

5.4.4 Event

event (component) indication (component 가/
 , group 가/ , language 가/ SP) event
 DB 가

가

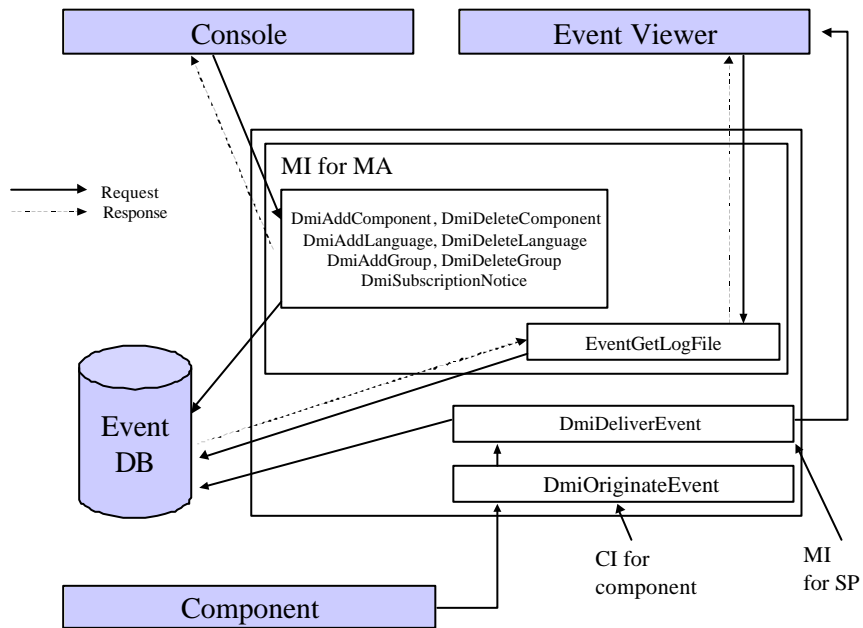
가

(filtering)

EV

event DB

30

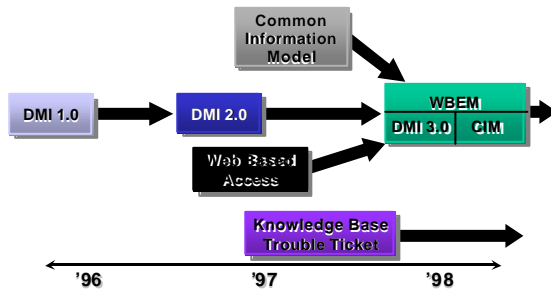


30. EVENT

6.

DMI 2.0 PC
 DMI MIF
 가 , , (CPU, I/O ,)
 windows 가 MS- 가
 PC
 PC NetPC
 가 , SNMP MIB [Bos95]
 SNMP 가

DMI Ver 3.0
 Web-based Enterprise Management (WBEM) [Fre96a, Fre96b]
 (web) 가 가
 [31].



31. DMTF

가

[32]

WBEM 가 ,

Management Object Format (MOF) [Dmt97]

가

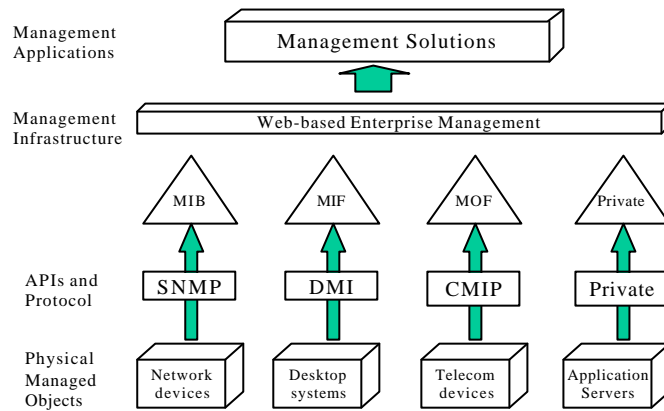
. WBEM

(schema)

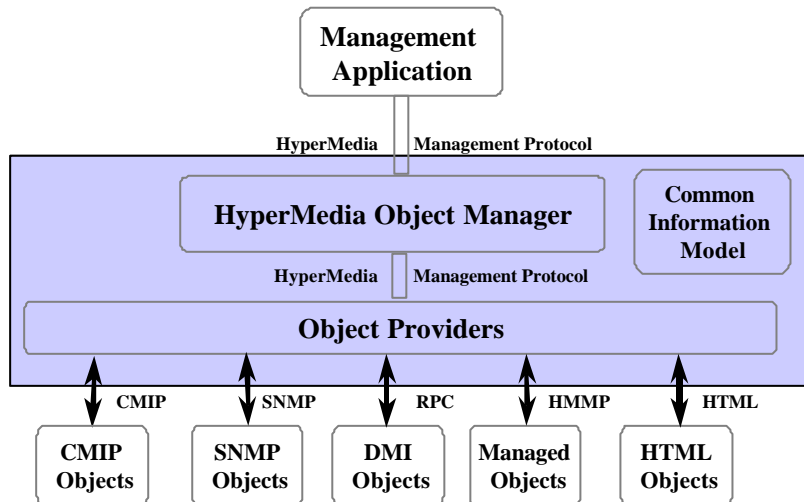
MIF, MIB (Management Information Base) [Wil93], GDMO (Guidelines for

Definition of Managed Objects) [Cct92e]

, [33].



32. 가



33. WBEM

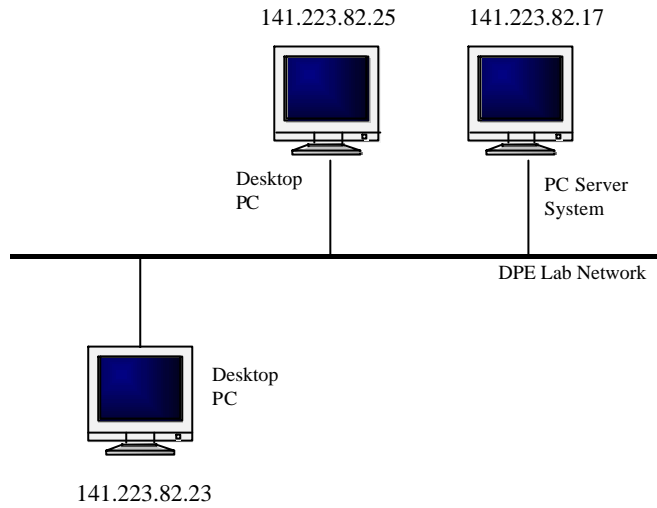
[]

- [Alo96] Alok Sinha, Network Programming in Windows NT, ADDISON WESLEY, 1996.
- [Art95] Arthur Dumas, Programming WinSock, Sams Publishing, 1995.
- [Bos95] S. Bostock, DMTF SNMP to DMI Mapping Standard, Novell Inc, July 1995.
- [Cct92a] CCITT Recommendation X.700, Management Framework Definition for Open Systems Interconnection (OSI) for CCITT Applications, 1992.
- [Cct92b] CCITT Recommendation X.701, Systems management overview, 1992.
- [Cct92c] CCITT Recommendation X.720, Management information model, 1992.
- [Cct92d] CCITT Recommendation X.721, Definition of management information, October 1992.
- [Cct92e] CCITT Recommendation X.722, Guidelines for the definition of managed objects, October 1992.
- [Dav96] David J. Kruglinski, Inside Visual C++, Microsoft Press, 1996.
- [Div96] Divakara K. Udupa, Network Management Systems Essentials, McGraw Hill, 1996.
- [Dmt96a] DMTF, Desktop Management Interface Specification, Version 1.1, 1996.
- [Dmt96b] DMTF, Desktop Management Interface Specification, Version 2.0, 1996.
- [Dmt94a] DMTF, Enabling your product for manageability with MIF files, Ver 1.0, November 1994.
- [Dmt94b] DMTF, LAN Adapter Standard Groups Definition, Ver 1.0, October 1994.
- [Dmt95a] DMTF, PC System Standard MIF Definition, Ver 1.13, March 1995.
- [Dmt95b] DMTF, Software Standard Group Definition Ver 2.0, November 1995.
- [Dmt97] DMTF, Common Information Model (CIM) Specification, Version 1.1, October 1997.
- [Fre96a] WBEM Consortium, Web-based Enterprise Management Proposal: Web-based Management Architecture Hypermedia Management Schema Overview, Revision 0.04, July 1996.
- [Fre96b] WBEM Consortium, Web-based Enterprise Management Proposal: HyperMedia Management Protocol Overview, 1996.
- [Ibm96] IBM, IBM SystemView Agent DMI Programmer's Guide, Second Edition, April 1996.
- [Int96] Intel, Intel LANDesk Software Development Kit, 1996.

- [Jef96] Jeffery Richter, *Advanced Window NT*, Microsoft Press, December 1996.
- [Joh92] John Bloomer, *Power Programming with RPC*, O'Reilly & Associates Inc, 1992.
- [Mic96] Micket Williams and David Hamilton, *Programming Windows NT 4 UNLEASHED*, Sams Publishing, 1996.
- [Mic95] Microsoft, *Visual C++ Tutorials: Development System for Windows 95 and Windows NT*, Microsoft Press, 1995.
- [Mic97a] Microsoft, *Complete documentation for Microsoft Visual C++ version 5.0: MFC Library Reference Part 1, Part 2*, Microsoft Press, 1997.
- [Mic97c] Microsoft, *Complete documentation for Microsoft Visual C++ version 5.0: Runtime Library Reference*, Microsoft Press, 1997.
- [Mic97d] Microsoft, *Complete documentation for Microsoft Visual C++ version 5.0: Language Reference*, Microsoft Press, 1997.
- [Mic97e] Microsoft, *WBEM Schema Overview, Revision 1.0*, 1997.
- [Ral94] Ralph Davis, *Windows NT Network Programming: How to Survive in a 32-bit Networking World*, ADDISON WESLEY, 1994.
- [Ric96] Richard Simon, *Windows 95 WIN32 Programming API Bible Vol.1, Vol2*, WAITE GROUP PRESS, 1996.
- [Slo94] Sloman, *Network and Distributed Systems Management*, ADDISON WESLEY, 1994.
- [Ste96] Steve Holzner, *Advanced Visual C++ 4*, M&T Books, 1996.
- [Sun96] Sun Microsystems, *Solstice Enterprise Agent Development Guide, Revision 50*, November 1996.
- [War93] Ward Rosenberry and Jim Teague, *Distributing Applications Across DCE and Windows NT*, O'Reilly & Associates Inc, 1993.
- [Wil93] William Stallings, *SNMP, SNMPv2, and CMIP, The Practical Guide to Network-Management Standards*, ADDISON WESLEY, April 1993.
- [Wil94] William Stallings, *DATA AND COMPUTER COMMUNICATIONS*, Prentice Hall, 1994.
- [Wil96] William Stallings, *SNMP, SNMPv2, and RMON, Practical Network Management Second Edition*, ADDISON WESLEY, July 1996.

[: PC]

34



34. (TESTBED)

SP CI-code 가 agent PC server 17
 desktop PC 가 25
 , MA 가 console desktop PC

23

? : DPE Lab

? : Windows NT 가 Magic Power Pro 620

Windos 95 가 PC

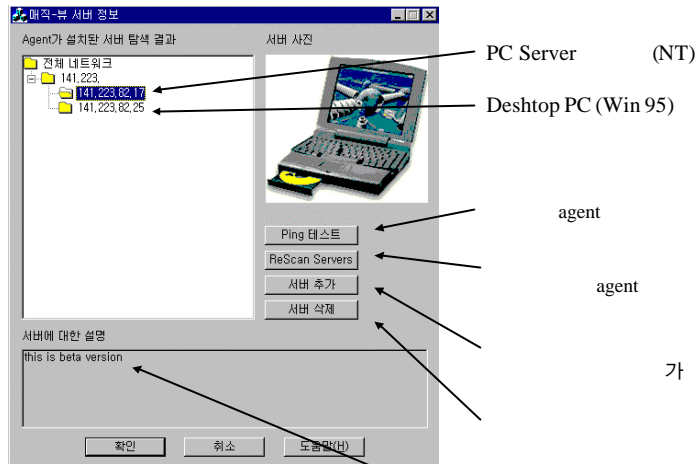
console

agent 가

B class

C class

[35]



agent

agent

가

(,)

35. AUTO-DISCOVERY

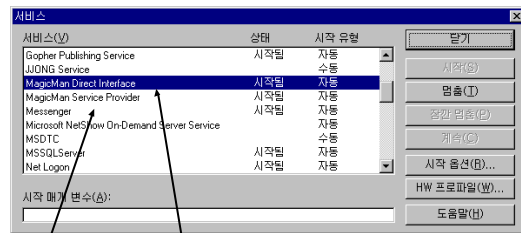
Agent 가

agent

windows

windows

[36]



Service Provider

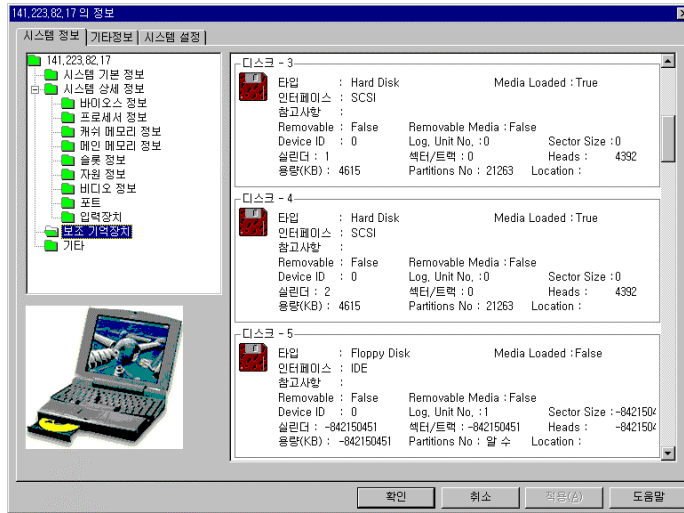
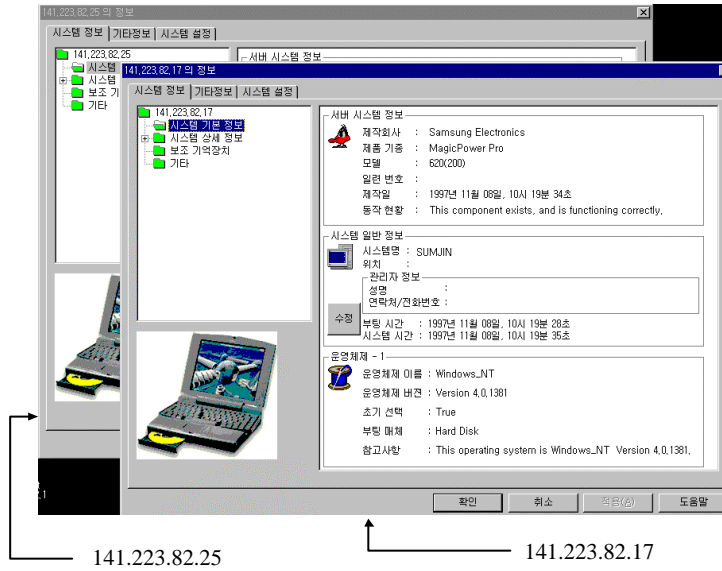
Component Instrumentation code

Windows

36. SP CI-CODE

Console

[37].



가 4

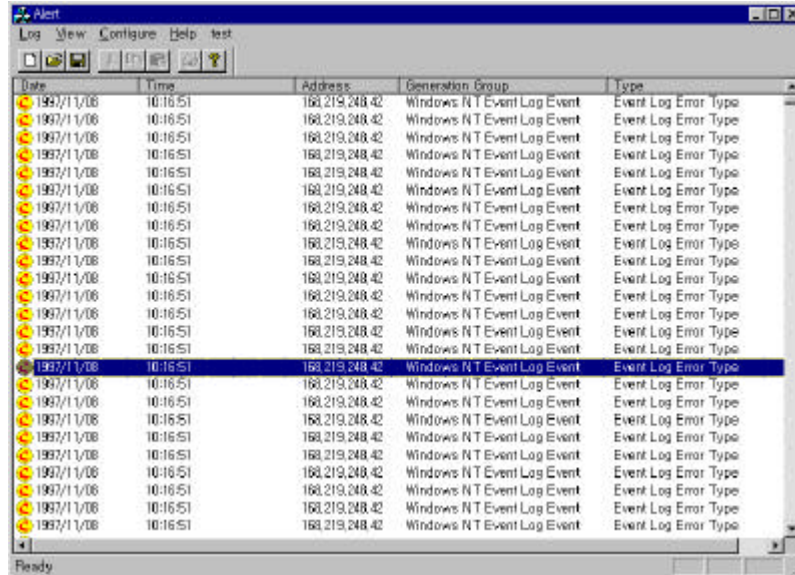
가 1

37.

event가

EV (Event Viewer)

[38]

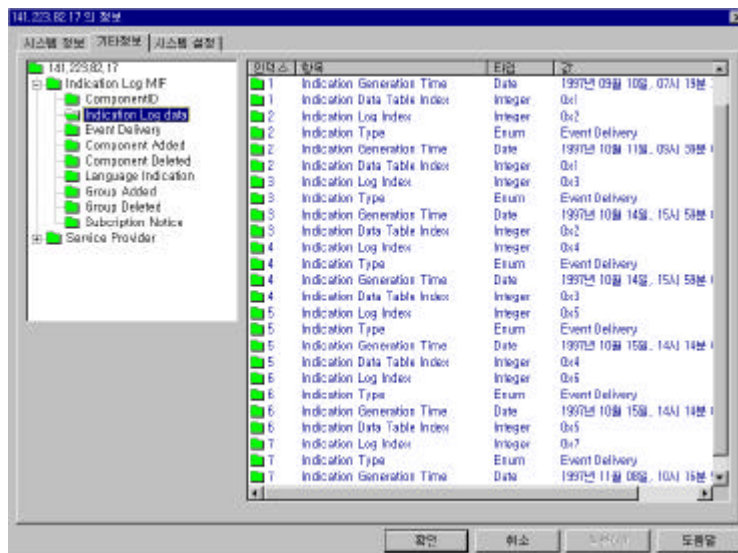


38. EVENT VIEWER (EV)

event indication

console

[39]



39. CONSOLE

EVENT LOG

: (金 鍾瑞)

: 1969 3 15

:

: 4 APT 401 714

1987 – 1993 : (B.S.)

1996 – 1998 : (M.S.)

1993 - : ()

J. W. Hong, J. Y. Kong, T. H. Yun, J. S. Kim, J. T. Park and J. W. Baek, "Web-based Intranet Services and Network Management", IEEE Communications Magazine, October 1997, pp. 100-110 (SCI).