

MCS , Sung-Pil Son, Design and Implementation of a Generic
9625M19 Management Information Repository Service for Network and
System Management,
1998, 58P, Advisor : Won-Ki Hong, Text in Korean.

ABSTRACT

Nowadays, the explosive growth of communication networks and computer systems affects our society heavily. To manage these networks and systems efficiently, automated network and system management systems and tools are needed. For these systems and tools, which are provided by various vendors to exchange and represent information needed to manage networks and systems, a standardized management technique is essential.

In response to these needs, some standardization organizations have published their own standards for network and system management (e.g., IETF SNMP, DMTF DMI, and OSI CMIP). These standards resemble one another in basic management model, which consists of manager and agent, but there are many differences in their data models and languages such as SNMP SMI, DMI MIF, and CMIP GDMO.

If a developer develops a management system based on a particular standard from scratch, it takes much time and cost. If a standard is modified or new standard is published, it is needed to modify many parts of the existing management system or newly develop. Further, multi-standard supporting management system is sometimes needed.

In this thesis, we present the design and implementation of a Generic Management Information Repository Service (GMIRS) as a solution for above problems. GMIRS, which will be used mainly by agents, provides repository service for the management information regardless of management standards and can be extended easily. It takes DMTF Common Information Model (CIM) as data model. The GMIRS prototype implemented here provides repository service only for SNMPv1 and DMI 2.0, but can be easily extended for other management standards.

1.	1
2.	3
2.1	Simple Network Management Protocol (SNMP).....	3
2.1.1	3
2.1.2	3
2.1.3	Structure of Management Information (SMI)	5
2.2	Desktop Management Interface (DMI)	7
2.2.1	7
2.2.2	7
2.2.3	Management Information Format (MIF)	10
2.3	Common Information Model (CIM)	12
2.3.1	12
2.3.2	13
2.3.3	Managed Object Format (MOF).....	16
3.	GMIRS	17
3.1	GMIRS	17
3.2	GMIRS	18
3.3	SNMP SMI Translator	22
3.4	DMI MIF Translator.....	25
3.5	SNMP Proxy.....	29
3.6	DMI Proxy	30
4.	GMIRS Prototype	32
4.1	SNMP SMI Translator	33
4.2	DMI MIF Translator.....	40
4.3	SNMP Proxy.....	45
4.4	DMI Proxy	46
4.5	GMIRS	47

5.	49
	52
	54
A.	SMI Technique Mapping Schema	54
B.	MIF Technique Mapping Schema	56

1	SNMP	4
2	SNMPv1 MIB	6
3	DMI	9
4	DMI	10
5	CIM 4 가	12
6	CIM meta schema structure	15
7	18
8	GMIRS	20
9	GMIRS	21
10	GMIRS SMI Technique Mapping	22
11	MIF Technique Mapping	25
12	GMIRS MIF Technique Mapping	27
13	DMI Management Interface	30
14	GMIRS Prototype	32
15	MagicMan	47
16	WBEM	50

1.

가

, / (Client/Server) [31] 가

가 World Wide

Web (WWW) 가가

가 (local-area

network, LAN) (wide-area network, WAN),

(router), (bridge), PC, (workstation),

(server) [9]. 가

(vendor)

(standard)

Internet Engineering Task Force (IETF) [25] Simple Network Management Protocol (SNMP) [1-3, 13-22], Desktop Management Task Force (DMTF) [24] Desktop Management Interface (DMI) [4], Open Systems Interconnection (OSI) Common Management Information Protocol (CMIP) [2, 33] (manager)

(agent) (management

information) (language)

SNMP

Structure of Management Information (SMI), DMI Management

Information Format (MIF), CMIP Guidelines for Definition of Managed Objects (GDMO)

Web-Based Enterprise Management (WBEM) [6, 7] SNMP, DMI, CMIP

WBEM IETF HyperMedia Management Protocol (HMMP) [32], DMTF Common Information Model (CIM) [5]

가 , 가 .

(Generic Management Information Repository Service, GMIRS)

. GMIRS WBEM CIM

. GMIRS

GMIRS

GMIRS prototype 가

SNMPv1 DMI 2.0

. 2

SNMP, DMI, CIM

, 3 4

GMIRS GMIRS

prototype

. 5

2.

가 SNMP, DMI, CIM

2.1 Simple Network Management Protocol (SNMP)

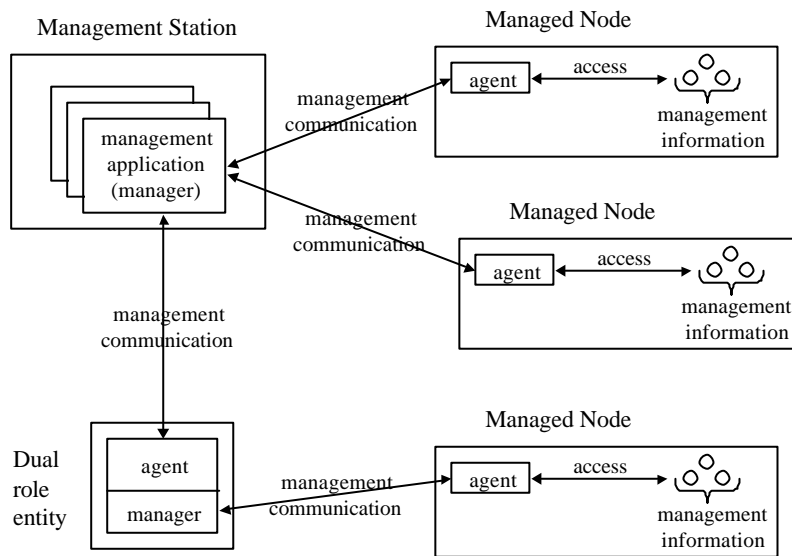
2.1.1

SNMP [1-3, 13-22] (standards)
Internet Engineering Task Force (IETF)
TCP/IP (router) (host)
, SNMP
SNMP
(vendor)
SNMP
가
SNMPv2, SNMPv1
. SNMPv2 SNMPv1

2.1.2

(managed network) SNMP
, 1

~~del~~ (agent) (processing entity) 가
 (managed node)
~~del~~ (management application) (manager)
 가
 (management station)
~~del~~ , - (dual-role entity)
~~del~~ , ,
 (management information)
~~del~~ 가
 (management protocol)



1

SNMP

SNMP Request for Comments (RFC) [13-22]
4 가

~~SNMP~~
~~SNMP~~
~~SNMP~~ (core set)
~~SNMP~~

SNMP (processing entity)
(management communication) (format) (meaning)
SNMP SNMPv1 SNMPv2, 가
SNMP GET, GETNEXT, GETBULK
(retrieving), SET (modifying), TRAP, INFORM
(event-reporting) 3 가 . SNMP
Structure of Management Information (SMI)
SNMPv1 SNMPv2 SMI ,
RFC [13-22]

2.1.3 Structure of Management Information (SMI)

SNMP 가 Management Information Base
(MIB) . SMI MIB
Abstract Syntax Notation One (ASN.1) [26]
(subset) SNMP 가
SNMP object identifier (OID)
OID
OID (ordered)

sequence) , OID
SMI

SNMPv1

```
<moduleName>  
DEFINITIONS ::= BEGIN  
  
<importedItems>  
  
<definitions>  
  
END
```

2 SNMPv1 MIB

2 SNMPv1 MIB

(outer wrapper) MIB
<importedItems>

SNMPv1 SMI INTEGER, Gauge,
Counter, TimeTicks, OCTET STRING, OBJECT IDENTIFIER, IpAddress, NetworkAddress,
Opaque , textual convention
MIB ,
OBJECT-TYPE (construct) , (row), (column),
OBJECT-TYPE OBJECT-IDENTITY, SEQUENCE,
TRAP-TYPE

2.2 Desktop Management Interface (DMI)

2.2.1

DMI Desktop Management Task Force (DMTF)
· SNMP가 (router), (hub), (bridge)
, DMI PC
· 1994 DMI (specification)
DMI 2.0 [4] · DMI 2.0 DMI 1.X
Remote Procedure Call (RPC) 가
· DMI 2.0

2.2.2

DMI 4가 가 .
~~///~~ (format)
~~///~~ Service Provider Entity
~~///~~ API (Application Programming Interface) , Service
Provider (management application)
Service Provider (component)가
~~///~~
DMI Management Information Format (MIF)
가
MIF 가 , , ,
(firmware) (entity)
가 , MIF Service Provider MIF
가 .

DMI Service Provider Component Instrumentation 가
 가 Service Provider API for Components .
 Component Instrumentation code DMI Service Provider 가
 가 Component Provider API .
 Component Interface (CI) , MIF (vendor)가

DMI Service Provider 가
 가 Service Provider API for Management .
 DMI Service Provider 가
 가 Management Provider API . Management
 Interface (MI) , Management

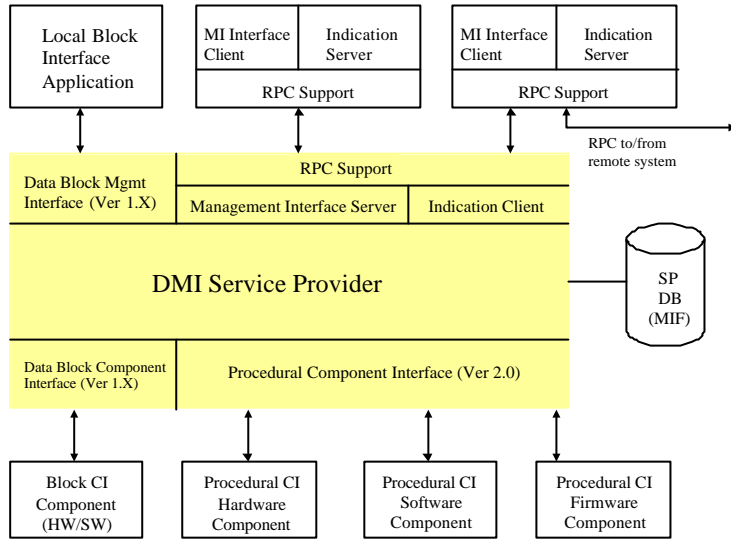
DMI Service Provider , MI
 CI . DMI 2.0 MI
 Remote Procedure Call (RPC) 가 DMI가
 RPC Distributed Computing Environment (DCE) RPC [10], Open
 Network Computing (ONC) RPC [11], Texas Instruments (TI) RPC 가 CI
 (local interface) .

3 DMI (functional
 block diagram) . MIF
 , DMI Service Provider
 . - (command-line)
 (Graphical User Interface, GUI) 가

, DMI Service Provider, Component Instrumentation
 3 (many-to-one-to-many)

가

DMI Service Provider



3 DMI

(control flow)

DMI Service Provider (unsolicited report) , Component Instrumentation (indication)

Get, Set, List 가

. Get Set

. List MIF

(meta) . List attribute

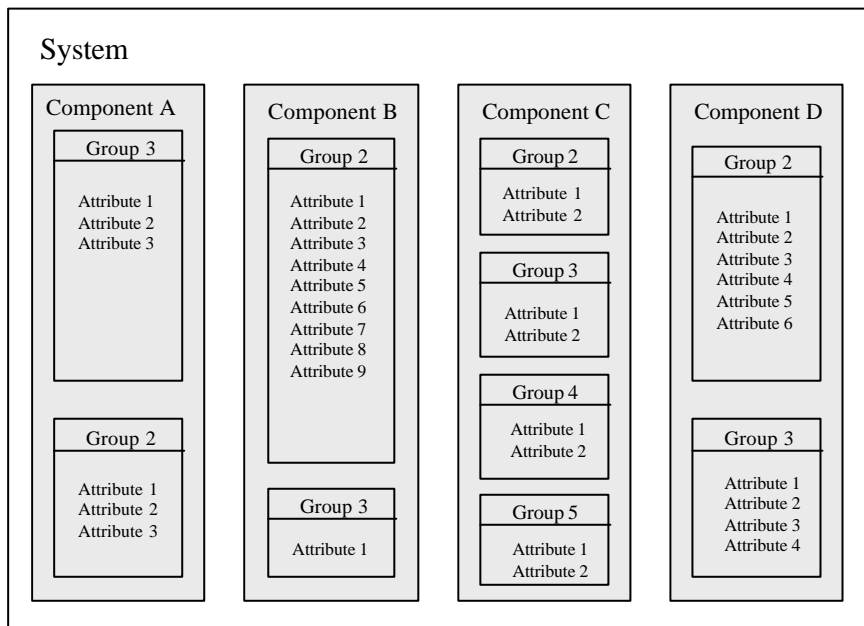
, MIF (semantic information) 가

Component Instrumentation Code MIF MIF

가 . DMI 2.0 [4]

2.2.3 Management Information Format (MIF)

DMI component
 가 attribute 가
 attribute
 가 group group (scalar)
 (instance) 가 group table
 , table (row,) (key)
 attribute component 가
 DMI 4



4 DMI

MIF SNMP SMI
 integer, integer64, gauge, counter, counter64, string,

displaystring, octetstring, date .

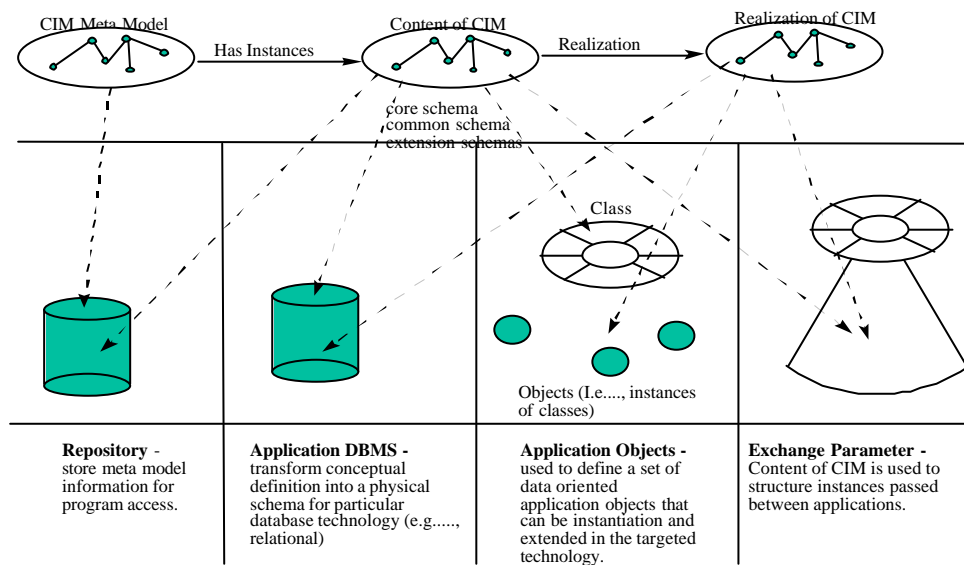
MIF component, group, attribute, table, enum, path
 . 'start' 'end'
 . group ,
class (statement) , table template
 . component, group, attribute Id 가 ,
(unique) . attribute
component Id, group Id, attribute Id .

4.2 MIF 가 .

2.3 Common Information Model (CIM)

2.3.1

1996 9 , DMTF (Web-based management) ,
 Common Information Model (CIM)
 CIM (subcommittee)가 . CIM HyperMedia Management Schema
 (HMMS) [27] , SNMP, DMI 2.0, CMIP
 Common Object Request Broker Architecture (CORBA) [28], Common Object
 Model (COM) [29] (object-based execution model)
 Java Management API (JMAPI) [30]
 . CIM 1.0 1997 4 , CIM 1.1
 [5]가 1997 9 .



5 CIM

4 가

CIM

(conceptual model) ,

5 CIM

4 가

. CIM 1.1 (repository)

2.3.2

(definitional)

(structural)

가 가 (import)

가 (export) 가

가 3 가 (mapping)

~~///~~ Technique Mapping

~~///~~ Recast Mapping

~~///~~ Domain Mapping

Technique Mapping SMI, MIF, GDMO

(meta constructs) CIM

Recast Mapping (source meta constructs)

(target meta constructs)

CIM

Domain Mapping 가 CIM

core common model, extension sub-schemas

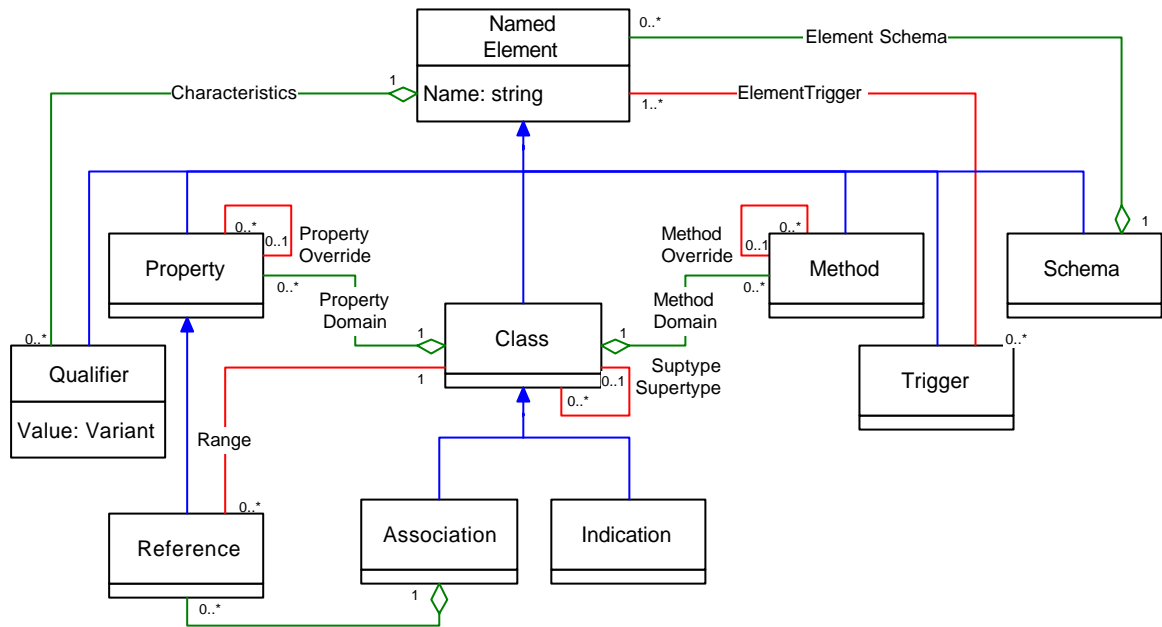
(meta-to-meta mapping) (content-to-

content mapping) . Core model, Common model, Extension schemas

~~del~~ Core model – (notion) 가
 (capture)
~~del~~ Common model – ,
 가 , ,
 , , ,
 (specific) , (schema)
 (technology specific schema)
 Core model Common model CIM schema
~~del~~ Extension schemas – Common model (technology-
 specific extension) (UNIX, Microsoft
 Windows)

6 CIM (meta schema structure) Unified
 Modeling Language (UML) [12]

~~del~~ schema (single owner) 가 class
~~del~~ class , property method
 (instance)
~~del~~ property class
~~del~~ method (signature, , method name, return type, parameters)
 , concrete class (implementation)
~~del~~ trigger class , , , property ,
 (state change) (recognition)



6 CIM meta schema structure

~~indication~~ trigger (side effect)
~~association~~ reference class ,
~~reference~~ association 가
~~qualifier~~ named element
 , property class (key) qualifier
 . qualifier (limited and
 controlled fashion)

2.3.3. Managed Object Format (MOF)

CIM Managed Object Format (MOF)
 Interface Definition Language (IDL) [10]

MOF (syntax) ,
 . MOF class, association, property, reference,
 method, instance qualifier . MOF (specification)

(compiler directive) .

(managed object) (namespace)
 class class instance 가
 , class
 instance . reference

(child namespace) 가 ,

가 , (sibling namespace)
 (namespace path)

qualifier property, class, instance - (meta-data)
 . qualifier '['
 ']'

instance
 , 'key' qualifier property instance

3. GMIRS

Generic Management Information Repository Service (GMIRS)

3.1 GMIRS

GMIRS

가

(agent)

, GMIRS

GMIRS

가

가

가

~~del~~

repository)

(data

~~del~~

~~del~~

~~del~~

~~del~~

~~del~~

~~del~~

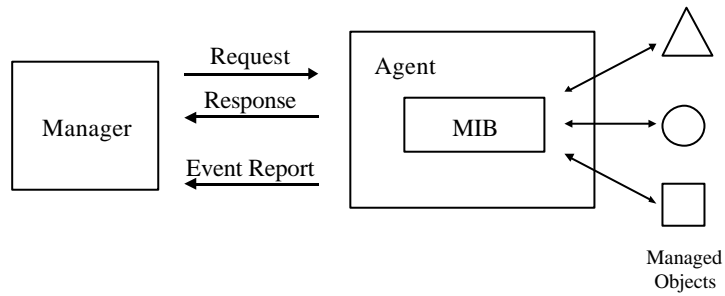
~~del~~

~~del~~

~~del~~

3.2 GMIRS

GMIRS (agent, DMI Service Provider) 가
 Management Information Base, MIB ()
 가



GMIRS (7 MIB) (set)

가

. GMIRS
가

,

.
, 가

,

GMIRS

가

~~가~~

~~가~~

(translator)

(proxy)

GMIRS

Common Information Model (CIM)

. CIM

MOF

. CIM

가

SNMP SMI DMI MIF

MOF

CIM

CIM

(CIM Object Manager, CIMOM)

CIMOM

(CIMOM DB)

API

CIMOM

GMIRS

8

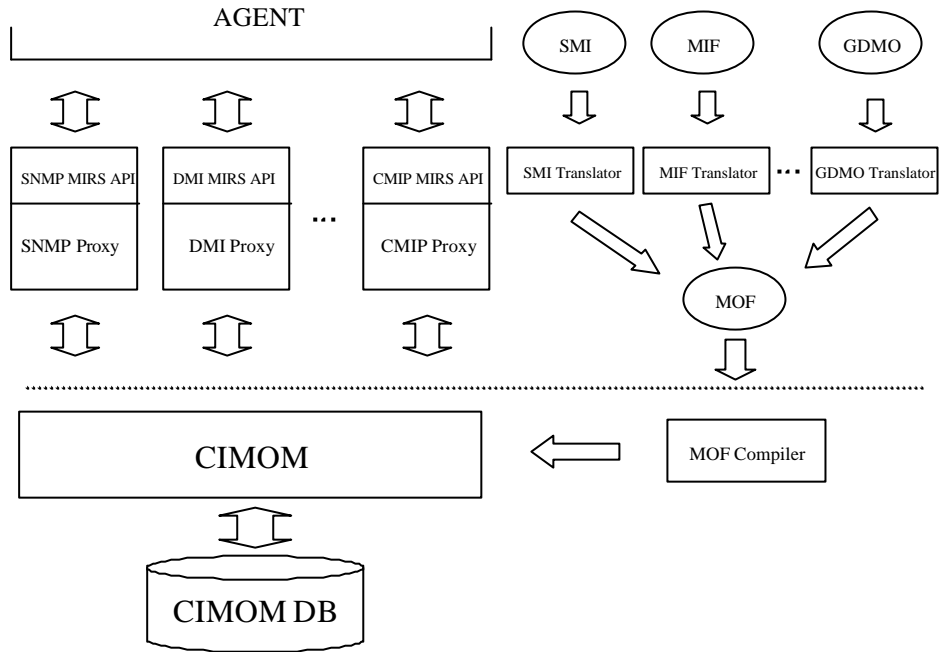
GMIRS

CIM

CIMOM, CIMOM DB

MOF

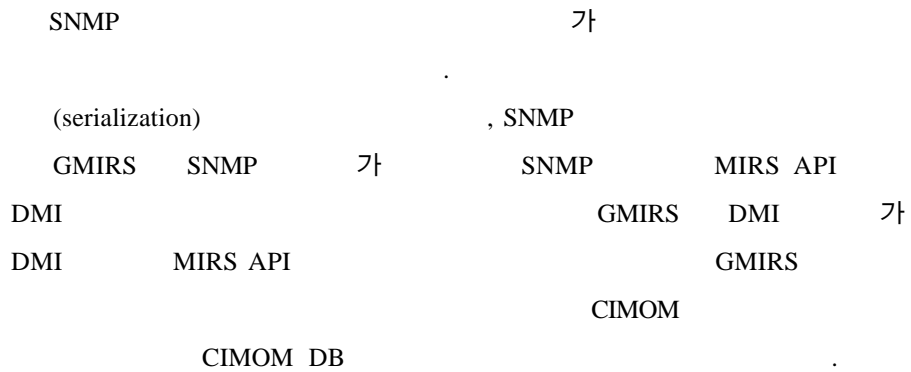
,

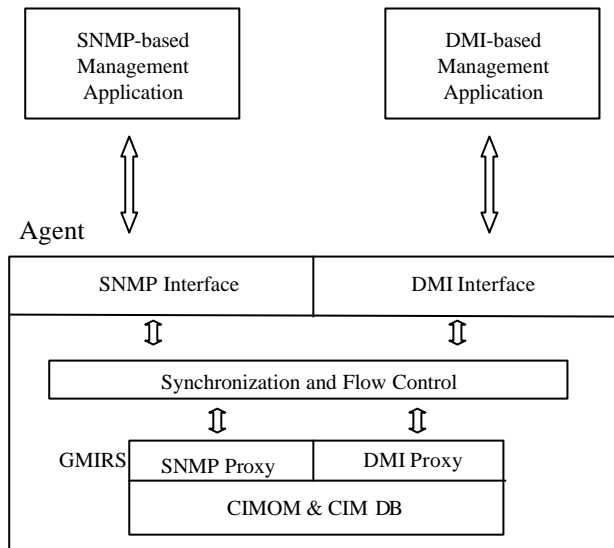


8 GMIRS

9 GMIRS 가

. DMI





9 GMIRS

GMIRS

가 MIRS API

SNMP

DMI

GMIRS prototype

가

SNMPv1 DMI 2.0

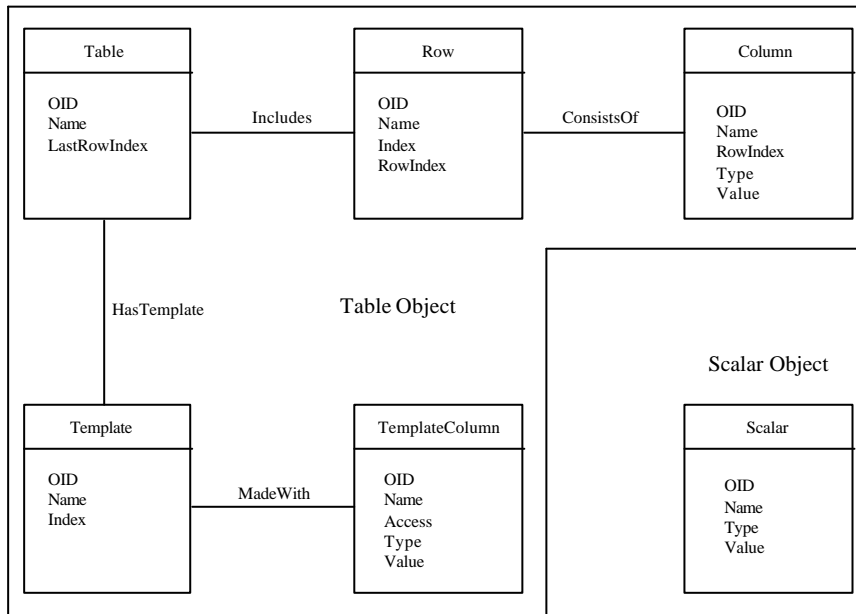
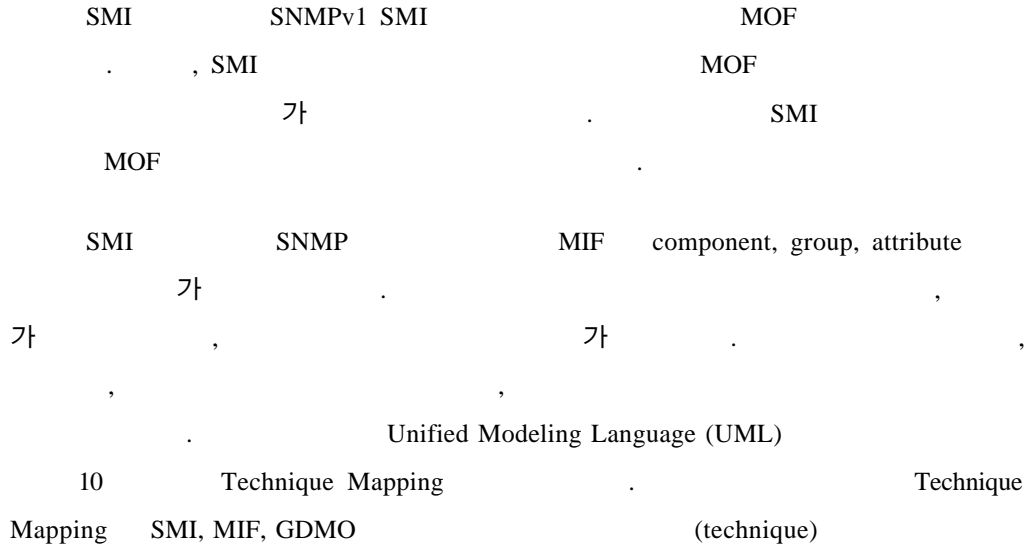
, CIMOM MOF

Microsoft WBEM SDK Beta 2

SNMPv1

DMI 2.0

3.3 SNMP SMI Translator



10 GMIRS SMI Technique Mapping

(meta constructs)	CIM	Technique Mapping
10		
Table, Template, TemplateColumn, Scalar		
OBJECT-TYPE		
		가
	10	class association
	class	'SNMP_' SNMP class
	(association class .)	class
SNMP_Table	-	가
SNMP_Template	-	가
		가 template
SNMP_TemplateColumn	-	가 가
		template
SNMP_Row	-	가
SNMP_Column	-	가
SNMP_Scalar	-	가
	class	association
SNMP_HasTemplate	- SNMP_Table	SNMP_Template
SNMP_MadeWith	- SNMP_Template	SNMP_TemplateColumn
SNMP_Includes	- SNMP_Table	SNMP_Row
SNMP_ConsistsOf	- SNMP_Row	SNMP_Column

class SNMP

SmpClass . A

[abstract]
class SmpClass
{
};

SMI 가 MOF class instance

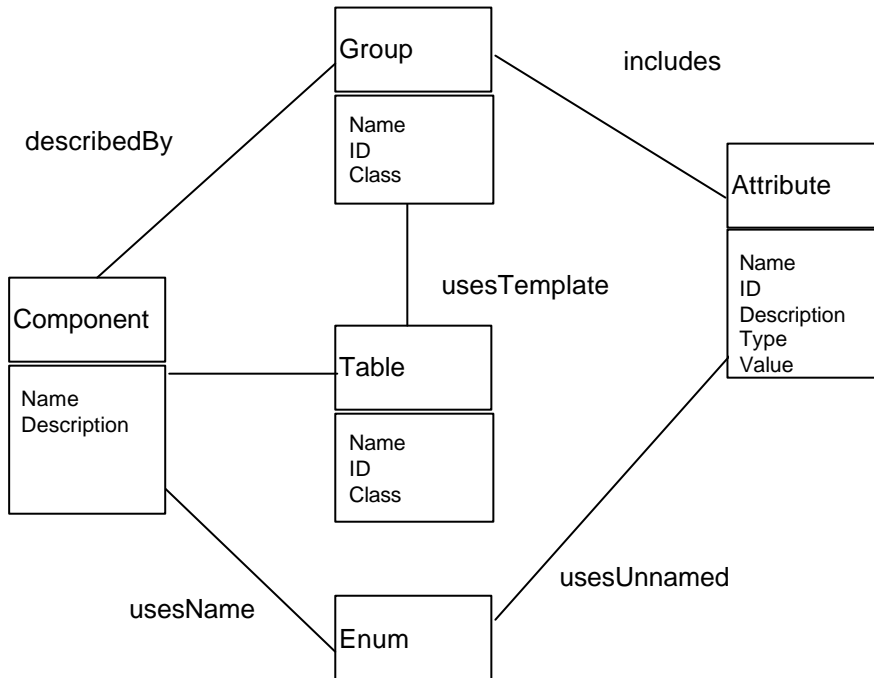
instance (namespace) “\\.\Root\SNMPv1” . SNMP class

SNMP (proxy) class instance

4.1 SMI 가 MOF

3.4 DMI MIF Translator

MIF MIF MOF
 , MIF MOF MOF
 , 가 MIF
 MOF
 CIM 11 MIF Technique Mapping



11 MIF Technique Mapping

Technique Mapping MIF
 . MIF Technique Mapping class
 . MOF class instance
 class 'DMI_' DMI

class

class	MIF	
del DMI_Component	- component	가 .
del DMI_Group	- group	가 .
del DMI_Attribute	- attribute	가 .
del DMI_Enum	- enum	가 .
del DMI_Path	- path	가 .
del DMI_Table	- table	가 .
	class	.
del DMI_EnumContent	- DMI_Enum	embedded class .
del DMI_TableRow	- DMI_Table	가 TableRow 가 .
del DMI_TableColumn	- DMI_TableRow	embedded class .
	association	.
del DMI_DescribedBy	- DMI_Component DMI_Table	DMI_Group .
del DMI_IncludesAttrib	- DMI_Group	DMI_Attribute .
del DMI_UsesTemplate	- DMI_Table DMI_Group	가 template .
del DMI_UsesName	- DMI_Component	DMI_Enum .
del DMI_UsesUnnamed	- DMI_Attribute	DMI_Enum .
del DMI_IncludesRow	- DMI_Table	DMI_TableRow .

DMI_IncludesPath - DMI_Component DMI_Path

class DMI

DmiClass

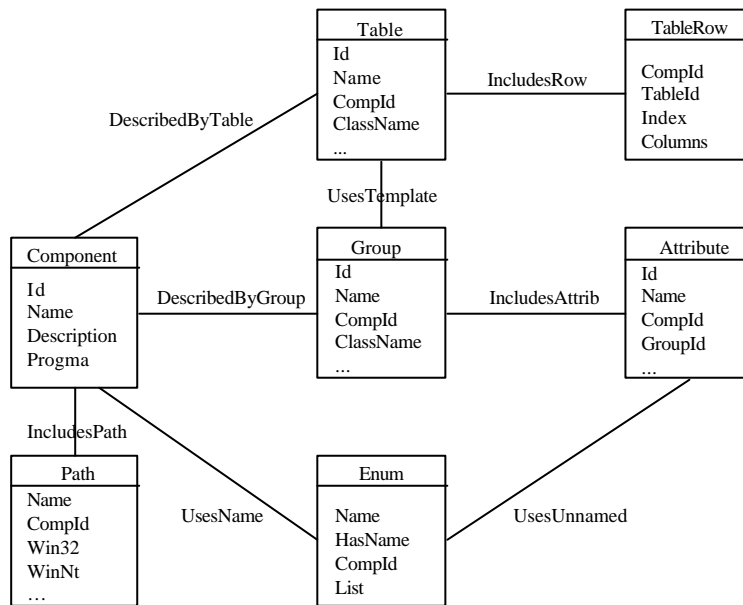
```
[abstract]
class DmiClass
{
};
```

UML

12

class

B



12 GMIRS MIF Technique Mapping

MIF component ID

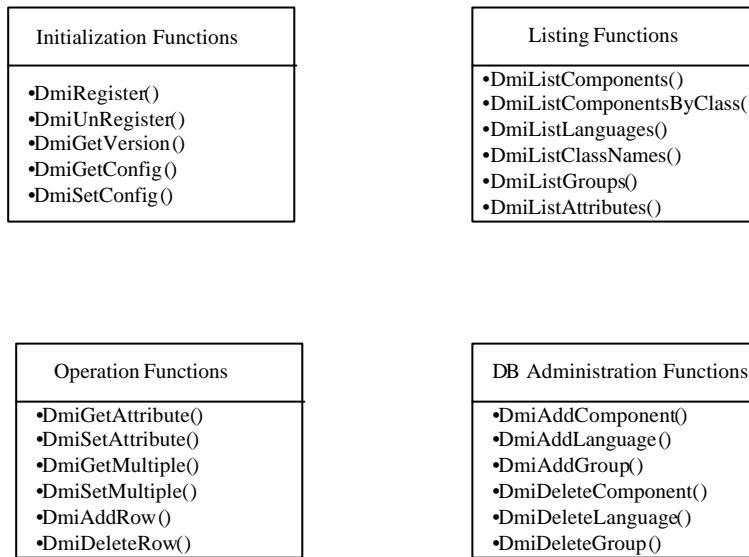
. DMI MIF (namespace)
“\\.\Root\DMIF” 가 . DMI (proxy)
class instance .
4.2 MIF MOF

3.5 SNMP Proxy

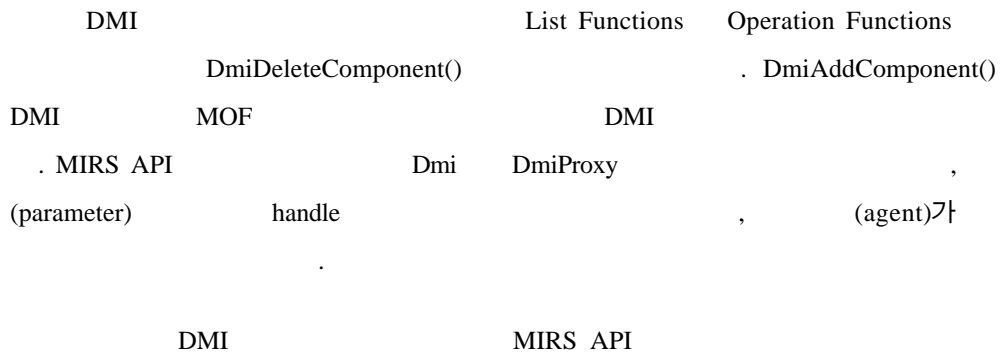
SNMPv1 GET, GETNEXT, SET, TRAP
 . SNMP (proxy) TRAP 3 가
 API . SNMP GET, GETNEXT, SET
 . GET
 .
 OID 가 . GETNEXT OID
 (lexicographical order) OID 가
 , 가 . SET
 ,
 .
 SNMP MIRS API .
 .
~~SnmpProxyError_t SnmpGet (SnmpOid_t oid, SnmpValue_t **value)~~
~~SnmpProxyError_t SnmpGetNext (SnmpOid_t oid, SnmpValue_t **value)~~
~~SnmpProxyError_t SnmpSet (SnmpOid_t oid, SnmpValue_t *value)~~
 GET, GETNEXT, SET API
 . SNMP
 .
 SET 가 가
 . (All or Nothing)
 .
 SNMP Proxy MIRS API 가
 oid SNMP

3.6 DMI Proxy

13 DMI Service Provider 가
(Management Application) Management Interface



13 DMI Management Interface



~~del~~ DmiProxyListComponents()
 - component .

~~del~~ DmiProxyListComponentsByClass()
 - class name 가 group row 가 component .

~~del~~ DmiProxyListClassNames()
 - component class name group Id .

~~del~~ DmiProxyListGroups()
 - component group .

~~del~~ DmiProxyListAttributes()
 - component group attribute .

~~del~~ DmiProxyGetAttribute()
 - attribute () .

~~del~~ DmiProxySetAttribute()
 - attribute () .

~~del~~ DmiProxyGetMultiple()
 - () attribute .

~~del~~ DmiProxySetMultiple()
 - () attribute .

~~del~~ DmiProxyAddRow()
 - 가 .

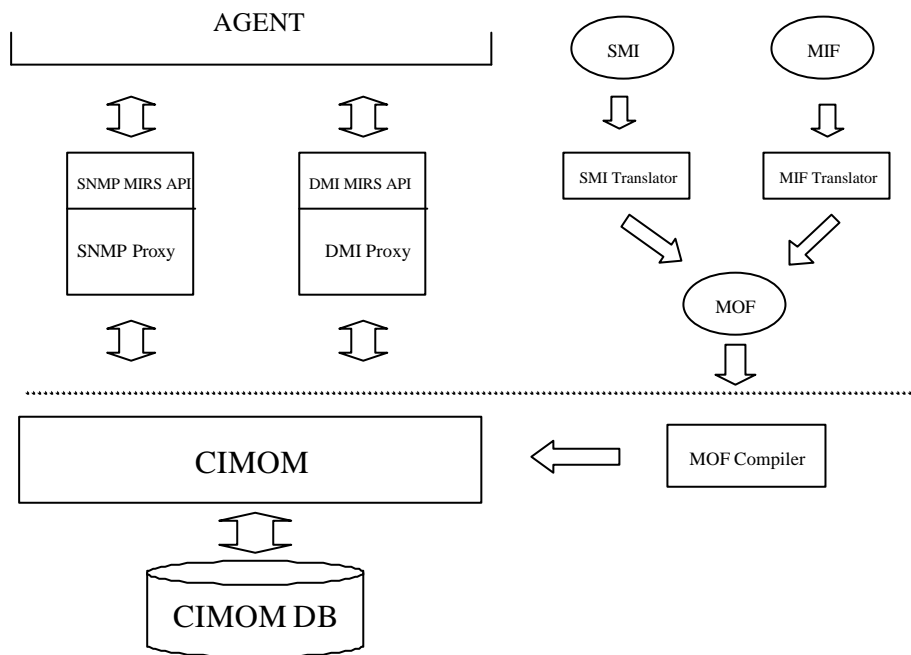
~~del~~ DmiProxyDeleteRow()
 - .

~~del~~ DmiProxyDeleteComponent()
 - component .

4. GMIRS Prototype

GMIRS prototype C C++
 Microsoft Visual C++ 5.0
 Windows NT , CIMOM MOF
 Beta 2 Microsoft WBEM SDK

GMIRS prototype 가
 SNMPv1 DMI 2.0 , 14



14 GMIRS Prototype

4.1 SNMP SMI Translator

```
SMI . SMI
      .
      MOF
      .
      MIB textual convention OBJECT-TYPE,
SEQUENCE, OBJECT-IDENTIFIER
      .
SMI MIB
```

```
typedef struct TextConList *TextConListP_t;
typedef struct TextConList // textual convention
{
    char *name;
    char *type;

    TextConListP_t next;
}TextConList_t;

typedef struct Sequence *SequenceP_t;
typedef struct Sequence
{
    char *name;
    char *type;

    SequenceP_t *next;
}Sequence_t;

typedef struct SeqList *SeqListP_t;
typedef struct SeqList // SEQUENCE
{
    char *name;
    Sequence_t *sequence;

    SeqListP_t next;
}SeqList_t;

typedef struct ObjectIdenList *ObjectIdenListP_t;
```

```

typedef struct ObjectIdentList          // OBJECT-IDENTIFIER
{
    char *name;
    char *oid;

    ObjectIdentListP_t next;
}ObjectIdentList_t;

typedef struct Index *IndexP_t;
typedef struct Index
{
    char *name;

    IndexP_t next;
}Index_t;

typedef struct ObjectTypeList *ObjectTypeList_t;
typedef struct ObjectTypeList          // OBJECT-TYPE
{
    char *name;
    char *syntax;
    char *access;
    char *status;
    char *description;
    char *reference;
    Index_t *index;
    char *oid;
    int checked;

    ObjectTypeP_t next;
}ObjectTypeList_t;

typedef struct MibModule                // mib module
{
    char *name;
    textConList_t *textConList;
    seqList_t *seqList;
    objectIdentList_t *objectIdentList;
    objectTypeList_t *objectTypeList;
}MibModule_t;

```

SMI

MOF

	SMI		MOF
		SMI	
MIB			가 가
OBJECT-TYPE			(row) ,
SEQUENCE		(sequence)	가
		SNMP	
가			

MacAddress ::= OCTET STRING (SIZE (6))

```
dot1dTpFdbTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot1dTpFdbEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A table that contains information about unicast ..."
    ::= { dot1dTp 3 }
```

```
dot1dTpFdbEntry OBJECT-TYPE
    SYNTAX Dot1dTpFdbEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about a specific unicast MAC address ..."
    INDEX { dot1dTpFdbAddress }
    ::= { dot1dTpFdbTable 1 }
```

```
Dot1dTpFdbEntry ::=
    SEQUENCE {
        dot1dTpFdbAddress
        MacAddress,
        dot1dTpFdbPort
        INTEGER,
        dot1dTpFdbStatus
        INTEGER
    }
```

```
dot1dTpFdbAddress OBJECT-TYPE
    SYNTAX MacAddress
```

ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "A unicast MAC address for which the bridge has ..."
 REFERENCE
 "P802.1d/D9, July 14, 1989: Section 3.9.1, 3.9.2"
 ::= { dot1dTpFdbEntry 1 }

dot1dTpFdbPort OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Either the value '0', or the port number of the port on which a frame ..."
 ::= { dot1dTpFdbEntry 2 }

dot1dTpFdbStatus OBJECT-TYPE
 SYNTAX INTEGER { other(1), invalid(2), learned(3), self(4), mgmt(5) }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The status of this entry. The meanings of the values are: ..."
 ::= { dot1dTpFdbEntry 3 }

(bridge)

RFC1286-MIB

[23] "The Forwarding Database for Transparent Bridges"

DESCRIPTION

"MacAddress ::= OCTET STRING (SIZE (6))" textual convention SMI

가

Type property

, MOF

3.3

class (

A) instance

. instance OID

property

OID 가 가 , Name property

(identifier)가

. SNMP

SNMP_Column

SNMP_Scalar

instance

(value) string

가

, Type property 가

class

```

        , Type property
        string

        dot1dTp { dot1dBridge 4 } dot1dBridge { mib-2 17 } , mib-2
1.3.6.1.2.1 dot1dTp OID 1.3.6.1.2.1.17.4
instance OID property OID

'INDEX' ( dot1dTpFdbAddress)
(key) , OID SNMP_Template Index property

ACCESS SNMP_TemplateColumn Access
property , 'read' 'write' qualifier Value property
, 가
, , DESCRIPTION instance
'description' qualifier

```

```

[description ("A table that contains information about unicast ...")]
instance of SNMP_Table
{
    OID = "1.3.6.1.2.1.17.4.3";
    Name = "dot1dTpFdbTable";
    LastIndex = 0;
};

```

```

[description ("Information about a specific unicast MAC address ...")]
instance of SNMP_Template
{
    OID = "1.3.6.1.2.1.17.4.3.1";
    Name = "dot1dTpFdbEntry";
    Index = "1.3.6.1.2.1.17.4.3.1.1";
};

```

```

[description ("A unicast MAC address for which the bridge has ...")]

```



```
instance of SNMP_TemplateColumn
{
    OID = "1.3.6.1.2.1.17.4.3.1.1";
    Name = "dot1dTpFdbAddress";
    Access = "read";
    Type = "OCTET STRING (SIZE (6))";
    Value = "";
};
```

[description ("Either the value '0', or the port number of the port on which a frame ...")]

```
instance of SNMP_TemplateColumn
{
    OID = "1.3.6.1.2.1.17.4.3.1.2";
    Name = "dot1dTpFdbPort";
    Access = "read";
    Type = "INTEGER";
    Value = "";
};
```

[description ("The status of this entry. The meanings of the values are: ...")]

```
instance of SNMP_TemplateColumn
{
    OID = "1.3.6.1.2.1.17.4.3.1.3";
    Name = "dot1dTpFdbStatus";
    Access = "read";
    Type = "INTEGER";
    Value = "";
};
```

```
instance of SNMP_HasTemplate
{
    TableOID = "1.3.6.1.2.1.17.4.3";
    Table = "\\\\.\\Root\\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3";
    Template = "\\\\.\\Root\\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3.1";
};
```

```
instance of SNMP_HasTemplate
{
    TableOID = "1.3.6.1.2.1.17.4.3";
    TemplateOID = "1.3.6.1.2.1.17.4.3";
    ColumnIndex = 1;
    Template = "\\\\.\\Root\\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3.1";
    TemplateColumn = "\\\\.\\Root\\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3.1.1";
};
```

```
instance of SNMP_HasTemplate
{
    TableOID = "1.3.6.1.2.1.17.4.3";
    TemplateOID = "1.3.6.1.2.1.17.4.3";
    ColumnIndex = 2;
    Template = "\\.\Root\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3.1";
    TemplateColumn = "\\.\Root\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3.1.2";
};
```

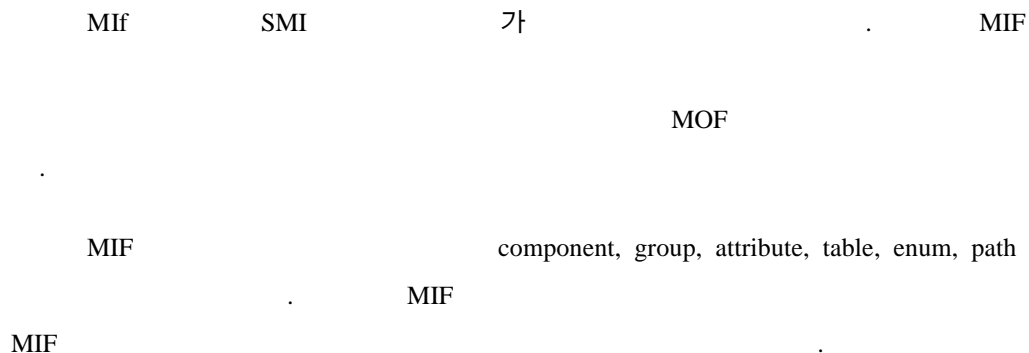
```
instance of SNMP_HasTemplate
{
    TableOID = "1.3.6.1.2.1.17.4.3";
    TemplateOID = "1.3.6.1.2.1.17.4.3";
    ColumnIndex = 3;
    Template = "\\.\Root\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3.1";
    TemplateColumn = "\\.\Root\SNMPv1:OID=" 1.3.6.1.2.1.17.4.3.1.3";
};
```

SNMP_Row, SNMP_Column, SNMP_Includes, SNMP_ConsistsOf instance
가 .

SNMP SMI - (command-line) ,

smi2mof <source smi> <target mof>

4.2 DMI MIF Translator



```
class Enum
{
    public:
        char Name[MAX_NAME];
        char Type[MAX_TYPE];
        int count;
        int Value[MAX_ENUM];
        char Symbol[MAX_ENUM][MAX_SYMBOL];
        Enum* Penum;
};
```

```
class Attribute
{
    public:
        char Name[MAX_NAME];
        unsigned int ID;
        char Description[MAX_DIS];
        char Type[MAX_TYPE];
        char Access[MAX_ACCESS];
        char Pragma[MAX_PRAGMA];
        char Storage[MAX_STORAGE];
        char Value[MAX_VALUE];
        Enum *Penum;
        Attribute *Pattribute;
};
```

```

class Group
{
    public:
        char Name[MAX_NAME];
        char Class[MAX_CLASS];
        unsigned int ID;
        char Description[MAX_DIS];
        int Key[MAX_KEY];
        int count;
        char Pragma[MAX_PRAGMA];
        Attribute *Pattribute;
        Group *Pgroup;
};

class Table
{
    public:
        char Name[MAX_NAME];
        char Class[MAX_DIS];
        unsigned int ID;
        char table[MAX_ROW][MAX_CUL][MAX_ITEM];
        int row, column;
        Table* Ptable;
};

class Path
{
    public:
        char Name[MAX_NAME];
        char Dos[MAX_PATHS];
        char Macos[MAX_PATHS];
        char Os2[MAX_PATHS];
        char Unix[MAX_PATHS];
        char Win16[MAX_PATHS];
        char Win32[MAX_PATHS];
        char Win9x[MAX_PATHS];
        char Winnt[MAX_PATHS];
        Path* Ppath;
};

class Component
{
    public:
        char Name[MAX_NAME];

```

```

        char Pragma[MAX_PRAGMA];
        char Description[MAX_DIS];
        Path *Ppath;
        Enum *Penum;
        Group *Pgroup;
        Table *Ptable;
};

```

```

                                MIF
MOF .
                                MIF          MOF      가
                                .          group  attribute

```

Start Component

Name = "Any Computer System, Model 100"

Description = "This component represents the base configuration of a system ..."

Start Path

Name = "Chassis Group Code"

Dos = "C:\\any\\dos\\chassis.ovl"

Win16 = "C:\\any\\win3x\\chassis.dll"

End Path

Start Group

Name = "ComponentID"

Id = 1

Class = "DMTF|ComponentID|001"

Description = "This group defines attributes common to all component. This ..."

Start Attribute

Name = "Manufacturer"

Id = 1

Access = Read-Only

Storage = Common

Type = String(64)

Value = "Any Computer System, Inc."

End Attribute

...

End Group

...

End Component

가 . MIF MOF . component ID 3

instance of DMI_Path

```
{
  Name = "Chassis Group Code";
  CompId = 3;
  Dos = "C:\\any\\dos\\chassis.ovl";
  Win16 = "C:\\any\\win3x\\chassis.dll";
};
```

instance of DMI_Component

```
{
  Id = 3;
  Name = "Any Computer System, Model 100";
  Description = "This component represents the base configuration of a system ...";
}
```

instance of DMI_Group

```
{
  Id = 1;
  Name = "ComponentID";
  CompId = 3;
  ClassName = "DMTF|ComponentID|001";
  Description = "This group defines attributes common to all component. This ...";
}
```

instance of DMI_Attribute

```
{
  Id = 1;
  Name = "Manufacturer";
  CompId = 3;
  GroupId = 1;
  Storage = "common";
  Access = "read";
  Type = "string(64)";
  EnumExist = FALSE;
```

```

    [read]
    Value = "Any Computer System, Inc.";
}

instance of DMI_IncludesPath
{
    CompId = 3;
    PathName = "Chassis Group Code";
    Component = "\\.\Root\DMI:Id=3";
    Path = "\\.\Root\DMI:Name=\Chassis Group Code\",CompId=3";
}

instance of DescribedByGroup
{
    CompId = 3;
    GroupId = 1;
    Component = "\\.\Root\DMI:Id=3";
    Group =
        "\\.\Root\DMI:Id=1,CompId=3,ClassName=\DMTF|Component|001\"";
}

instance of DMI_IncludesAttrib
{
    CompId = 3;
    GroupId = 1;
    AttribId = 1;
    Group =
        "\\.\Root\DMI:Id=1,CompId=3,ClassName=\DMTF|Component|001\"";
    Attrib = "\\.\Root\DMI:Id=1,CompId=3,GroupId=1";
}

```

DMI MIF - (command-line) ,

mif2mof <source mif> <target mof> <component Id>

MIF component Id
MOF

4.3 SNMP Proxy

SNMP

MIRS API C

```

typedef unsigned long SnmpProxyError_t;
typedef char *SnmpOid_t;
typedef struct SnmpValue {
    char type[100];
    char value[256];
};
typedef char *SnmpSetValue_t;

SnmpProxyError_t SnmpGet (SnmpOid_t oid, SnmpValue_t **value)
SnmpProxyError_t SnmpGetNext (SnmpOid_t oid, SnmpValue_t **value)
SnmpProxyError_t SnmpSet (SnmpOid_t oid, SnmpSetValue_t value)

```

CIMOM 가

. CIMOM IWbemServices, IWbemClassObject Component Object
 Model (COM) . CIMOM WBEM Query Language
 (WQL) . WQL Structured Query Language (SQL)
 CIM 가

```

/// Select <* | property_list> From <ClassName> Where <selection_criteria>
/// References of “{” <target_object> “}” [ where <condition_list> ]
/// Associators of “{” <target_object> “}” [ where <condition_list> ]

```

Select, From, Where

. Reference of <target
 _object> 가 (association)
 , Associators of association

IWbemServices ExecQuery ExecQuerAsync

4.4 DMI Proxy

DMI	MIRS API	DMI	MI
			DmiProxy
		가	
DmiProxyErrorStatus_t	DmiProxyListComponents (DmiProxyRequestMode_t	requestMode,
		DmiProxyUnsigned_t	maxCount,
		DmiProxyBoolean_t	getPragma,
		DmiProxyBoolean_t	getDescription,
		DmiProxyId_t	compId,
		DmiProxyComponentList_t	**reply);
DmiProxyErrorStatus_t	DmiProxyGetAttribute (DmiProxyId_t	compId,
		DmiProxyId_t	groupId,
		DmiProxyId_t	attribId,
		DmiProxyAttributeValues_t	*keyList,
		DmiDataUnion_t	**value);
DmiProxyErrorStatus_t	DmiProxySetAttribute (DmiProxyId_t	compId,
		DmiProxyId_t	groupId,
		DmiProxyId_t	attribId,
		DmiProxyAttributeValues_t	*keyList,
		DmiProxySetMode_t	setMode,
		DmiDataUnion_t	*value);
DmiProxy ErrorStatus_t	DmiProxyGetMultiple (DmiProxyMultiRowRequest_t	*request,
		DmiProxyMultiRowData_t	**rowData);
	SNMP	SNMP	MIRS API
CIMOM	가		가

4.5 GMIRS

GMIRS PC
 MagicMan . MagicMan DMI 2.0 , Windows
 NT Windows 95 . GMIRS MagicMan Service Provider
 , Service Provider GMIRS DMI Proxy DMI MIRS API
 (management application)

15 MagicMan ()



15 MagicMan

Listing Functions DmiGetAttribute(),

DmiGetMultiple()

15

,

DmiSetAttribute()

DmiSetMultiple()

. Service Provider

DMI MIRS API

,

attribute

(GMIRS

.)

Component Instrumentation code

가

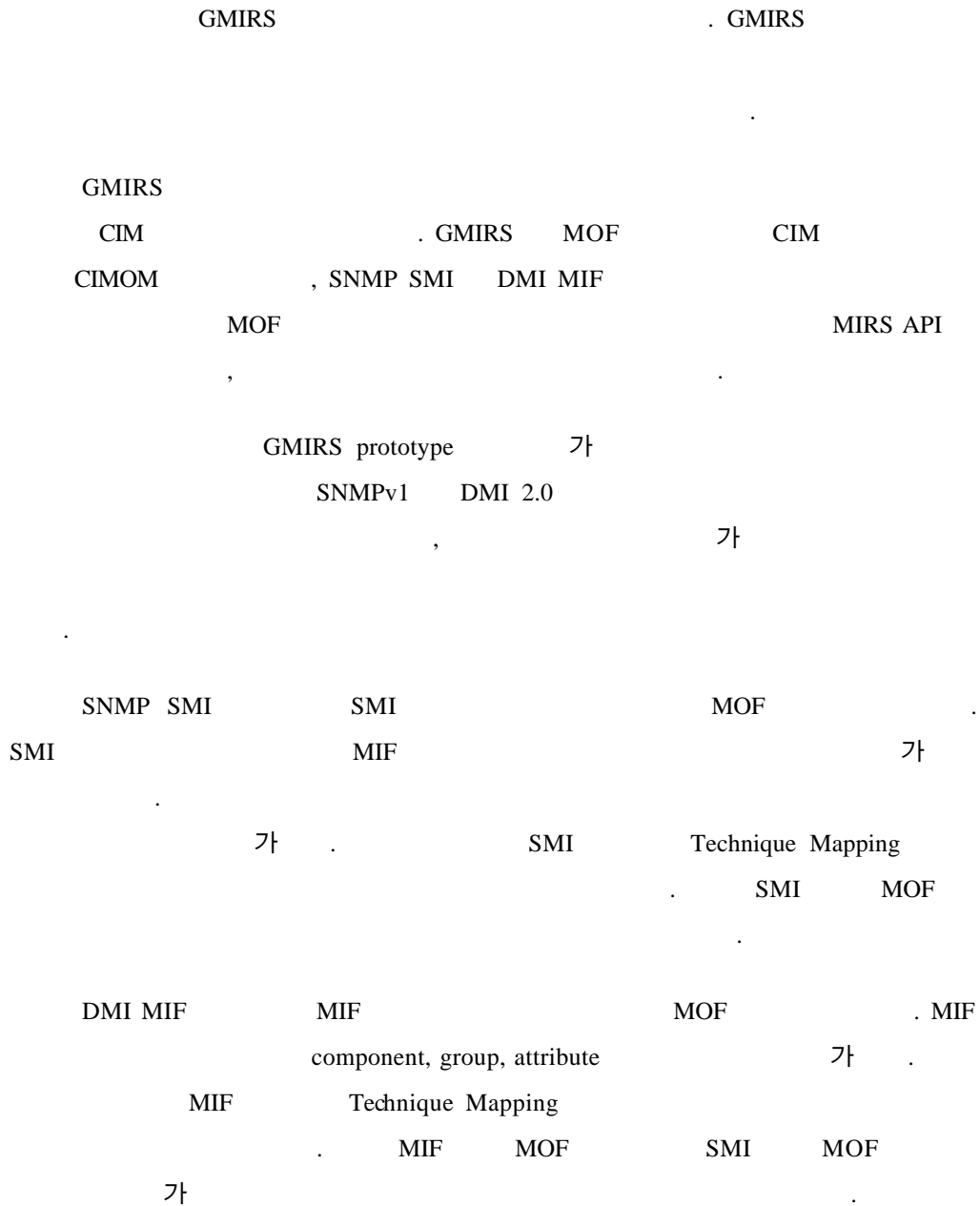
SNMP

(agent)

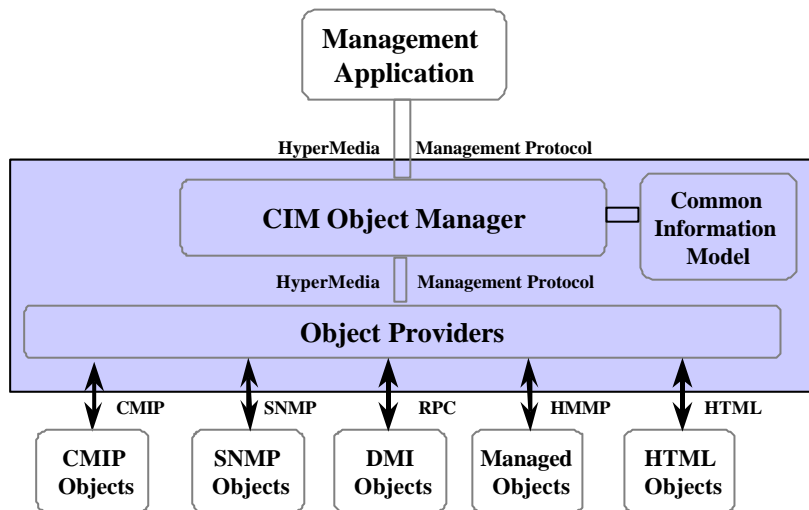
,

-

5.



CIMOM MOF CIMOM MOF
 .
 SNMP GET, GETNEXT, SET SNMP MIRS API
 , DMI Management Interface (MI) Listing Functions
 Operation Functions DMI MIRS API . API
 CIMOM
 SQL WQL
 CIMOM MOF Microsoft WBEM SDK Beta 2



16 WBEM

GMIRS

(agent)

가

가

, CMIP, SNMPv2

,
(meta schema) , CIM , core model CIM
extension schemas common model,

16 WBEM

- [1] Perkins, David. "Understanding SNMP MIBs", Prentice Hall, Inc., 1997.
- [2] Stallings, William, "SNMP, SNMPv2, and CMIP : The Practical Guide to Network-Management Standards", Addison-Wesley, 1993.
- [3] Stallings, William. "SNMP, SNMPv2, and RMON : practical network management - 2nd ed.", Addison-Wesley, 1996.
- [4] Desktop Management Task Force (DMTF), Inc., "Desktop Management Interface (DMI) Specification Version 2.0", March, 1996.
- [5] Desktop Management Task Force (DMTF), Inc., "Common Information Model (CIM) Specification, Version 1.1", September, 1997.
- [6] Microsoft WBEM SDK, <http://www.microsoft.com/management/wbem/>.
- [7] WBEM <http://wbem.freerange.com/>.
- [8] Bostock, B. "DMTF SNMP to DMI Mapping Standard", RFC 950713, July, 1995.
- [9] Tanenbaum, Andrew. "Computer Networks, Third Edition", Prentice-Hall, 1996.
- [10] Shirley, John. "Guide to Writing DCE Applications, Second Edition", O'Reilly & Associates, Inc., 1994.
- [11] Bloomer, John. "Power Programming with RPC", O'Reilly & Associates, Inc., 1992.
- [12] Booch, Grady, and James Rumbaugh. "Unified Method for Object-Or Document Set", Rational Software Corporation, 1996, <http://www.rational.com/ot/uml.html>.
- [13] Case, J. "A Simple Network Management Protocol (SNMP)", RFC 1157 , 1990.
- [14] Case, J. "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, 1996.
- [15] Case, J. "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, 1996.
- [16] Rose, M. "Structure and Identification of Management Information for TCP/IP

- based Internets”, RFC 1155, 1990.
- [17] Rose, M. ”Concise MIB Definitions”, RFC 1212, 1991.
 - [18] Case, J. “ Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC 1902, 1996.
 - [19] Case, J. “Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC 1903, 1996.
 - [20] Case, J. “Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC 1904, 1996.
 - [21] McCloghrie, K. “Management Information Base for Network Management of TCP/IP-based internets: MIB-II”, RFC 1213, 1991.
 - [22] Case, J. “Management Information Base for Version 2 of the Simple Network Management Protocol”, RFC 1907, 1996.
 - [23] Decker, E. “Definitions of Managed Objects for Bridges”, RFC 1286, 1991.
 - [24] Desktop Management Task Force, <http://www.dmtf.org>.
 - [25] Internet Engineering Task Force, <http://www.ietf.org>.
 - [26] Abstract Syntax Notation One (ASN.1), X.208, ISO 8824.
 - [27] HyperMedia Management Schema,
<http://wbem.freerange.com/wbem/obsm.htm>.
 - [28] OMG, “The Common Object Request Broker : Architecture and Specification Revision 2.0”, OMG TC Document, July 1995.
 - [29] Component Object Model, <http://www.microsoft.com/activex/default.htm>.
 - [30] Java Management API, <http://www.javasoft.com/products/JavaManagement>.
 - [31] Orfali, Robert, “The Essential Client/Server Survival Guide, Second Edition”, 1996.
 - [32] HyperMedia Managemet Protocol,
<http://wbem.freerange.com/wbem/hmmp.htm>.
 - [33] Common Management Information Protocol Specification , X.711, X.712.

A. SMI Technique Mapping Schema

```
#pragma
    namespace("\\\\.\Root\SNMPv1")
#pragma schema("Meta")
#pragma schema("SNMPv1")

[abstract, Schema ("SNMPv1")]
class SmpClass
{
};

[Schema ("SNMPv1")]
class SNMP_Table : SmpClass
{
    [key, read]
    string OID;
    [read]
    string Name;
    [read, write]
    uint32 LastRowIndex;
};

[Schema ("SNMPv1")]
class SNMP_Template : SmpClass
{
    [key, read]
    string OID;
    [read]
    string Name;
    [read]
    string Index;
};

[Schema ("SNMPv1")]
class SNMP_TemplateColumn :
    SmpClass
{
```

```
    [key, read]
    string OID;
    [read]
    string Name;
    [read]
    string Access;
    [read]
    string Type;
    [read]
    string Value;
};
```

```
[Schema ("SNMPv1")]
class SNMP_Row : SmpClass
{
    [key, read]
    string OID;
    [read]
    string Name;
    [read]
    string Index;
    [key, read]
    uint32 RowIndex;
};
```

```
[Schema ("SNMPv1")]
class SNMP_Column : SmpClass
{
    [key, read]
    string OID;
    [read]
    string Name;
    [key, read]
    uint32 RowIndex;
    [read]
    string Type;
    string Value;
};
```

```
[Schema ("SNMPv1")]
class SNMP_Scalar : SmpClass
{
    [key, read]
    string OID;
```

```

    [read]
    string Name;
    [read]
    string Type;
    string Value;
};

//=====
//Associations
//=====

[Association, Schema ("SNMPv1")]
class SNMP_HasTemplate :
    SmpClass
{
    [key, read]
    string TableOID;
    SNMP_Table ref Table;
    SNMP_Template ref Template;
};

[Association, Schema ("SNMPv1")]
class SNMP_MadeWith : SmpClass
{
    [key, read]
    string TableOID;
    [key, read]
    string TemplateOID;
    [key, read]
    uint32 ColumnIndex;
    SNMP_Template ref Template;
    SNMP_TemplateColumn ref
        TemplateColumn
};

[Association, Schema ("SNMPv1")]
class SNMP_Includes : SmpClass
{
    [key, read]
    string TableOID;
    [key, read]
    uint32 RowIndex;
    SNMP_Table ref Table;
    SNMP_Row ref Row;
};

};

[Association, Schema ("SNMPv1")]
class SNMP_ConsistsOf :
    SmpClass
{
    [key, read]
    string TableOID;
    [key, read]
    string RowOID;
    [key, read]
    uint32 RowIndex;
    [key, read]
    uint32 ColumnIndex;
    SNMP_Row ref Row;
    SNMP_Column ref Column;
};

```

B. MIF Technique Mapping Schema

```
#pragma
    namespace("\\\\.\\Root\\DMI")
#pragma schema ("Meta")
#pragma schema ("DMI")

[abstract, Schema ("DMI")]
class DmiClass
{
};

[Schema ("DMI")]
class DMI_Path : DmiClass
{
    [key, read]
    string    Name;
    [key, read]
    uint32    CompId;
    [read]
    string    Dos;
    [read]
    string    MacOS;
    [read]
    string    Os2;
    [read]
    string    Unix;
    [read]
    string    Win16;
    [read]
    string    Win32;
    [read]
    string    Win9x;
    [read]
    string    WinNt;
};

[Schema ("DMI")]
class DMI_EnumContent : DmiClass
{
    [read]
    string    Name;
```

```
    [read]
    sint32    Value;
};
```

```
[Schema ("DMI")]
class DMI_Enum : DmiClass
{
    [key, read]
    string    Name;
    [key, read]
    boolean    HasName;
    [key, read]
    uint32    CompId;
    [read]
    uint32    Count;
    [read]
    EnumContent    List[]
};
```

```
[Schema ("DMI")]
class DMI_Component : DmiClass
{
    [key, read]
    sint32    Id;
    [read]
    string    Name;
    [read]
    string    Description;
    [read]
    string    Pragma;
};
```

```
[Schema ("DMI")]
class DMI_Group : DmiClass
{
    [key, read]
    uint32    Id;
    [read]
    string    Name;
    [key, read]
    uint32    CompId;
    [key]
    string    ClassName;
    [read]
```

```

    string    Key;
    [read]
    string    Description;
    [read]
    string    Pragma;
};

```

```

[Schema ("DMF")]
class DMI_Attribute : DmiClass
{
    [key, read]
    uint32    Id;
    [read]
    string    Name;
    [key, read]
    uint32    CompId;
    [key, read]
    uint32    GroupId;
    [read]
    string    Description;
    [read]
    string    Pragma;
    [read]
    string    Storage;
    [read]
    string    Access;
    [read]
    string    Type;
    [read]
    boolean   EnumExist;
    string    Value;
};

```

```

[Schema ("DMF")]
class DMI_Table : DmiClass
{
    [key, read]
    uint32    Id;
    [read]
    string    Name;
    [key, read]
    uint32    CompId;
    [key, read]
    string    ClassName;
};

```

```

    [read]
    string    Key;
    [read]
    string    Pragma;
    [read, write]
    uint32    LastIndex;
};

```

```

[Schema("DMF")]
class DMI_TableColumn : DmiClass
{
    [read]
    uint32    Id;
    string    Value;
};

```

```

[Schema("DMF")]
class DMI_TableRow : DmiClass
{
    [key, read]
    uint32    CompId;
    [key, read]
    uint32    TableId;
    [key, read]
    uint32    Index;
    DMI_TableColumn Column[];
};

```

```

//=====
//Associations
//=====

```

```

[Association, Schema ("DMF")]
class DMI_DescribedByGroup :
    DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    uint32 GroupId;
    DMI_Component ref Component;
    DMI_Group ref Group;
};

```

```

[Association, Schema ("DMF")]
class DMI_DescribedByTable :
    DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    uint32 TableId;
    DMI_Component ref Component;
    DMI_Table ref Table;
};

[Association, Schema ("DMF")]
class DMI_IncludesAttrib :
    DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    uint32 GroupId;
    [key, read]
    uint32 AttribId;
    DMI_Group ref Group;
    DMI_Attribute ref Attrib;
};

[Association, Schema ("DMF")]
class DMI_UsesTemplate : DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    uint32 TableId;
    DMI_Table ref Table;
    DMI_Group ref Group;
};

[Association, Schema ("DMF")]
class DMI_IncludesRow : DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    uint32 TableId;
    [key, read]
    uint32 RowIndex;
    DMI_Table ref Table;
    DMI_TableRow ref Row;
};

[Association, Schema ("DMF")]
class DMI_UsesName : DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    string EnumName;
    DMI_Component ref Component;
    DMI_Enum ref Enum;
};

[Association, Schema ("DMF")]
class DMI_UsesUnnamed :
    DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    uint32 GroupId;
    [key, read]
    uint32 AttribId;
    DMI_Attribute ref Attrib;
    DMI_Enum ref Enum;
};

[Association, Schema ("DMF")]
class DMI_IncludesPath : DmiClass
{
    [key, read]
    uint32 CompId;
    [key, read]
    string PathName;
    DMI_Component ref Component;
    DMI_Path ref Path;
};

```