

## 생산장비 객체화와 개방형 가공 셀 구축 연구(II) -개방형 가공 셀 구축-

김선호\*, 김동훈\*, 박경택\*

### Open Manufacturing System Using MMS Service and Object Oriented Manufacturing Devices(2nd Report)

Sun Ho Kim\*, Dong Hoon Kim\*, Kyoung Taik Park\*

#### ABSTRACT

Manufacturing devices must be integrated for constructing CIM environment, but integration of heterogeneous system meets several economical and technological difficulties. In this case, MMS(Manufacturing Message Specification) can be an effective method, but it has problems which MMS products are generally very expensive and majority of the existing manufacturing devices do not support MMS. This paper describes the development of a gateway for Non-MMS-Compatible CNC machine tools to serve MMS and the implementation of OSI(Open System Interconnection) upper layer on TCP/IP. The developed system is applied to construct cell controller with heterogeneous devices under CIM environment and to evaluate interoperability and portability.

**Key Words** : VMD(가상제조장치), Cell Controller(셀 제어기), OMS(개방형 생산 시스템), TCP/IP, MMS(생산 메시지 규약), CIM(컴퓨터 통합 생산 시스템)

#### 1. 서론

MMS(Manufacturing Message Specification) 서비스를 이용해 가공 셀(Cell)을 구축하고자 할 때, 다양한 벤더에 의해 제작된 생산장비를 어떻게 효율적으로 결합할 것인가 하는 것이 문제가 될 수 있다. 이러한 문제는 기존 사용중인 생산장비를

활용해 가공 셀을 구축하고자 할 때, 더욱 문제가 된다. 종전에는 다양한 생산장비를 이용해 가공 셀을 구축하고자 하는 경우, 컨트롤러의 기능을 분석하고 그에 적합한 방법을 선택해 전용라인을 구축하고 1:1로 인터페이스 하는 방법이 사용되었다. 이러한 예를 Fig. 1에 나타내었다. 이러한 방법은 구성이 간편한 장점은 있으나, 확장성이나

\* 한국기계연구원 자동화연구부(Automation Eng. Dept., KIMM)

이식성에서 많은 단점을 가진다. 또한, 제어구조가 계층적이기 때문에 정보교환에 많은 어려움이 있었다.<sup>(1)</sup>

이러한 문제를 해결하기 위해 80년대 이후부터 생산자동화를 위한 통신기술의 표준인 MAP(Manufacturing Automation Protocol)이 등장하게 되었다.<sup>(2,3)</sup> 생산장비간 통신에서의 객체지향적 개념을 도입한 MMS는 MAP의 최상위 응용계층에 위치하며, 상위 응용계층 메시지 및 서비스 표준으로 제안되었고, 그 후 ISO/IEC 9506 표준으로 채택되었다.<sup>(4)</sup>

MMS는 네트워크상의 분산된 응용 소프트웨어 태스크(Task) 사이의 교환 메시지 및 서비스의 표준을 정의하고 있으며 CNC(Computer Numerical Control), Robot, PLC(Programmable Logic Controller) 등을 지원하고 있다. MMS를 지원하는 통신 프로토콜은 MMS on Full MAP, MMS on Mini-MAP, MMS on Ethernet, MMS on TCP/IP(Transmission Control Protocol/Internet Protocol)등으로 발전되고 있다. OSI(Open Systems Interconnection) 표준을 기반으로 하는 MAP은 그 복잡성 때문에 널리 보급되지 못하고 상품성을 잃어 감에 따라 MMS를 TCP/IP 및 필드버스(Fieldbus) 상에 구현하려는 연구도 이루어지고 있다. OSI 7계층 중에서 3-6 계층을 OSI 프로토콜을 사용하고 있는 MMS on Ethernet는 몇몇 업체에서 상용화는 했으나,<sup>(5)</sup> 가격이 비싼편이며, 무엇보다 환경설정 및 운용이 다소 복잡한 점이 단점으로 인식되고 있다.

결론적으로 이러한 국제적인 표준제정에도 불구하고 FMS(Flexible Manufacturing System)나 CIM(Computer Integrated Manufacturing) 등을 구축할 때 MMS가 보편적인 방법으로 사용되지 못하고 있는 실정이다. 그 이유로서는 첫째, 현재 적용 가능한 MMS를 구성하는 프로토콜이 어렵고 MMS 지원상품이 비싸다는 점과, 둘째, 현재 운영 중인 생산장비의 대다수가 MMS를 지원하지 않는다는 점이다.<sup>(6,7)</sup>

이러한 문제를 해결하기 위한 최근의 연구로는 대중성을 고려하여 MMS 자체를 인터넷망의 TCP/IP 프로토콜 상에서 구현하려는 연구와 CNC, PLC 등의 내부에 MMS 기능을 실장하려는 연구로 이루어지고 있다. 그러나 MMS 서비스를

TCP/IP로 매핑하는 방안에 대해서는 몇 가지 방법이 논의되고 있으나 아직까지 표준은 제정되어 있지 않다.

따라서 MMS가 갖는 이식성, 확장성과 같은 특성이 고려된 셸 환경을 구축하기 위해서는, TCP/IP와 같은 대중성과 경제성이 고려된 MMS 서비스의 개발과 더불어, 기존장비를 객체화하여 MMS 표준을 이용할 수 있는 게이트웨이(Gateway)와 같은 중간기술에 대한 연구가 필요하다. 이러한 연구의 일환으로 기존에 사용중인 공작기계의 CNC를 MMS 서비스를 지원하는 게이트웨이를 이용해 객체화하는 방법이 연구되었다.<sup>(7,8)</sup>

본 연구에서는 개발된 게이트웨이를 이용해, TCP/IP로 MMS 서비스를 대응시키는 방식인 MMS on TCP/IP를 가공 셸 컨트롤러에 적용하고자 했다. 이를 위해 MMS 표준에 적합한 CNC VMD(Virtual Manufacturing Device)를 개발하고, 이를 게이트웨이에 적용했다. 본 논문에서는 이러한 연구결과를 소개한다.

## 2. 적용된 MMS의 특성 및 환경구축

MMS on TCP/IP 구현의 대표적인 방식으로는 어댑테이션 레이어(Adaptation Layer) 및 컴프레스트 스택(Compressed Stack) 방식 등이 있는데 SISCO, Marben, Cycle Software 등 대부분의 MMS on TCP/IP 벤더 제품들이 확장성이 있는 어댑테이션 레이어 방식을 채택하고 있다.<sup>(9,10)</sup> 그러나 응답 속도면에서 실시간을 필요로 하는 환경에서는 오

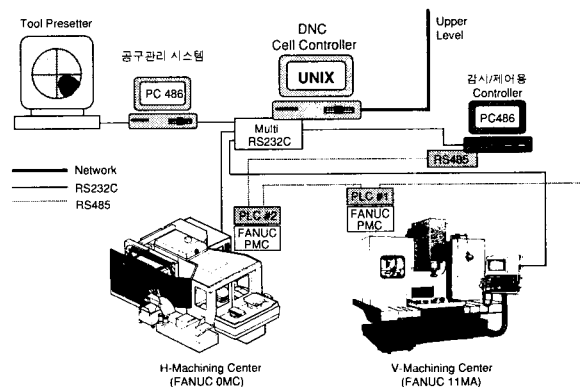


Fig. 1 Machining cell with heterogeneous CNC

Table 1 Methods of MMS on TCP/IP

방식	Adaptation Layer 방식	Compressed Stack 방식
장점	-OSI 7 Layer의 장점 유지 -타 종류의 OSI 응용 프로토콜을 그대로 사용 가능	-응답시간 빠름(실시간 응용) -네트워크 구성이 간단 -경제적임
단점	-응답 속도가 늦음 -OSI 7 Layer를 모두 이 용함으로써 부담이 됨	-프로토콜 설계 복잡 -Layer개념의 장점 활용 제한

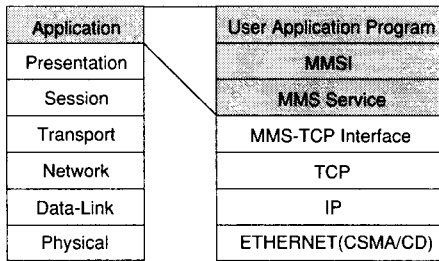


Fig. 2 Application layer of MMS on TCP/IP

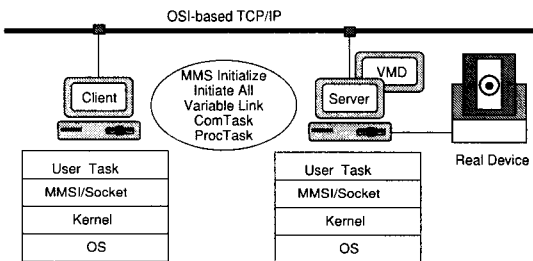


Fig. 3 Environment for MMS on TCP/IP service

히려 네트워크 구성이 간단하고 저렴한 컴프레스드 스택 방식이 장점이 있다. 이러한 특징을 Table 1에 정리했다. 어댑테이션 레이어 방식은 TCP/IP 위에서 OSI의 세션 층과 프리젠테이션 층을 구동시키기 위해 TCP/IP와 세션 층 사이에 RFC1006 어댑테이션 레이어를 두는 것이며, 컴프레스드 스택 방식은 2개의 레이어 스택 위에 MMS 프로토콜을 제공하는 ISO/TC184/SC5/WG2 N578을 사용하는 것으로, 서로 장단점이 있지만 향후 효율성 입장에서 보면 컴프레스드 스택 방식

이 유리할 것으로 보인다.<sup>(11)</sup>

본 연구에서 채택한 MMS on TCP/IP의 구현 방식은 2가지 방식의 장단점을 고려한 방식으로 N578과 유사한 방법을 사용하고 있다.<sup>(12,13,14)</sup> 특징으로는 하위의 프로토콜에 관계없이 사용자에게 일관성 있는 API(Application Program Interface)를 제공하는 방식을 선택했다. 즉, 컴프레스드 스택 방식과 유사하지만 응용층에 MMS 서비스를 위한 다양한 API 함수를 두어 보다 소프트웨어 측면을 강화한 방식이다.

본 연구에서 MMS를 서비스하기 위해 적용한 통신계층을 Fig. 2에 나타내었다. 이는 OSI 7계층과 실제 응용시 MMS 운용을 위한 재구성한 시스템 응용 계층구조를 나타낸 것으로 응용 계층은 응용 프로그램과 MMS, MMSI(Manufacturing Message Specification Interface)가 존재하게 되며, MMSI는 표준 API로서 응용 계층에서 표준 규약인 MMS를 이용하고 이를 이용하는 측면에서 공통의 상호 인터페이스를 둬으로써 MMS 인터페이스라는 표준 API의 라이브러리를 사용하여 응용 프로그램을 작성하였다.

MMS on TCP/IP 운용 대상기계는 OAC(Open Architecture Controller) 및 FANUC 컨트롤러에 의해 제어되는 공작기계가 사용되었다. 그리고 실제 장비의 제어 및 모니터링을 위한 VMD 서버가 구성되었으며, TCP/IP 기반에 별도의 클라이언트를 연결하였다. MMS on TCP/IP 운용 실험 환경은 Fig. 3과 같이 공작기계 제어감시를 위한 VMD 서버를 구성하였으며, TCP/IP 기반의 LAN에 별도의 클라이언트를 연결하였다. 클라이언트와 VMD 서버는 멀티태스킹이 가능한 윈도우상에서 MMS 인터페이스 및 통신 소켓 서비스 인터페이스(Socket Service Interface)를 담당하는 DLL(Dynamic Link Library) 형식의 미들웨어(Middleware)를 두었다. 이는 MMS 서비스를 운용하기 위해 필요한 각 프로세스 태스크간에 작업전환을 위해서는 태스크간의 역할분담 및 CPU 사용의 효율적인 분담이 요구되기 때문이다. 즉, 호스트와의 연결 및 설정을 담당하는 Comtask 및 클라이언트와 서버 역할을 담당하는 ProcTask 등을 미들웨어상에서 생성한다.

응용 프로그램을 작성하기 위한 관점에서 보면, 미들웨어상에서 MMS 응용 프로그램을 운용

Table 2 Used API functions

· extern "C" __declspec(dllimport) UINT MMS_initialize(); // MMS 서비스를 위한 프로그램 초기화
· extern "C" __declspec(dllimport) UINT Initiate_all(); // Client-Server간 연결 초기화
· extern "C" __declspec(dllimport) UINT Initiate(char *CH_name); // 연결을 위한 통신 채널 설정
· extern "C" __declspec(dllimport) int _get(char *VarName); // 데이터 수집
· extern "C" __declspec(dllimport) int get(char *VarName,int VmdNo); // VMD 별 데이터 수집
· extern "C" __declspec(dllimport) int _put(char *VarName); // 데이터 전송
· extern "C" __declspec(dllimport) int put(char *VarName,int VmdNo); // VMD 별 데이터 전송
· extern "C" __declspec(dllimport) UINT PI(int pi_command,char *pi_Name,char *pi_arg); // 프로그램 호출
· extern "C" __declspec(dllimport) UINT _link_variable(char *VarName,void *Phys_Addr,int Var_Len); // 통신할 변수 링크
· extern "C" __declspec(dllimport) UINT link_variable(char *VarName,void *Phys_Addr, int Var_Len,int VmdNo); // VMD 별 통신할 변수 링크
· extern "C" __declspec(dllimport) UINT link_func_pointer(char *pi_objname,void (*PI_FUNC_NAME)); // PI 내부 함수 지정

하기 위해서는 MMS 서비스 관련 라이브러리(Library)를 지정한 후, DLL에서 제공하는 API들을 사용하게 된다. Table 2와 같이 해당 API들이 extern으로 선언되어 상위 MMS 응용 프로그램과 인터페이스 하게 된다. 본 연구에서 사용된 API는 선행연구에서 개발한 것을<sup>(15)</sup> 기반으로 하여 윈도우 프로그램의 Visual C++ 6.0 언어 환경에 맞게 재정리하여 사용하였다. 이러한 개념을 Fig. 4에 나타내었다. 그림에서 보이듯이 통신 메시지를 처리하는 ReceiveDaemon과 컴퓨터간의 연결 설정을 처리하는 ConnectDaemon 및 MMS 서비스 메시지를 처리하는 MsgDaemon 등이 상호 작동된다.

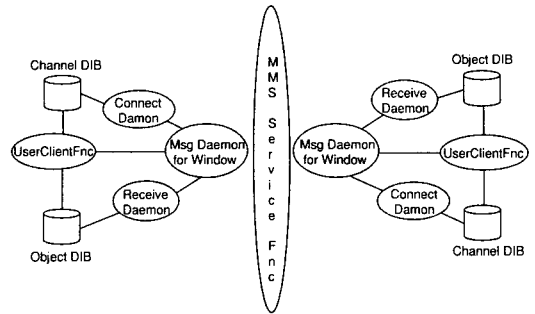


Fig. 4 Operating mechanism of MMS service

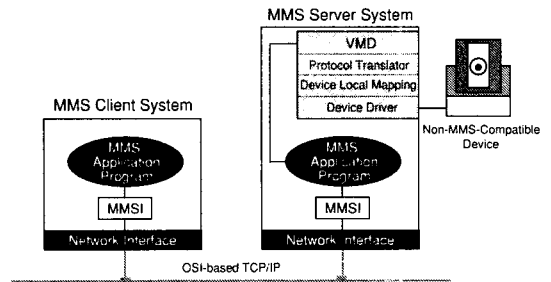


Fig. 5 MMS client/server system

### 3. CNC VMD 구성

MMS 서비스를 실제장비에 연결하기 위한 MMS 클라이언트/서버 시스템의 구성 예를 Fig. 5에 나타내었다. 클라이언트와 서버는 표준 네트워크 인터페이스를 통해 TCP/IP와 연결된다.<sup>(12,13,14,15)</sup> MMSI와 연결되는 MMS 응용 프로그램은 서버쪽이 복잡한 구조를 갖는다.

서버에서의 응용 프로그램은 기계의 특성과 기능에 대한 가상 모형인 VMD 모듈, 프로토콜 전환을 위한 프로토콜 전환 모듈, 실제장비를 제어하거나 모니터링하기 위한 디바이스 로컬 매핑(Device Local Mapping) 모듈 그리고 장비 구동기 모듈로 구성된다.

윈도우 상에서 TCP/IP를 기반으로 구현된 VMD용 API는 get(identifier), put(identifier), PI(service\_name, identifier)의 3가지의 함수를 제공한다. MMS 클라이언트에서 VMD 서버에게 요구할 수 있는 서비스의 종류는 read, write, download, start, stop, resume, kill 등이 있다. 여기서 start,

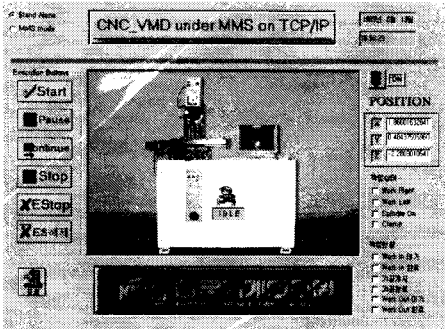


Fig. 6 VMD server operation

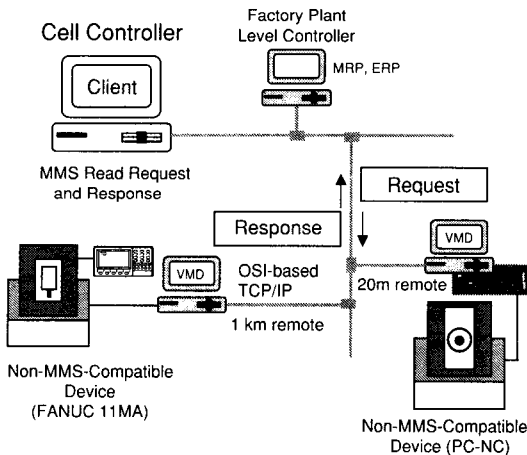


Fig. 7 Machining cell with open architecture controller environment

Table 3 Sample of object

```

CNCTable table; // declare table pointer variable
table = createCNC( 1, "N_CNC");
// create the table. get the pointer value
if ( table == NULL ) { // print error message
}
table->N_NumAxes = 3;
// fill in the number of Machine Axes
table->N_Machine_CS = 1;
// fill in Machine Coordinate Sys No.
    
```

stop, resume, kill은 PI 기능을 수행하게 된다.<sup>(15)</sup>

VMD의 내부 기능으로는 표준화된 객체(object)의 생성, 관리 및 삭제 기능을 갖는다. 이때, 주요 기능으로는 ISO/IEC 9506-4<sup>(4)</sup>에 정의된 주요 데이터 스트럭처(Data structure)를 클래스(Class) 형태로 선언하고 생성된 데이터 스트럭처 형태로 유지 관리 기능 등이 있다.

주요 데이터 스트럭처 생성 함수 기능은 다음과 같다.

- createVmd() : VMD 객체를 생성한다.
- createCNC() : Domain CNC 에 필요한 객체를 생성한다.
- createProgInv() : Domain Program Invocation을 생성한다.
- createPos() : Unnamed type (position) 객체를 생성한다.

도메인 CNC에 객체를 생성하기 위한 객체의 사용 예를 Table 3에 나타내었다. 여기서 N\_CNC 도메인 한정 명명된 변수(Named Variable) 객체는 Table 4에 정리했다.<sup>(4)</sup> Table 4에서 데이터 스트럭처는 C 언어 형식으로 정의했다. 그 이유는 응용 프로그램을 구성할 때 Visual C++6.0을 사용 언어로 했기 때문이다. 따라서, 이는 사용자의 언어 환경에 따라 달라질 수 있다.

구성된 CNC VMD 운용화면을 Fig. 6에 나타내었다.

#### 4. 개방형 가공 셀 구성 및 평가

원거리에 위치한 생산장비들을 결합하여 셀 환경을 구축하고자 하는 경우, RS232C, RS485 등이 적용될 수 있으나 거리문제, 비용문제 그리고 통신 신뢰성이 문제가 될 수 있다.<sup>(1)</sup> 본 연구에서는 기존에 설치되어 있는 TCP/IP 망을 이용함으로써 이러한 문제를 해결하고자 했다. 개발된 MMS on TCP/IP를 이용해 Fig. 7과 같이 KIMM 모델 플랜트 가공 셀을 구성했다. 가공 셀은 CIM 환경하에서 스케줄 및 생산관리를 담당하는 상위 시스템으로부터 작업지시를 받아 하위 생산장비를 관리하는 역할을 하게 된다. 구성된 셀은 MMS 비호환 생산장비(Non-MMS Compatible Device)인 FANUC



Fig. 8 Machining cell on MMS service

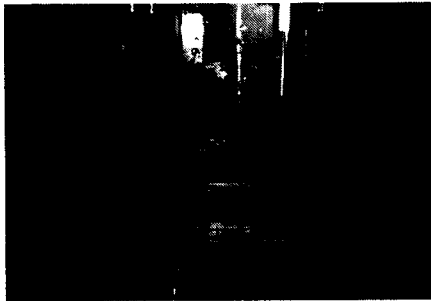


Fig. 9 Machining by MMS service command

11MA와 PC-NC를 대상으로 VMD를 실장하였다. 실제장비와 VMD의 인터페이스 층으로는 FANUC 11MA와 VMD 사이에는 PLC를, 그리고 PC-NC와 VMD 사이에는 PMAC 보드를 두었다.<sup>(7)</sup> 개방형 셀이 갖는 특성을 활용하기 위해, PC-NC는 셀 제어기로 부터 20m의 비교적 가까운 위치에 두었고, FANUC 11MA는 1km 떨어진 원격지에 설치했다. 원격지에 위치한 모델 플랜트에는 공작기계 외에 CMM(Coordinate Measuring Machine), AGV(Automatic Guided Vehicle), ASRS(Automatic Storage and Retrieval System), 트랜스퍼 머신 등이 있으나, 본 연구에서는 CNC 공작기계만을 대상으로 하였다. Fig. 8에 구성된 가공 셀을 나타내었으며, Fig. 9는 MMS 서비스에 의해 가공이 개시되는 장면을 나타낸 것이다.

MMS 운용실험에서는 여러 객체 중에서 변수(Variable) 객체와 PI(Program Invocation) 객체를 정의하여 실험을 하였다. 객체의 정의는 가공 셀에 적합한 최소한의 내용을 선택했다. 선택된 객체

Table 4 Named variables

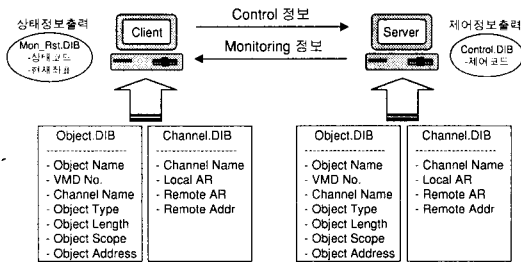
```

typedef struct CNCTable {
    int N_NumAxes; // Number of Machine Axes
    float N_ZeroAxes; // Axes Zero Offset
    int N_AxesValTrans;
    // Axes Value Transformation
    int N_Machine_CS;
    // Machine Coordinate System
    int N_ActProg_CS;
    // Active Program Coordinate System
    float N_GeometryTrans;
    // Geometry Transformation
    float N_FRL; // Feedrate Limit
    bool N_FRO_ON; // Feedrate Override ON/OFF
    float N_FRO; // Feedrate Override
    int N_OSP; // Optional Stop
    bool N_RPO_ON; // Rapid Override ON/OFF
    float N_RPO; // Rapid Override
    float N_NumSpindles; // Number of Spindles
    float N_SSL; // Spindle Speed Limit
    bool N_SSO_ON;
    // Spindle Speed Override ON/OFF
    float N_SSO; // Spindle Speed Override
    float N_DEV_STATE; // Device State
    float N_Referenced;
    // Device Origin Setup Query
    CString N_StoreType;
    // Type of Storage Medium
    UINT N_MaxStoreEntries;
    // Maximum Num of Data Stor Entry Objects
    UINT N_MaxStoreCapacity;
    // Maximum Octet Count of Data
    UINT N_CurrentEntries;
    // Current Number of Data Store Entries
    UINT N_RemainingEntries; // Number of Entries
    UINT N_CurrentCapacityUsed;
    // Octet Count of Data
    UINT N_RemainingCapacity; // Octet Count
} CNCTable;
    
```

CNCTable table;

Table 5 Components for monitoring and control using MMS service

감시	제어
-가동상태(가동/비가동) -가공 위치 데이터	-기동 -기동정지



- \* Object Name : 객체 식별자 이름
- \* VMD No. : 객체가 연결된 VMD 번호
- \* Channel Name : 객체가 사용하는 채널 식별자 이름
- \* Object Type : 객체 형
- \* Object Length : 객체 길이
- \* Object Scope : 객체 범위
- \* Object Address : 객체의 하드웨어 주소
- \* Local AR : 해당 스테이션 자신의 고유번호
- \* Remote AR : 원격 스테이션의 고유번호
- \* Remote Addr : 원격 호스트 명칭

Fig. 10 System environment for MMS service

를 Table 5에 정리했다. 감시 및 제어 내용은 모두 변수(Variable)로 정의했으며, 그 중에서 가동/비가동 상태는 1 바이트(Byte)형, 축 정보는 플로트(Float)형 그리고 제어정보는 PI 객체로 정의했다. 정의된 객체를 대상으로 클라이언트와 서버간에 실제 메시지 전송을 위해서는 초기화 과정이 필요하다. Fig. 10에 나타낸 것처럼 정보처리 관점으로 보면, 먼저 클라이언트와 서버는 각각의 객체 및 채널 정보를 정의한 DIB(Data In Base) 과일을 이용하여 환경을 초기화한다. 객체정보 파일은 주고 받을 변수에 대한 다양한 상세 정보를 정의하며 통신하게 될 채널명을 설정한다. 여기서 설정된 채널 명은 채널정보 파일에 연결시킬 호스트명과 파라미터가 정의되어 있다.

초기화 작업을 마친 후, 셀 환경하에서의 셀 제어기의 기능실험을 수행했다. 기능실험은 상위에서 결정된 스케줄에 의해 각 공작기계로 프로그램을 다운로드 하거나 컨트롤러에서 가지고 있는 프로그램을 호출한다. 클라이언트는 서버의 기동(Cycle Start) 및 현재위치(Position) 값 확인 등 제어정보를 상호 운용이 가능한 객체변수를 이용하여 코드를 보내며, VMD 서버는 머시닝센터의 제어 기능을 수행한 후 필요시 상태 값을 전송한다.

이상과 같이 정의된 객체를 이용해 MMS 운용 실험을 수행한 결과, 클라이언트인 셀 제어기로부터 VMD 서버로 데이터를 요청(Request)한 후, 응답(Response)을 받는 데 약  $1,000 \pm 50 \text{msec}$ . 정도가 소요되었다. 그러나 클라이언트에서 기동제어를 요청했을 때 실제장비가 반응하는 응답시간은 그 절반 정도의 시간, 즉  $500 \pm 30 \text{msec}$ .가 소요된다고 할 수 있다. 응답시간은 통신 카드(실험조건 : 10Mbps)의 사양과 회선망(실험조건 : 10Mbps)의 상태에 따라 단축이 가능하다. 실험결과는 셀/에리어(Cell/Area) 영역<sup>(7)</sup>의 관점에서 판단할 때 실시간 제어가 이루어 졌다고 할 수 있다.

이를 종전에 주로 사용하던 전용라인을 이용한 연결 방법과 비교하면 다소 느리다고 할 수 있다. 종전의 방법에서는 상태정보를 요청한 후 응답까지의 소요시간이  $240 \pm 20 \text{msec}$ . 정도가 소요되었었다.<sup>(1)</sup> 개방형 가공 셀은 이러한 통신속도를 제외하고는 원격제어성능, 확장성, 이식성 등에서 많은 장점을 가지고 있다고 할 수 있다.

## 5. 결 론

본 연구는 선행연구에서 개발된 게이트웨이를 이용해, TCP/IP로 MMS 서비스를 대응시키는 방식으로 개방형 가공 셀을 구축하고자 했다. 이를 위해 MMS 표준에 적합한 CNC VMD를 개발하고 이를 개발된 게이트웨이에 적용했다. MMS 서비스를 이용한 시범 셀은 이기종 컨트롤러를 갖는 2대의 공작기계를 이용해 사무실 환경에서 사용중인 LAN을 이용했다. 본 연구를 통해 다음과 같은 몇가지 결론을 얻었다.

- 1) MMS on TCP/IP에서 운용이 가능한 CNC VMD를 개발하고, 이를 게이트웨이에 적용함

으로서 공작기계의 객체화가 가능했다.

- 2) 개발된 CNC VMD 게이트웨이와 TCP/IP 망을 이용해 개방형 가공 셀 구축이 가능했다. TCP/IP 망은 기존 사용중인 LAN을 활용했다.
- 3) MMS on TCP/IP 서비스를 이용한 개방형 가공 셀은 감시 및 제어를 위한 통신속도가 전용라인을 이용하는 경우보다 다소 느리게 평가되었다. 또한, 경우에 따라서는 네트워크의 상태에 더욱 영향을 받을 수 있다.
- 4) 전용라인을 이용해 가공 셀을 구축하는 경우, 장비의 이동에 따른 이식성, 장비 추가에 따른 작업에 많은 기술과 시간이 필요한 반면, 개방형 셀에서는 원격제어성능, 확장성에서 많은 장점을 갖는다.

### 참고문헌

1. 김선호, 이승우, 안남식, 김성복, 안중환, "DNC 시스템 개발," 한국정밀공학회지, 제12권 제12호, pp. 19-29, 1995.
2. 김정호, 이상범, "MAP 네트워크에서 MMS 운영을 위한 가상제조기기의 설계," 한국정보처리학회지, 제2권 제2호, pp. 397-405, 1995.
3. 권옥현, MMS의 이해와 응용 프로그래밍, 서울대 제어계측 신기술 연구센터, 1997.
4. ISO/IEC 9506-4, Manufacturing Message Specification-Part 4, 1992. 12. 15
5. FANUC Users Manual 15, 150, 16, 160, 18, 180 Series
6. Cheah Raymond Seng-Sim, Lee Bu-Sung and Lim Raymond Long, "Design and Implementation of an MMS Environment on ISODE," Computer Communications, Vol. 20, No. 15, pp. 1354-1364, 1997.
7. 김선호, 김동훈, 박경택, "생산장비 객체화와 개방형 가공 셀 구축 연구(I) -생산장비 객체화-," 한국정밀공학회지, 제16권 제5호, pp.91-97, 1999.
8. 김선호, 김동훈, "이기종 CNC로 구성된 개방형 가공 셀(Cell) 구축 연구," 대한기계학회 동역학 및 제어부문/생산설계부문 공동학술대회 논문집, pp. 170-175, KAIST, 1999.
9. Castori, P., MMS products, <http://litwww.epfl.ch/MMS>
10. SISCO, MMS-EASE reference manual revision 10, 1991.
11. 신용호, 이진환, 이태억, 박정현, "개방형 자동화 통신 기술의 동향과 도입 및 구현 전략," 대한산업공학회춘계학술대회논문집, D06.3, 1998.
12. 강성귀, 김동성, 박홍우, 김영신, 권옥현, 변철웅, "MMS on TCP/IP를 이용한 시범 플랜트 구축," pp.688-691, 13회 KACC 논문집, 1998.
13. 박홍성, 최명환, 김용석, 이철민, 강영진, "로봇용 가상 생산기기 구현," pp. 13-17, 제6회 G7 첨단 생산시스템 Workshop, 1998.
14. 이호길, 한만철, 박홍성, 김선호, 박경택, 표준 네트워크 실장 및 응용기술 개발-2차년도 연차보고서, 통상산업부, 과학기술처, 1998.
15. 박홍성, 김홍석, 이호길, 진창기, 강영진, 최명환, "TCP/IP 상의 MMS를 위한 Easy API 구현," pp. 75-80, 제7회 G7 첨단 생산시스템 Workshop, 1999.