

## M2M환경에서의 지식진화형 지능공작기계

- FIPA-OS를 사용하는 Ping Agent 분석 및 Dialogue Agent 모듈설계

김동훈\* · 송준엽\*\*

## Knowledge Evolution based Machine-Tool in M2M Environment

- Analysis of Ping Agent Based on FIPA-OS and Design of Dialogue Agent Module

Dong-Hoon Kim\* · Jun-Yeob Song\*\*

### ABSTRACT

Recently, the conventional concept of a machine-tool in manufacturing systems is changing from the target of integration to the autonomous manufacturing device based on a knowledge evolution. Subsequently, a machine-tool has been the subject of a cooperation through an advanced environment where an open architecture controller, high speed network and internet technology are contained. In the future, a machine-tool will be more improved in the form of a knowledge evolution based device. In order to develop the knowledge evolution based machine-tool, this paper proposes the structure of knowledge evolution and the scheme of a dialogue agent among agent-based modules such as a sensory module, a dialogue module, and an expert system. The dialogue agent has a role of interfacing with another machine for cooperation. To design of the dialogue agent module in M2M(Machine To Machine)environment, FIPA-OS and ping agent based on FIPA-OS are analyzed in this study. Through this, it is expected that the dialogue agent module can be more efficiently designed and the knowledge evolution based machine-tool can be hereafter more easily implemented.

Key Words : Machine-Tool(공작기계), Knowledge evolution(지식진화), Dialogue agent(대화용통신모듈), Cooperation(협업), M2M, FIPA-OS

---

\* 지능형정밀기계연구부 선임연구원

\*\* 지능형정밀기계연구부 책임연구원

## 1. 서론

앞으로의 생산시스템에서는 공작기계가 협력의 주체가 된다. 즉, 공작기계가 다양한 내외부적 요인들과 협력을 유지하면서 스스로 지식을 진화시킬 수 있는 M2M(Machine To Machine) 환경을 만들어 갈수 있게 될 것이다[1,3]. 본 연구에서는 지식진화형 지능공작기계의 개발을 위한 대화 모듈 에이전트 설계를 위하여 표준 플랫폼 관련 조사와 간단한 Ping 에이전트 분석을 통하여 Dialogue Agent 모듈 설계에 대한 내용을 소개하고자 한다. 이러한 연구의 배경은 지식진화형 지능 공작기계를 개발하기 위해서는 인간 전문가를 대신할 다양한 지식과 이에 적합한 지식처리가 필요하다[2-4]. 그러기 위해서는 무엇보다 기계간 협력을 위한 에이전트의 요구가 필수적이다[5-16]. 우선, 본 논문에서는 이러한 대화 모듈의 에이전트 설계에 앞서, FIPA-OS라는 표준 Framework에 대한 개념을 소개한다. 그리고 이를 사용하는 Ping Agent라는 간단한 에이전트에 대해 분석하고 이를 통하여 M2M 환경에서의 지식진화형 지능기계 개발을 위한 필수 요소인 대화 모듈적인 Dialogue Agent의 Scheme를 제시한다. 앞서 선행 연구에서 지식진화형 공작기계의 의미로 볼 때 3개의 지식진화 메카니즘을 가지고 있다고 언급하였다. 첫째는 인간이 가진 감각기능을 활용하는 것이다. 둘째는 대화기능이다. 즉, 인간만이 가진 언어능력을 이용해서 다른 전문가와의 대화를 통해 지식을 얻고 교환하면서 간접적 경험을 통해 지식을 습득해가는 지식 메카니즘이다. 셋째는 추론기능이다. 이러한 감각기능, 대화기능 그리고 추론기능이 센서 모듈 (Sensory Module), 대화모듈 (Dialogue Module), 그리고 전문가시스템 (Expert System)으로 대체가 가능하다.

본 논문에서는 세가지 모듈 중 기계간 협력을 위해서 필수적인 대화 모듈에 대한 에이전트적 개념 설계를 제시한다. 이를 위해 에이전트 기반 Framework인 FIPA-OS와 이를 이용한 간단한

에이전트를 제시하였고, 이의 발전된 모습인 Dialogue Agent를 제시하였다. 대화모듈 역할을 하는 Dialogue Agent는 Communication Agent를 이용해 다른 기계와 대화를 하게 된다. 즉, 어떤 수행에 대한 임무가 주어졌을 때 관련 지식을 가지고 있는 다른 기계와의 대화를 통해 간접적 경험지식을 습득하게 된다. 이를 위해 3가지의 서브 에이전트를 구성했다. Machinability Agent는 가공성에 대한 지식을 교환하기 위해 사용된다. 즉, 새로운 재료와 공구 및 가공 형상이 주어졌을 때 이를 효과적으로 가공할 수 있는 가공조건에 대한 지식을 수집하게 된다. Manufacturability Agent는 운용성에 대한 지식을 수집하게 된다. 즉, 자신이 맡겨진 수행을 정상적으로 수행을 할 수 있을 것인가에 대한 지식을 수집하게 된다. Plannability Agent는 효과적인 가공공정에 대한 지식을 수집하게 된다.

## 2. M2M 환경

생산시스템에서 공작기계는 통합(Integration)의 대상이 되어왔으나 이러한 기술들이 개발된다면 협력(Cooperation)의 주체가 될 수 있을 것이다. 인간 전문가의 역할이 최소화되고 기계 전문가가 인간 전문가를 대신 할 M2M(Machine to Machine) 환경의 생산 시스템의 구조를 Fig. 1에 나타내었다. M2M 환경을 통해 교환될 수 있는 정보로는 기계 종속적 지식과 기계 독립적 지식이 있다. 이러한 정보들은 M2M에 연결되는 e-machine뿐 아니라 CAM 업체, 공구생산 및 판매업체, 소재생산 및 판매업체, 원격 서비스 중계업체 등과 유기적으로 연결되어 실시간으로 정보를 교환하면서 지식을 진화시킬 수 있게 될 것이다.

## 3. FIPA-OS

다른 기계와의 대화를 통해 간접적 경험지식을

습득하며, 창구를 맡는 역할을 하는 것이 dialogue agent이다. 이러한 dialogue agent와 communication agent는 Fig. 2에 제시된 FIPA (Foundation of Intelligent Physical Agent) 에이전트 표준을 구현한 소프트웨어 에이전트 플랫폼인 FIPA-OS(Open Source)를 기반으로 구현될 것이

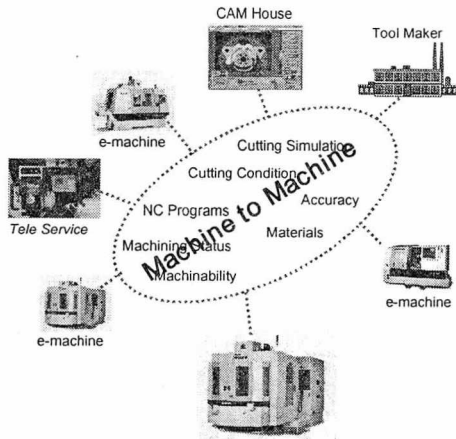


Fig. 1 M2M 생산 시스템의 구조

다. 에이전트 표준을 지향하는 플랫폼은 이태리에서 제안된 JADE, 일본의 Comtec, 미국의 AAP 및 영국의 Notel Networks에서 제안된 FIPA등이 있다. 이 중 FIPA가 가장 멀티 에이전트 표준에 충실하다고 알려져 있다 [2,11]. FIPA-OS는 에이전트의 소멸과 생성 및 ACL(Agent Communication Language) 메시지 통신을 제공하기 위하여 다음의 기본 에이전트 및 요소를 포함하고 있다. DF(Directory Facilitator), 에이전트 관리시스템(Agent Management System), 에이전트 통신채널(Agent Communication System), IPMT(Internal Platform Message Transport), 에이전트 셸(Agent Shell) 등이 있다. DF는 특정형의 에이전트의 검색을 위한 서비스를 제공한다. 에이전트 관리시스템은 에이전트 등록 및 해지를 담당하며, 에이전트 통신채널(ACC: Agent Communication Channel)은 에이전트간의

메시지 통신을 지원한다. 에이전트 셸은 에이전트를 만드는 기본 틀을 제공한다. 에이전트 셸은 Java 기반 클래스 형태로 존재하며, 새로운 에이전트는 기반 클래스에서부터 상속된 형태로 제작된다. 이 밖에 에이전트 셸은 ACL 메시지 관리 및 메시지 프로토콜의 표준에 관한 클래스를 포함하고 있다. IMPT는 특정 에이전트 셸을 기반으로 제작된 에이전트를 위해서 메시지 라우팅 서비스를 제공한다. 이러한 FIPA 에이전트 표준을 구현한 소프트웨어 에이전트 플랫폼인

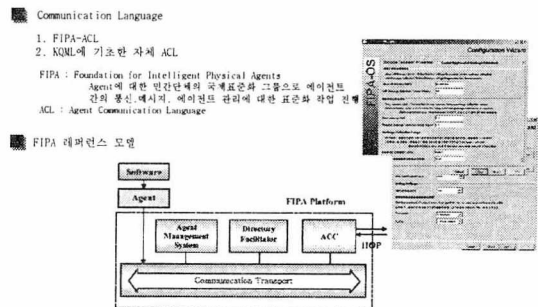
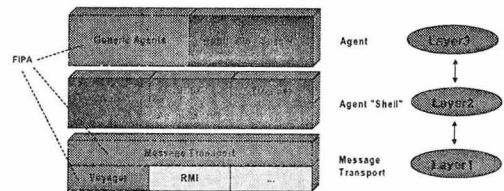


Fig. 2 FIPA-OS 분석

FIPA-OS는 논리적으로 3 layer로 나눌 수 있다  
 • 각각은 Plug-In 형태의 component들로 구성된다



• RMI: Remote Method Invocation  
 가장 프로그램 언어의 개발환경을 사용하여 서로 다른 컴퓨터 상에 있는 객체들이 본산 내트웍 내에서 상호 작용하는 객체지향 프로그래밍 작성할 수 있는 방식. RMI는 일반적으로 IPMI라고 불려지는 것의 차버 버전  
 • JESS: Java Expert System Shell  
 객체의 Java Applet으로 구성되며 앞으로는 차버가 지원되는 브라우저가 필요  
 • Wrapper:  
 Software that accompanies resources or other software for the purposes of improving convenience

Fig. 3 FIPA-OS의 Layered Model 분석

FIPA-OS를 분석하여 dialogue agent를 구현을 목표로 우선, dialogue agent의 scheme를 제시하고자 한다.

FIPA-OS는 Fig. 3처럼 논리적으로 3 layer로 나눌 수 있다. 각각은 Plug-In 형태의 component 들로 구성된다. 이 중 Message Transport layer

가 본 장에서는 관심을 갖는 부분 중 하나이다.

예로서 Video라는 항목이 있다고 하자. 이것 주문하는 신청할 때 아래와 같이 FIPA97 message 안에서 XML content를 작성할 수 있다.

```
<!DOCTYPE ecommerce SYSTEM
"http://www.alcatel.be/xml/dtds/ecommerce.dtd">
<!ELEMENT ecommerce (order|request|offer)>
<!ELEMENT (order|request|offer) (video)+ >
<!ELEMENT video (title, actors, languages)+ >
<!ATTLIST video tape
('VHS'|'BetaCam'|'SuperVHS') 'VHS'>
<!ELEMENT actors (actor)+ >
<!ELEMENT (actor|title) (#PCDATA)>
<!ELEMENT languages EMPTY>
<!ATTLIST languages dubbed NAME #IMPLIED
subtitled NAME #IMPLIED >
Based on the above DTD, an example of a FIPA
message, expressing a request to order a
```

particular

```
movie may look as follows:
request
:sender lisa@iiop://www.geocities.com/acc
:receiver vshop@iiop://www.starpictures.com/acc
:language XML
:ontology
http://www.alcatel.be/xml/dtds/ecommerce.dtd
:content "
<?xml version="1.0">
<ecommerce>
<order>
<video tape='VHS'>
<title>Titanic</title>
<actors><actor>Dicaprio</actor></actors>
<languages dubbed='french'>
</video>
</order>
</ecommerce>"
```

## 4. Ping Agent

본 장에서는 Agent의 구동 시에 스트링 송수신을 위한 데모 프로그램 구현을 위해서 필요한 프로그램과 무엇을 어떻게 이용할 것인가라는 문제에 대해 고찰하고자 한다. 우선 JVM, JDK, FIPA-OS의 버전을 맞추어 구동해야 하며, Ping Agent (Fig. 4 참조)를 고치고 Agent Loader를 분석해서 GUI Window 생성을 하고 메시지를 주고 받을 수 있도록 한다. 그리고, Ping Agent를 분석해서 Agent Loader에 등록하는 법을 정리한다. Agent Loader의 GUI와 form, dialog 과일을 분석해서 Text송수신 프로그램에 사용할 dialog를 만든다. Ping Agent를 분석해서 대상에게 메시지를 보내고 받는 것을 적절히 수정한다. 이렇게 해서 만들어지게 될 에이전트의 모형은 다음과 같다. (Fig. 5 참조)

Service Name	Ping Agent
Service Address	PingAgent@hurcher5.cs.rmit.edu.au:9999/JACK
Service Description	Replies with an "alive" (no quotes in the version sent) confirmation message (with a FIPA performative INFORM) when it receives a FIPAQUERY-REF message with the single word "ping" in the body.
Content Language	Plain Text
Protocol	FIPA QUERY
Interface	<p>The PingAgent accepts a message like the following</p> <pre>(QUERY-REF :sender (agent-identifier :name ds0@goanna.cs.rmit.edu.au:3000/JADE :addresses (sequence http://goanna.cs.rmit.edu.au:9999/acc )) :receiver (set (agent-identifier :name PingAgent@goanna.cs.rmit.edu.au:9999/JACK) :content Ping ))</pre> <p>The PingAgent generates a reply as follow</p> <pre>(INFORM :sender (agent-identifier :name PingAgent@goanna.cs.rmit.edu.au:3000/JADE :addresses (sequence http://goanna.cs.rmit.edu.au:9999/acc )) :receiver (set (agent-identifier :name ds0@goanna.cs.rmit.edu.au:3000/JADE :addresses (sequence http://goanna.cs.rmit.edu.au:9999/acc )) :content Alive ))</pre>

### Ping Agent Example

The Ping Agent will try to ping all the other ping agents that it knows about every 5 minutes, and

will respond to any pings it receives. It does this with tasks:

IdleTask: Is started first. Gets all other agents from DF. After 5mins it starts pinging (PingALLTask). It also responds to any pings it receives (PingResponseTask).

PingAllTask: Sends a ping to all agents (PingTask).

PingTask: Sends a ping.

PingResponseTask: Replies to pings.

Fig. 4 Ping Agent 예시

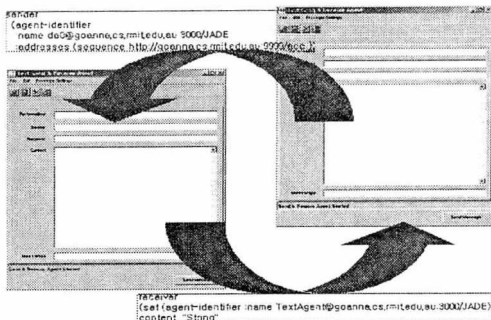


Fig. 5 데모 Agent의 모형

## 5. Dialogue Agent

Dialogue agent의 scheme는 Fig. 6과 같다. 어떤 수행에 대한 임무가 주어졌을 때 관련 지식을 가지고 있는 다른 기계와의 대화를 통해 간접적인 경험지식을 습득하고 지식화하는 역할을 한다. 내부의 sensory agent와 decision agent는 communication agent를 통해 dialogue agent의 인터페이스부와 연결이 된다. Communication agent는 M2M, 즉 다른 기계의 external agent와도 통신 기능을 가질 수 있다. Dialogue agent의 인터페이스부를 통한 데이터는 Interpreter에서 수신된 데이터의 유효성을 판단한다. 그리고 dialogue engine을 거쳐 reasoning에서 협력을 위한 agent의 정보를 관리하는 social knowledge

와 함께 요청된 task에 맞는 적합한 행동을 결정하게 된다. 결과는 다시 dialogue engine과 인터페이스부를 거쳐 communication agent로 보내진다.

Dialogue agent는 Fig. 7처럼 궁극적으로는 내부의 기계가 어떤 작업을 할 때 내부의 지식 외에 외부의 지식이 필요하면 이를 위해서 외부 기계에서 축적된 지식을 가져와 자신의 지식을 향상시키는데 도움을 주기 위해 대화 창구 역할을 하는 것이 목표이다. 이것은 단지 정보 공유를 통한 지능형 기계 개발을 위한 것이 아니라 전문가를 대신할 현장의 인간과 기계 자신, 그리고 원격지의 다른 기계간의 유용한 지식 정보의 전달 및 업그레이드를 위한 매개체 역할을 가지며, 단계적으로 지식 진화형 지능제조설비를 개발하기 위한 기능을 가지도록 발전 될 것이다.

## 6. 결론

본 연구는 선행 연구에서 수행해 왔던 지식진화 기능을 갖는 지능기계의 개발을 위해 필요한 SA(Sensory Agent), DA(Dialogue Agent), DSA(Decision Support agent)의 3가지 기반 연구 중 DA의 설계를 위해 표준 기반 환경인 FIPA-OS와 이를 사용하는 Ping Agent를 분석하고, 이를 통해 DA의 효과적인 설계를 하기 위한 기반 연구를 수행하였다. 본 연구를 통하여 지식진

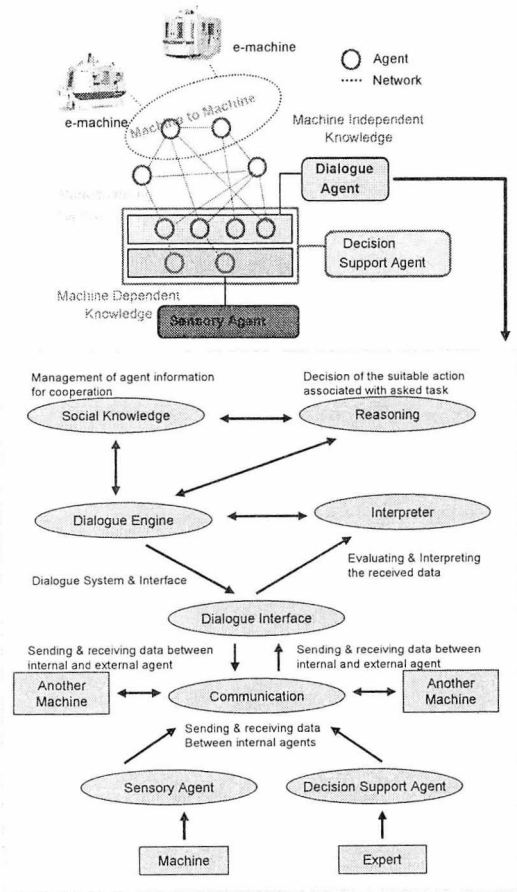


Fig. 6 Scheme of dialogue agent

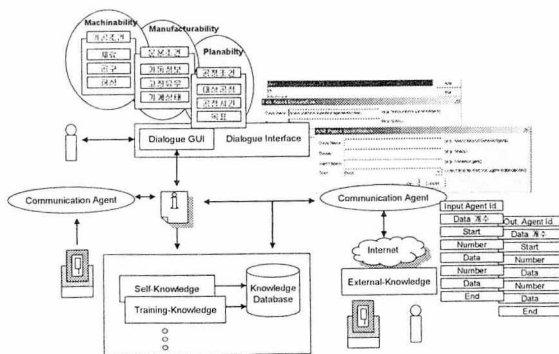


Fig. 7 Structure diagram of dialogue agent

화형 지능기계 개발을 위해 이에 적합한 환경에서의 dialogue agent의 역할과 기능이 기계간 협력을 위하여 고려되었다. 즉 지식 진화의 구조와 이에 따른 지식의 객체 모델 중 에이전트 기반의 대화모듈에 대한 개념이 제시되었다. 우선은 간단한 Ping Agent의 분석 및 수정 작업으로 FIPA 기반의 메시지 인터페이스 중심으로 연구를 진행하였으나, 병행 연구되고 있는 SA와 DSA에 관한 연구와 함께 향후에는 DA의 구체적인 기능을 확장성 있게 구현해 나갈 예정이다.

### 참고 문헌

1. 김동훈, 김선호, 이승우, 임선중, 이안성, 박경택, 고흥식, "지능공작기계 개발을 위한 Dialogue Agent의 Scheme 설계", 한국정밀공학회 춘계학술대회 논문집, pp. 817-820, 2004.
2. 김동훈, 김경돈, 김찬봉, 김선호, 고흥식, "개방형 CNC 공작기계의 운용상의 고장에 대한 그 원인진단 및 서비스", 제 5회 고속지능형 가공시스템기술 워크샵, pp. 151-155, 2004.
3. 김선호, 김동훈, 이승우, 임선중, 이안성, 박경택, "지식 진화형 지능공작기계 지식구조 설계-", 한국정밀 공학회 추계학술대회논문집, pp. 509-512, 2003.
4. 김선호, "지식기반형 지능화 기계와 지식진화형 지능화 기계," 한국정밀공학회지, 제19권 제2호, pp.17-25, 2002.
5. 박홍석, "에이전트 기술 응용 Shop Floor 제어방안," 한국정밀공학회지, 제18권 제4호, 2002.
6. 황지현, 최경현, 이석희, "지능에이전트를 이용한 개방형 셀 제어기 개발," 한국정밀공학회 2001년도 춘계학술대회논문집, 2001.
7. 허준규, 최경현, 이석희, "가상기업을 위한 멀티에이전트 기반 태스크 할당 시스템에

- 관한 연구," 한국공작기계학회논문집, 제12권 제 3호, 2003.
8. 최중민, "에이전트의 개요와 연구방향," 정보과학회지 15권 3호, pp 7-16, 1997.
  9. 김선호, 김동훈, 박경택, "생산장비 객체화와 개방형 가공 셀 구축 연구(I) -생산장비 객체화-," 한국정밀공학회지, 제16권 제5호, pp.91-97, 1999.
  10. 김선호, 김동훈, 박경택, "생산장비 객체화와 개방형 가공 셀 구축 연구(II) -개방형 가공 셀 구축-," 한국정밀공학회지 제17권 제10호, pp.41-48, 2000.
  11. Dong-Hoon Kim, Sun-Ho Kim, Kwang-Sik Koh, "A Scheme on Internet-based Checking for Variant CNC Machine in Machine Shop", 2004 International Conference on Control, Automation, and Systems, pp. 1732-1737, 2004.
  12. Poslad S. J., Buckle S. J., and Hadingham R., "The FIPA-OS agent platform: Open source for open standards", Proceedings of PAAM 2000, Manchester UK, 2000
  13. Cantamessa, M., "Agent-based modeling and management of manufacturing systems," Computers in industry, Vo. 34, pp.173-186, 1997.
  14. Foundation for Intelligent Physical Agents, FIPA97 Specification Version 1.0 Part 1
  15. Foundation for Intelligent Physical Agents, FIPA97 Specification Version 1.0 Part 2 (section 5.2)
  16. Ross Mayne, Additions to CORBA on the Horizon - The Portable Object Adapter, Communicate, Volume 4 Issue 1, July 1998, pp 29-32