

분산, 개방, 지능형 제품정보관리시스템¹

김영호*, 강석호**, 이수홍***, 유상봉****

A Distributed, Open, Intelligent Product Data Management System

Yeongho Kim*, Suk-ho Kang**, Soo-hong Lee*** and Sang Bong Yoo****

ABSTRACT

An efficient management of product information is critical to the enhancement of corporate competitiveness. Product information is often distributed in terms of organization, geography, and time. With the rapid increase in the use of Internet or its associated Web and intranets, it has been accelerating to integrate the distributed information. In this research, a Distributed, Open, Intelligent Product Data Management system, namely DOI-PDM, is developed. This system deals with product information following the STEP standard, and its operation is controlled by a Web-based workflow management system. CORBA is employed to ensure interoperability among distributed objects. It is intended to achieve openness by using various standards, such as STEP, KQML, SGML, and WfMC specifications. One major difficulty in managing product information is that the information is changing dynamically, and hence the process managing it needs also be properly modified according to the dynamics. It however is very hard to have a prior anticipation to the modification requirements in practice. We introduce a process adaptation model and a use of intelligent agent technology. An agent is delegated to modify process definitions during run-time. A method of agent implementation and a proper communication scheme among the agents are proposed. Also developed in this research is a STEP-based content search system which can co-work with the agents by providing relevant product information that the agents require.

Key words : PDM, Workflow, Agent, STEP, Process Modification

1. 서 론

기업 경쟁력 제고를 위한 주요 선결과제 가운데 하나로 제품을 신속히 개발할 수 있는 환경을 구축하는 것을 꼽고 있다. 이를 위해서는 개발자들이 제품 정보를 원활하게 공유하는 것이 필수적이며, 더불어 이 정보를 통합 관리할 수 있어야 한다. 이 연구의 목적은 제품정보관리(PDM: Product Data Management) 시스템의 기본 기능에 분산, 개방, 지능형 개념을 강화하는 것으로, 이 논문에서는 이를 위한 시스템 구조와 구현 방안을 사례와 함께 논의하였다.

분산된 정보를 통합해서 관리하는 시스템에 대한 관심이 증가하고 있는데, 전형적인 예로 문서관리시스템이나 그룹웨어가 있다. 이들은 부분적으로는 이미 구축되어 활용되고 있는데, 이를 통해서 과거보다 획기적으로 업무생산성을 증대할 수 있었다는 사례도 발표되고 있다^{1,2)}. 그러나, 제품정보를 다루는 영역에서 이 같은 개념을 성공적으로 적용하기 위해서는 아직 해결해야 할 문제가 많은 것이 사실이다. 제품개발 프로세스는 다수의 제품개발 담당자들이 이질적(heterogeneous)이고 지역적으로 분산된(distributed) 컴퓨팅 환경에서 상이한 업무를 독립적(autonomous)으로 수행하면서도 상호 협력(cooperation)하는 공동작업(collaboration)이 증시되는 과정이다. 더욱이 제품정보를 표현하는 데이터의 형식이 매우 다양하고, 또 시스템마다 그 포맷이 달라 어려움이 배가되고 있다. 그러나, 최근의 데이터 통신기술 발달과 각종 네

*중심회원, 서울대학교 산업공학과

**서울대학교 산업공학과

***중심회원, 연세대학교 기계공학과

****중심회원, 인하대학교 자동화공학과

¹본 연구는 한국과학재단 특정기초연구비 (97-02-00-09-01-3) 지원으로 수행되었으며 지원에 감사를 드립니다.

이터 표준 및 활용에 대한 연구가 활성화되면서, 이러한 어려움을 해결할 수 있는 길을 열어주고 있는 것 또한 사실이다.

한편, 정보기술의 발달은 과거 단순 데이터 교환을 넘어 지식 공유를 지원하는 방향으로 전개되고 있다. 이는 컴퓨터가 인간을 대신해서 수행할 수 있는 일의 범위가 확대되어온 지금까지의 발전 과정을 살펴볼 때 아주 자연스러운 방향이라고 할 수 있지만, 초기 인공지능 연구자들이 구상했던 완벽한 지능을 가진 시스템이 실용화되기에는 아직 미흡하다. 그러나 잘 정의되어 있는 범위 내에서는 소위 전문가시스템이라고 하는 어느 정도 제한된 지능적 활동을 할 수 있는 시스템이 실제로 활용되고 있다. 최근에는 이러한 시스템에 통신 기능을 첨가함으로써 분산된 지식을 활용하는 방안에 대해 연구가 다각적으로 진행되고 있다. 이런 지능적 시스템에 대한 연구의 결과를 제품개발에 응용하여 인간의 노력을 대체하고자 하는 노력 또한 활발히 진행되고 있다.

본 연구에서 개발하는 시스템의 요구사항을 정리하면 다음과 같다.

- 1) 개방형: 시스템간 원활한 데이터 공유를 위해 개방형 데이터 표준 사용
- 2) 분산형: 분산체제가 제공하는 서비스간의 상호운용성(interoperability)
- 3) 프로세스 관리: 절차에 따라 진행되는 업무 프로세스 계획, 통제, 관리
- 4) 지능형: 분산된 지식 자원을 활용하도록 도와주는 지능을 갖춘 시스템

먼저, 개방형 표준으로 본 연구에서는 제품정보 교환을 위한 국제표준인 STEP(STandard for Exchange of Product Model data)을 활용한다. 그리고 OMG(Object Management Group)에서 제시하는 분산객체 환경 규약인 CORBA(Common Object Request Broker Architecture)를 채택하여 분산객체 환경을 실현한다. 또, 제품개발 과정에서 정보가 생성, 배포, 사용되는 프로세스를 관리하기 위한 워크플로우 관리시스템(Workflow Management System)을 개발한다. 이 또한 개방성을 높이기 위해 국제표준규약인 WfMC(Workflow Management Coalition)의 레퍼런스(reference) 모델을 기반으로 개발한다. 한편, 구성원간의 효과적 지식공유를 위해 지능적 에이전트를 활용하고, 동적으로 변화하는 에이전트에 적합한 에이전트 통신 구조를 제안한다. 이는 업무 진행 상황이나 중간 결과에 따라 사용자의 개입을 최소화하면서 프로세스의 흐름을 지능적으로 통제할 수 있게 한다. 마지막으로

편리한 사용자 인터페이스 또한 중요한 요구사항으로 3차원 모델과 문자 정보의 효과적인 시각화를 위해서 각각 VRML(Virtual Reality Modeling Language)과 SGML(Standard Generalized Markup Language) 표준을 이용하여 시스템을 개발하였으나 이는 이 논문의 범위 밖이므로 논의를 생략하였다.

분산, 개방, 지능형 제품정보관리(DOI-PDM: Distributed Open Intelligent Product Data Management)시스템은 여러 가지 장점을 제공한다. 첫째, 개방형 데이터 표준을 지원하므로 플랫폼 독립적인 정보 공유가 가능하다. 둘째, 분산된 제품 정보에 효과적으로 접근하고, 이를 재사용하는 것을 지원한다. 셋째, 워크플로우 시스템을 이용하여 업무와 정보의 투명성을 체계적으로 관리하고 그 흐름을 정확히 통제한다. 넷째, 업무처리 중간 결과에 따라 시스템이 지능적으로 그 흐름을 적절히 통제한다. 다섯째, 정보 및 지식 공유와 공동 작업을 지원하여 개발자들간의 의사교환을 신속히 할 수 있다. 결국 이 같은 장점들은 전체 제품 개발 과정을 단축시켜 개발 생산성을 높일 수 있고, 이는 기업 경쟁력 제고와 연결될 수 있을 것이다.

2. 시스템 구조 및 분산 환경

먼저 시스템의 구조와 기능을 전체적으로 조명하고, 분산시스템 구현의 근간을 이루는 객체환경 구현에 대해 개괄적으로 설명한다

2.1 시스템 구조와 기능

Fig. 1은 시스템의 전체 구조이다. 시스템은 개념적으로 정보공유 기반구조와 통합 업무수행환경의 두 부분으로 구성되어 있다. 정보공유 기반구조는 제품정보를 저장하는 데이터베이스를 구축하는 것과 표준데이터를 응용하고 네트워크 상에서 정보를 공유하기 위해 분산객체 관리환경을 구현하는 것을 포함한다. 이 연구에서는 OMG의 CORBA를 채택하였는데, 이에 대해서는 2.2절에서 알아본다. 그리고, 데이터 표준으로는 STEP과 KQML(Knowledge Query and Manipulation Language)을 사용하였다.

통합 업무수행환경은 위의 기반구조를 바탕으로 실제로 제품정보를 공유하고 처리하는 응용시스템들을 말한다. 이는 다음 네 개의 기능으로 구성되어 있다.

- 표준데이터 인터페이스
- 워크플로우 관리
- 지능적 에이전트

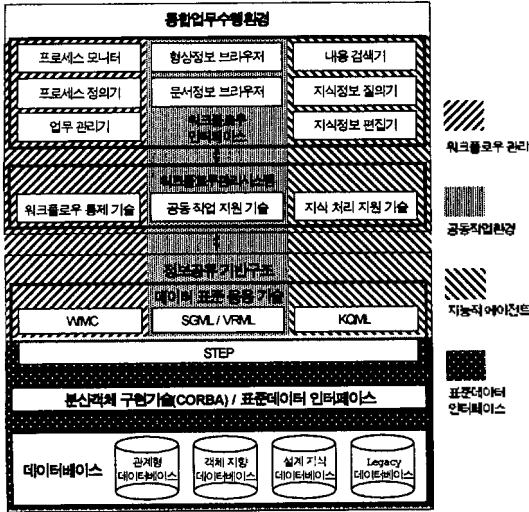


Fig. 1. System architecture.

• 공동작업 환경

표준데이터 인터페이스 기능은 분산이종의 데이터베이스와 지식베이스를 통합 관리하고, 데이터 저장과 접근 인터페이스를 제공하는데, 이는 다른 세 기능이 공통으로 요구하는 데이터 서비스이므로 기반구조에 포함시켰다.

워크플로우 관리는 업무의 절차와 이 절차 상에서 통제되어야 하는 정보의 흐름을 관리하는 기능을 수행한다. 제품개발 프로세스의 특성을 고려하여 이에 적합한 시스템을 설계하고 구현하였다. 제품개발 프로세스는 프로세스 진행 도중에 업무흐름이 동적으로 변하는 경우가 자주 발생하는데, 이때 지능적 에이전트 시스템이 개입하여 상황에 적절한 프로세스 진행 방안을 도출하여 워크플로우관리시스템에 제공하고 지능적으로 프로세스를 통제한다. 특히, 이 연구에서는 표준데이터 인터페이스의 내용검색을 통한 향상된 정보검색 서비스에 중점을 두고 있다. 이는 워크플로우 관리, 지능적 에이전트, 공동작업 환경의 사용자가 제품 또는 부품의 특성을 이용하여 다양하게 제품정보를 검색하는 것을 말한다.

공동작업 환경은 사용자가 제품 개발 과정에서 이용하는 여러 가지 응용시스템과 PDM의 단위 기능을 말한다. 이 논문의 주목적은 분산환경에서 정보를 개방적으로 교환하고, 프로세스를 지능적으로 통제하는데 있으므로, 공동작업 환경에 대해서는 정보를 시각화하는 일부 기능만을 고려한다. 일단 사용자에게 정보가 전달되면 이 정보를 가공하고 이용하는 것은 응용시스템에 따라 다르고, 이는 이 논문의 범위를 벗

어난다.

통합 업무수행환경에서도 개방성을 높이기 위하여 표준을 적용하고 있는데, 워크플로우관리시스템은 WIMC의 참조모델을 준수하였으며, 서로 다른 에이전트 간의 통신을 위한 메시지 프로토콜로는 KQML을 사용하였다. 그리고 시각화를 위한 데이터 표준으로는 SGML과 VRML을 채택하였는데 이들은 각각 문서정보와 형상정보에 대한 ISO 표준[5, 6, 7, 8, 9, 10]으로 이를 이용하여 사용자는 웹 상에서 정보를 브라우저할 수 있다.

2.2 분산객체환경 구현

DOI-PDM은 분산환경에서 운영된다. 분산 시스템은 다수의 개별적인 컴퓨터들이 분산 시스템 소프트웨어와 함께 컴퓨터 네트워크를 통해 연결되어 정보 자원을 서로 공유할 수 있는 시스템을 의미한다. 이는 사용자에게 지역적으로 분리되어 있는 여러 장소의 컴퓨팅 장비와 정보 자원들을 마치 하나의 통합된 시스템에서 운영하고 처리하는 듯한 투명성을 제공한다. 분산환경을 구현한다는 것은 이러한 이기종 하드웨어와 소프트웨어간의 호환성을 어떻게 제공하느냐 하는 것이다.

최근 분산된 시스템의 구성요소를 각기 객체로 인식하는 분산객체환경에 대한 연구가 활발히 진행되고 있다. 분산객체환경을 실현하기 위한 기술로는 CORBA와 DCOM (Distributed Component Object Model)이 있다. 이중 OMG에서 제시하는 분산객체환경 규약인 CORBA가 공개 표준으로서의 장점이 있으므로 본 연구에서도 이를 채택하기로 하였다. CORBA에 대한 기술적 내용은 참고문헌에서 구할 수 있으며, 여기서는 CORBA를 활용하는 일반적인 이유 또는 효과를 아래와 같이 정리하였다^[13,18].

• CORBA는 플랫폼, 운영체제, 네트워크 프로토콜, 프로그래밍 언어 간의 투명성(transparency)을 제공한다. 즉, 표준 인터페이스(standard interface)를 통해 분산된 여러 객체가 제공하는 서비스 간의 상호운용성을 확보하는 것이 가능하다.

• IDL(Interface Definition Language)을 통해 인터페이스를 공유하므로 클라이언트와 서버를 독립적으로 개발할 수 있고, 이는 요소 기반의 소프트웨어 개발을 가능하게 한다.

• 캡슐화(encapsulation), 상속성(inheritance), 다형성(polymorphism) 등 객체지향기술의 특징을 그대로 이어 받아 소프트웨어의 견고한 설계(robust design), 간편한 유지보수, 용이한 확장, 코드 재활용 등의 이

점을 살릴 수 있다.

• 각기 다른 프로그래밍 언어로 작성된 실행 객체가 IDL에 정의된 표준 인터페이스를 통해서 서비스를 제공하는 객체에 접근할 수 있으므로 레거시(legacy) 데이터나 레거시 소프트웨어와의 통합도 용이하다.

이 연구에서 개발하는 PDM은 그 구성요소가 서로 다르고 또 분산되어 있는 전형적인 분산시스템이다. 컴퓨터와 네트워크 같은 서로 다른 하드웨어 및 운영체제가 사용된다. 상이한 언어로 개발되고 다른 데이터 포맷을 가지는 서로 다른 소프트웨어가 각기 필요에 따라 도입되어 독립적으로 운영되는 일은 더욱 흔하다. 한편, 비즈니스 자체가 계속 변화하므로 이를 지원하는 응용소프트웨어도 지속적으로 변화해야 한다. CORBA를 이용하면 변화될 필요가 있는 특정 부분만을 갱신하는 것이 가능하다. 이 연구에서도 CORBA의 장점을 활용하여 앞 절의 시스템 기능들을 독립적으로 그리고 동시병렬적으로 설계하고 구현할 수 있었으며, 개발된 시스템의 통합 또한 용이하였다.

3. 설계 프로세스의 특징과 사례

PDM을 가장 포괄적으로 정의하면, 이는 제품과 관련된 모든 정보의 생성 및 접근을 통제하여 전체 제품수명주기 상에서 정보 공유를 원활히 하고, 또 그 흐름을 효율적으로 관리하는 소프트웨어이다. 이 시스템은 일반적으로 워크플로우 관리, 문서/도면 관리, 제품구조관리, 형상관리, 공동작업관리(groupware) 등을

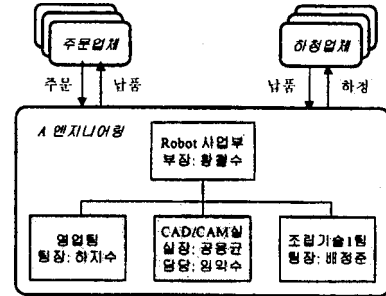


Fig. 2. Organization of engineering company A.

기본기능으로 들고 있다^[4]. 이런 기능들은 대부분 이미 단위 시스템으로도 개발되어 독립적으로 이용되고 있으므로 PDM은 이들을 통합한 것이라고도 볼 수 있다. 이 연구에서는 PDM의 기본기능에 다음과 같은 향상된 서비스를 부가함으로써 분산, 개방, 지능 개념을 강화한다.

- 프로세스 실행 시의 동적 변경 기능
- 에이전트를 활용한 PDM 기능의 자동화
- 내용검색을 통한 효과적 설계정보 탐색

먼저 설계 프로세스 사례와 PDM 사용 시나리오를 정리하였는데, 이는 DOI-PDM의 향상된 기능을 설명하기 위함이다.

3.1 설계 시나리오

Fig. 2는 A엔지니어링사 로봇 사업부의 조직도이다. 이 회사는 전형적인 수주생산을 하며, 제품별로

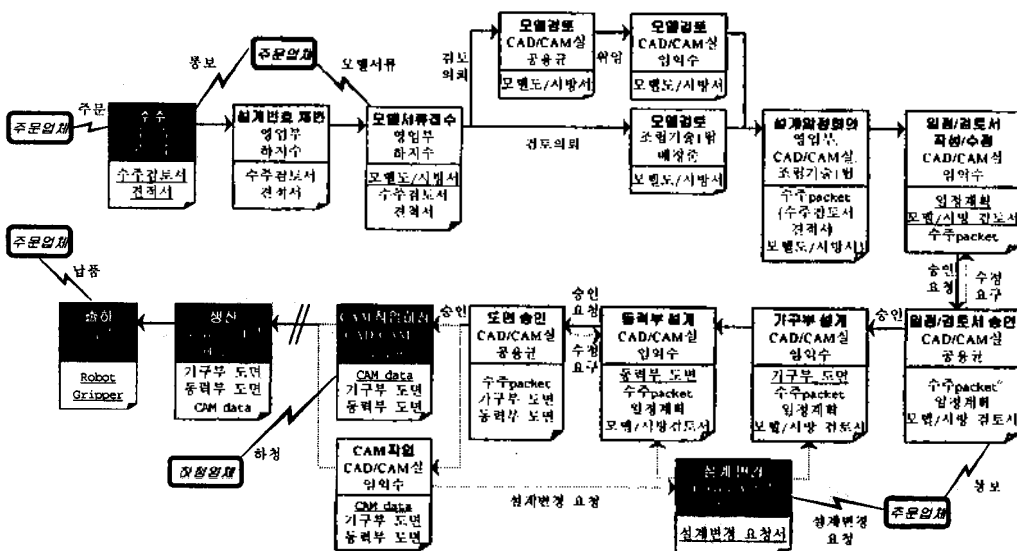


Fig. 3. A scenario of product development process.

몇 개의 사업부가 있는데, 로봇 사업부는 로봇 그리퍼(gripper)를 외부에서 주문 받아, 설계, 생산한다. 업무부하에 따라 업무의 일부를 전문 하청업체에 외주를 주기도 한다. 이 사업부는 영업팀, CAD/CAM실, 그리고 조립기술1팀으로 구성되어 있는데, 영업팀은 수주와 납품을, CAD/CAM실은 설계 및CAM 데이터 생성을, 조립기술1팀은 그리퍼 생산을 담당한다.

Fig. 3은 A사에서 그리퍼를 수주하여 납품하기까지 발생하는 일련의 과정 가운데 신규설계를 위한 한 가지 프로세스 예를 보여주고 있다. 이 프로세스는 PDM을 효율적으로 이용하기 위하여 과거의 프로세스를 변경한 것이다. 프로세스는 순차적으로 진행되는 업무들로 구성되어 있으며, 각 업무는 업무이름, 담당 부서, 담당자, 그리고 업무에서 생성하는 문서와 참조하는 문서로 표시되어 있다. 업무는 크게 단위업무와 복합업무 두 가지로 분류되어 있는데, 단위업무는 더 이상 나눌 수 없는 최소 단위의 업무이고, 복합업무(회색으로 표시된 업무)는 몇 개의 단위업무로 구성된 별도의 프로세스를 간단히 표시한 것이다.

업무를 연결하는 화살표는 업무진행 순서 또는 정보의 흐름을 나타낸다. 실선은 반드시 수행되어야 할 일을 나타내고, 점선은 경우에 따라서는 수행할 필요가 없는 것을 나타낸다. 그리고 주문업체나 하청업체 같은 회사 외부와의 정보교환은 꺾은 선으로 표시하였다.

3.2 PDM 기능과 사용 시나리오

Fig. 4는 Fig. 3의 설계 프로세스를 지원하는 DOI-

PDM의 사용 시나리오를 보여준다. 바깥쪽 층은 Fig. 3의 프로세스를 간단히 표시하고 있다. 가운데 층에는 DOI-PDM이 제공하는 기능을 이를 주로 사용하는 업무와 연결하여 나타내었다. 앞에서 언급한 문서/도면 관리, 형상관리, 공동작업관리 등의 기본기능을 찾을 수 있다. 그리고 지능화를 위한 에이전트 기능들은 회색으로 강조하여 표시하였다. 이런 기능들을 이용하여 수행되는 업무와 업무에서 생성, 사용하는 정보는 워크플로우관리시스템에 의해 통합적으로 관리된다. 마지막으로 데이터 또는 지식의 저장소가 그림의 가장 안 쪽 층에 표시되어 있다. A사의 설계 프로세스는 기본기능과 향상된 서비스를 이용하여 다음과 같이 진행된다.

주문업체의 주문은 영업팀에서 접수하여 일정한 수주처리 과정을 거친다. 이 프로세스는 ERP 시스템을 이용하여 처리되며, 다만 설계에 필요한 몇 가지 데이터만을 공유한다. 최종 확정된 수주 조건은 바로 주문업체에 통보되며, 이때 주문업체는 설계에 필요한 구체적 사양을 포함한 모델서류를 제출한다. 이는 모두 웹에서 이루어지며, 특히 모델서류의 설계정보는 STEP 표준을 따르므로 A사는 이를 바로 활용할 수 있다. 모델서류는 문서관리 기능을 이용하여 저장된다.

수주가 완료되면 설계 프로세스가 시작된다. 가장 먼저 설계번호를 부여하고, 동시에 신규설계와 수정설계 가운데 한 가지 프로세스를 결정한다. 설계번호는 향후 설계 업무에서 생성되는 각종 데이터를 참조하기 위한 중요한 키(key)로 활용된다. 이 회사는 그룹

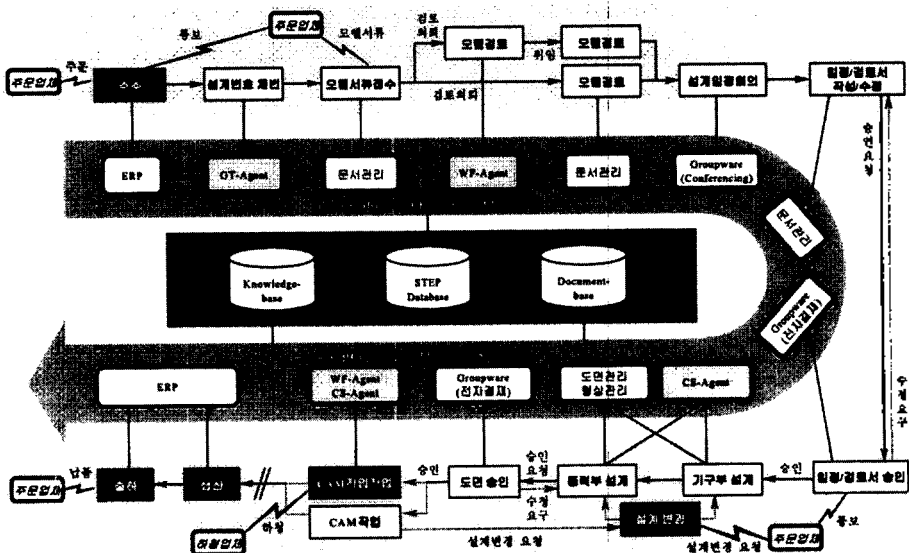


Fig. 4. DOI-PDM scenario.

테크놀러지(group technology) 개념을 적용하며, 이를 위한 GT 전문가시스템이 있다. 이때 에이전트는 전문가시스템과 함께 수주 시 받은 제품정보를 이용하여 신규설계 혹은 수정설계를 결정한다. 워크플로우관리시스템은 에이전트가 추천하는 프로세스대로 진행하게 된다. 여기서는 신규설계를 하기로 결정했다고 하자.

접수된 모델서류는 설계번호와 함께 이를 검토해야 하는 CAD/CAM실 및 조립기술 1팀의 담당자에게 전달된다. 검토의뢰를 받은 CAD/CAM실 부장은 앞으로 제품설계를 담당하게 될 담당자를 선정하여 위임한다. 이는 부장이 직접 할 수도 있지만, 워크플로우관리시스템과 연동하여 운영되는 WF-에이전트가 결정할 수도 있다. WF-에이전트는 설계할 제품의 종류와 설계실 인력의 설계일정 및 업무부하를 고려하여 적절한 담당자를 선정하고, 설계번호를 포함한 검토의뢰문을 전달한다. 담당자는 전달된 설계번호를 이용하여 문서관리시스템에서 검토할 서류를 쉽게 검색할 수 있다.

모델검토가 완료되면 개략적인 향후 설계일정계획을 수립한다. 과거에는 담당자들이 모두 모여 일정을 협의하였지만, 지금은 네트워크 컨퍼런싱(conferencing) 기능으로 편리하게 협의할 수 있다. 회의에서 확정된 일정계획과 검토결과서가 작성되어 검토서류와 함께 문서관리시스템에 저장되며, 이는 CAD/CAM실장의 승인을 받아야 한다. 실장의 승인이 나면 바로 일정에 따라 설계를 시작하지만 수정요구가 있을 때는 이 과정을 반복한다. 이 과정은 전자결재로 이루어 진다.

한편, 설계가 진행되는 도중에 주문업체, CAM 작업, 외주업체 등으로부터 여러 가지 이유로 설계변경 요청이 있을 수 있다. 이 요청은 별도의 프로세스를 거치는데, 변경이 확정되면, 해당되는 부품의 재설계가 이루어 진다. 이때 형상관리 기능은 설계이력을 관리한다. 이 시나리오에서는 설계가 막바지에 이르렀을 때 주문업체로부터 설계변경요청이 있었고, 이를 반영하기로 결정하였다고 하자. 이제 원래 설계프로세스를 따르면 납기를 맞출 수 없게 되었다. 다행히 이 회사는 변경된 설계사양과 비슷한 제품을 과거에 설계한 적이 있다. 따라서 설계자는 도면관리 기능을 이용하여 비슷한 과거 도면을 검색하고 이를 참조하여 약간의 수정만으로도 설계를 마칠 수 있는 수정설계 프로세스를 적용할 수 있으므로 설계기간을 단축할 수 있다. 이때에도 에이전트를 통해 내용검색을 하여 과거 도면을 효율적으로 검색할 수 있다. 뿐만 아니라 기구부와 동력부 설계는 동시병렬적으로도 진행할 수 있으므로 역시 설계기간을 단축할 수 있다. 마지막으

로 기구부 설계를 두 부분으로 나누어 두 명의 설계자가 작업하여 설계기간을 단축할 수도 있다. 이 같이 설계기간 단축을 고려하여 새로운 설계프로세스가 결정되는데 이는 역시 에이전트를 통하여 지능적으로 결정된다.

승인된 도면에 대해 조립기술1팀에서 사용하게 될 CAM 작업을 한다. 이 작업은 내부에서 할 수도 있고, 제품이나 업무 부하에 따라 전문 CAM 업체에 용역을 줄 수도 있다. 내부 CAM 작업의 경우 다시 에이전트를 활용하여 과거 비슷한 CAM 작업 결과를 활용하며, 외주를 줄 경우에는 별도의 외주 프로세스를 거친다. CAM 작업 완료는 생산 및 출하 프로세스와 연결되는데 이는 다시 ERP 시스템에 의해서 관리된다.

4. 워크플로우관리시스템

워크플로우관리시스템은 정보 및 문서 전달의 전자화, 그리고 일관적 데이터 접근 및 제어를 통해서 업무 프로세스의 자동화된 관리를 지원하여 프로세스 개선, 통제, 관리, 공동작업을 지원하는 소프트웨어 시스템을 일컫는다^[2,12,16]. 이 시스템은 업무 환경이 자동화, 전자화 및 네트워크화 됨에 따라 그 중요성이 더욱 부각되고 있다. DOI-PDM이 추구하는 궁극적 목표는 제품개발을 위한 효과적 정보 자원 공유이고, 이 정보 공유는 업무의 흐름 상에서 발생한다. 따라서 워크플로우관리시스템은 DOI-PDM의 기반이라고 할 수 있다.

이 절에서는 제품개발 프로세스를 지원하는 관점에서 기존의 워크플로우관리시스템이 가지는 한계를 해결하기 위한 방안과 여기에 필수적으로 활용되는 에이전트와의 인터페이스에 대해 중점적으로 설명할 것이다.

4.1 워크플로우관리시스템과 제품개발 프로세스

워크플로우관리시스템에 대한 연구는^[15,17,19]를 포함하여 많이 있고, 상용 시스템도 다수 있다. 이들은 주로 프로세스 주기가 상대적으로 짧고, 자동화된 업무가 많고, 매우 정형화된 업무를 다루는 것을 목적으로 개발되었다. 그러나 제품개발 프로세스를 지원하는 관점에서 보면, 아직 기술적으로 해결해야 할 몇 가지 문제가 있다. 자주 언급되는 것으로 동적 프로세스 변경, 에러 처리 및 복구, 프로세스 표현력 강화, 서로 다른 워크플로우시스템 간의 상호운영성 등이 그것이다.

제품개발 프로세스는 일반적으로 몇 주 혹은 몇 달 동안 지속되는 장기적인 프로세스로 프로세스를 사전에 명확히 정의하기 어렵고, 또 이를 실행 시에 동적으로 변경해야 할 필요성이 높다는 특성이 있다. 예를 들어 전 절의 시나리오에서 프로세스가 진행되는 도중에 내 외부의 요인에 의해 프로세스 자체가 수정되어야 하는 경우가 여기에 해당된다. 이 경우 프로세스 실행 시 어느 시점에서는 반드시 변경된 사항이 반영되어야 한다.

이와 같은 임시적 혹은 임의적(ad-hoc) 변경 요구에 적절히 대처할 수 있는 적응성(adaptability)은 대다수 워크플로우관리시스템이 안고 있는 문제로 지적되고 있다. 이는 기존의 방법이 한 번 정의된 프로세스가 완료될 때까지 변화가 없다고 가정하고, 사전에 정의된 루틴을 따라 모든 작업흐름이 결정되는 확정 프로세스(rigid process)모델을 사용하기 때문인데, 일반적으로 이 모델로는 변화에 유연하게 대응하는 것이 근본적으로 불가능하다.

4.2 동적 프로세스 변경

실행 시 프로세스를 동적으로 변경할 수 있다면 이를 통해 프로세스의 적응성을 높일 수 있을 것이다. 프로세스 적응성이란 프로세스를 수행하는 도중에 외부 환경 변화 또는 내부 업무 결과에 따라 이미 계획된 업무 프로세스를 변경할 수 있는 기능을 말한다. 먼저 프로세스 적응성이 요구되는 상황을 분류하여 각각에 대해 적응성 모델을 생각해볼 필요가 있다.

4.2.1. 적응성 모델의 분류

프로세스 적응성을 이해하기 위해서는 Meijler 등¹¹⁾이 제시한 여섯 가지 분류가 매우 유용한데, 이들을 제품개발과의 관련성을 고려하여 수정한 것을 다음에 설명하였다.

1) 임의프로세스 모델 (ad-hoc process model)

프로세스와 이를 구성하는 일련의 작업들이 임의적이어서 사용자가 다음 작업을 선택하는데 완전히 자유로운 모델을 말한다. 이 모델은 확정모델이 가지는 모든 제약으로부터 벗어날 수 있어 적응성을 최대한 보장할 수는 있으나, 이것을 일반적으로 가장 좋은 해결책이라고 보지는 않는다.

2) 선택제한 모델 (restricted choice model)

사용자가 정의할 수 있는 작업의 선택이 상황 의존적이고, 제한적인 모델을 임의프로세스 모델과 맞서는 선택제한 모델이라고 한다. 선택할 수 있는 후속 업무나 프로세스를 사전에 템플릿(template) 형태로 정의하고, 변화에 대응하기 위해 이 중에서 하나를

선택할 수 있다. 다음의 구조기반 모델, 목표지향 모델, 내용기반 모델은 선택제한 모델의 확장된 형태라고 볼 수 있다.

3) 구조기반 모델 (structure-based model)

구조기반 모델은 프로세스 정의가 프로세스를 따라 처리되는 업무 객체(object)-여기서는 제품의 구조에 의존적인 모델을 말한다. 예를 들어, 그리퍼는 구조적으로 기구부와 동력부로 나누어져 있으므로 설계도 이 특징을 반영하여 진행된다. 또, 로봇트를 설계하는 프로세스는 그리퍼보다는 복잡하며 로봇트 구조에 매우 의존적이다.

4) 목표지향 모델 (goal-oriented model)

목표지향 모델은 프로세스의 최종 혹은 부분적 목표를 달성하는 것을 고려하여 최적의 프로세스를 설계하는 모델을 말한다. 이 경우 최적 프로세스를 계산하는 추론 기능이 필요하며, 이는 사용자가 직접 할 수도 있고 또 지능적 에이전트에게 위임할 수도 있다. 예를 들어, 설계변경으로 납품기한을 지키는 것이 어려워질 수 있다. 이때, 납기를 준수할 수 있는 새로운 프로세스 대안을 생성하고 예상 완료시점을 사전에 검토하여, 최적의 프로세스를 사용할 수 있을 것이다.

5) 내용기반 모델 (content-based model)

업무 중간 결과의 내용에 따라 후속 프로세스가 결정되는 모델을 말한다. 이 모델은 위 두 모델보다 상대적으로 간단한데, 예상되는 모든 중간결과를 미리 예측할 수 있다면, 사전에 프로세스를 모델링 해 둘 수 있고, 실행 시에 적합한 프로세스를 선정할 수 있기 때문이다. 그러나 예측하지 못한 결과에 대해서는 임의프로세스 모델을 이용하여 후속 프로세스를 새로 설계해야 한다.

6) 업무레벨변경모델 (task-level refinement model)

프로세스는 사전에 정의되지만 업무레벨의 세부적 내용은 그렇지 않거나 일부만 정의할 수 있고 이를 실행 시에 결정 또는 수정할 수 있는 모델을 말한다. 다른 모델은 모두 프로세스 자체가 변경되어야 하지만 이 모델은 프로세스를 구성하는 업무에 대한 정의만 변경하는 것을 다룬다. 예를 들어 특정 업무를 수행할 담당자나 하청업체를 프로세스 설계 시 결정할 필요도 없고 또 실제로도 결정할 수 없는 경우 이 모델이 적당하다.

이 연구에서는 목표지향 모델, 구조기반 모델, 내용기반 모델로 나누어 지는 제한적 선택 모델을 고려하기로 하는데, 일반적으로 기업은 납입기한, 부품구조, 설계사양 등에 있어서 완전히 자유롭지 않아서 제약

조건 없이 임의적으로 프로세스를 선택하는 경우는 거의 없기 때문이다.

4.2.2. 에이전트를 이용한 적응성 실현

지능적 에이전트는 적응성 모델에 기반하여 실행 시에 발생하는 프로세스 변경을 처리한다. 지능적 에이전트를 이용한 동적 프로세스 변경은 기존 시스템이 갖지 못한 새로운 기능으로 매우 큰 의의가 있다고 하겠다. 이 연구에서 고려하고 있는 에이전트와 그들의 역할은 다음과 같다.

- EC(Engineering Change) 에이전트: 설계 변경 요청을 감지하고, 설계변경 요인을 파악한다.
- WF(WorkFlow) 에이전트: 워크플로우 시스템과 사용자 혹은 다른 시스템과의 상호작용을 중재하며, 프로세스 변경 사항을 워크플로우 엔진에 반영한다.
- GT(Group Technology) 에이전트: GT 개념을 이용하는 전문가 시스템과 함께 과거 유사 설계 정보를 추출하고, 사용자의 정책 또는 제약조건을 고려하여 신규설계 혹은 수정설계를 결정하여 해당 프로세스를 추천한다.
- SM(Schedule Management) 에이전트: 일정계획 시스템 또는 그룹웨어 시스템과 다른 시스템간의 메시지 교환을 중재하면서 프로세스의 일정을 조정한다.
- CS(Content Search) 에이전트: 내용검색 시스템과 다른 시스템간의 상호작용을 중재하고, 다른 에이전트의 요청이 있을 때 도면, 문서, 설계이력 등 내용 기반검색의 결과를 반환한다.
- PA(Process Adaptation) 에이전트: 프로세스 변경이 필요할 때 해당 적응성 모델을 선정한다.

이들 에이전트는 자율성을 가지는 지능형 에이전트(intelligent agent)로 각각 독립적인 영역을 가지지만 서로 상호 작용한다. 프로세스의 동적 변경은 에이전트 간의 통신을 통한 자율적, 지능적 작용에 의해 구현되는데, 다음은 각 에이전트가 프로세스 변경을 처리하는 절차에 대한 설명이다.

먼저 다음과 같은 상황을 가정하자. 워크플로우 시스템이 신규설계 작업을 관리하는 중에 있고, 도면 작업이 완료되어 현재 CAM작업이 시작되었다. 이때 주문업체로부터 설계변경 요청이 접수되었고 이를 수용하기로 하였다고 하자.

처리 절차와 사용되는 에이전트는 Fig. 5와 같다. 먼저 EC 에이전트는 변경 요구를 수용하기로 결정한 사용자의 입력으로 활성화된다. EC 에이전트로부터 변경 요인을 전달 받은 GT 에이전트는 신규설계를 할지 수정설계를 할지 결정한다.

여기서 CS 에이전트가 개입하여 설계이력을 제공하

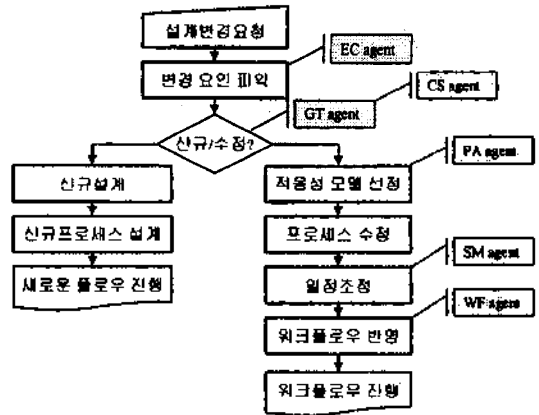


Fig. 5. Agent involvement in process adaptation.

고, 이를 참조하여 의사결정을 내린다. 수정 설계의 경우 진행 중인 프로세스를 설계변경 요청에 따라 수정해야 한다. 이때 PA 에이전트는 적절한 적응성 모델을 선정한다. 수정된 프로세스가 남기에 영향을 미칠 때 이를 조정하기 위해 SM 에이전트가 일정을 조정한다. 이러한 일련의 과정이 성공적으로 수행되면 WF 에이전트는 이를 워크플로우의 진행에 반영하여 프로세스를 수정한다.

4.3 시스템 구조와 구현

워크플로우관리시스템과 동적 프로세스 변경을 지원 하는 모듈은 크게 프로세스 디자이너, 엔진, 클라이언트, 모니터. 에이전트의 다섯 가지 모듈로 구성되는데, 에이전트 모듈을 제외하면 WfMC의 다섯 가지 표준 인터페이스를 충실히 따르고 있다. 시스템의 구조는 Fig. 6과 같다.

클라이언트, 모니터, 에이전트는 웹 환경에 맞추어 자바 애플릿(applet)으로 구현하였으므로 사용자가 손

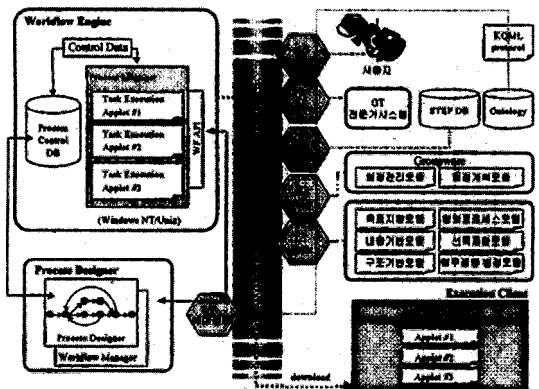


Fig. 6. Architecture of work flow management system.

쉽게 다운로드 받아 이용할 수 있다. 또, 요소기반 모델링 기법을 따라 개발하였으므로 작업자 환경과 요구사항에 맞는 메뉴를 바로 구성할 수 있다. 에이전트의 구현에 대해서는 다음 장에 보다 더 구체적으로 설명하였다.

5. 지능적 에이전트

에이전트는 나름대로의 지식베이스와 추론기능이 있어 상황에 적절한 프로세스 진행 방안을 도출하고, 워크플로우관리시스템은 이를 이용하여 지능적으로 프로세스를 통제한다. 또, 에이전트는 업무환경 변화를 감지하고, 이에 대해 능동적으로 대처하며, 필요 시 다른 에이전트들과의 협력한다.

5.1 에이전트의 특징

에이전트는 다음과 같은 특징을 가진다. 우선, 에이전트는 일종의 전문가 시스템이므로 전문 영역을 가진다. 문제를 해결하는 동안 자신의 영역과 관련되지 않은 지식이 요구될 때는 그 지식에 전문적인 다른 에이전트에게 정보를 요청한다.

둘째, 목표 달성을 위한 여러 대안들 중 적절한 것을 선정할 수 있다. 에이전트는 전체 목표 달성을 위해 작업을 부분으로 나누어 수행하며, 이들을 조합하여 전체 목표를 이루어 낸다. 각 에이전트는 자신의 의사결정 알고리즘이나 외부 의사결정지원시스템의 도움으로 적절한 대안을 선정한다.

셋째, 에이전트가 수행한 결과는 다른 에이전트에 입력되는데 이를 통해 외부세계에 대한 인식, 내부 상황 고려, 활동 대안 선정, 실행 결과 인식 등으로 이어지는 일련의 작업이 순환 반복된다. 에이전트의 개념과 특징을 이와 같이 규정할 때에 에이전트 기반 기술은 전통적인 공정계획 프로그램이나 의사결정 시스템과는 달리 실시간 프로그래밍에 가까운 것임을 알 수 있다^{123,24)}.

마지막으로 외부 세계와 인터페이스가 가능하다. 즉, 에이전트 간의 통신이 가능해야 한다. 에이전트는 자신의 영역에서 지능적 활동을 수행하며, 필요한 지식 혹은 자원을 사용하기 위해서 다른 에이전트의 도움을 받는다.

5.2 적응성을 위한 에이전트간의 관계

앞에서 설명한 에이전트들은 상호작용 한다. 이러한 상호작용은 통일된 ontology를 공유하고, KQML이라는 공통의 규약으로 서로 통신함으로써 이루어질 수

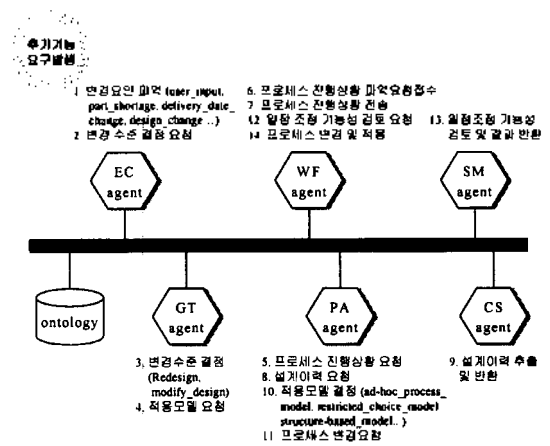


Fig. 7. Agent interactions.

있다. 이들간의 구체적인 관계는 Fig. 7과 같다. 그림의 번호 순서는 각 에이전트가 자신의 역할을 수행하는 한 가지 예를 보여 주고 있다.

에이전트들은 KQML 규약의 “performative”를 통해서 메시지의 의도를 전달하며, 실제로 수행해야 할 일들은 Content Layer에 실어서 보낸다. 에이전트가 사용할 수 있는 “performative”에는 다음과 같은 것들이 있다.

- 기본 질의: evaluate, ask-if, ask-in, ask-one, ask-all
- 반응: reply, sorry
- 일반 정보: (un)tell, (un)achieve, cancel
- 능력 정의: advertise, monitor, import, export
- 네트워크: (un)register, forward, broadcast, route

EC 에이전트가 GT 에이전트에게 수정 설계를 할 것인지 재설계를 할 것인지를 의뢰하는 KQML performative 예를 들면 Fig. 8과 같다.

예에서 보는 바와 같이 에이전트의 “Content Layer” 언어를 IDL로 하였는데, 대부분 연구에서는 이를 KIF, PROLOG와 같은 비절차적 언어를 사용한다. 이런 언어들은 메시지에 담고자 하는 내용의 표현이 자유로운 장점은 있으나, 이를 해석하기 위해 KIF 분석기 등 중간 처리 과정이 필요하고 절차적 언어로

```
ask
: content    "Redesign_or_modify
              (DESING_CHANGE,SC1203, Heo)"
: language  IDL
: ontology   Robot_Design
: from       EC_agent
: to         GT_agent
```

Fig. 8. KQML message example.

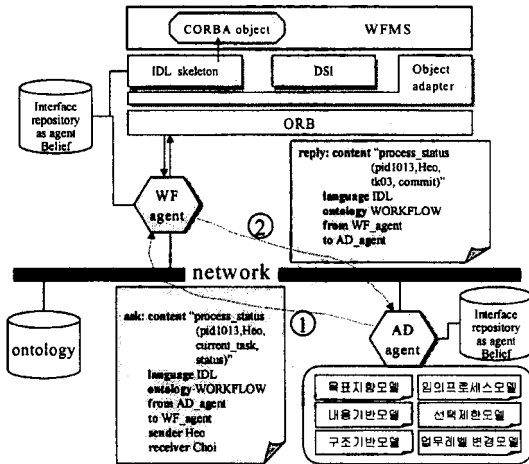


Fig. 9. KQML message exchange procedure.

구현된 시스템과 통합이 어려운 점이 있다. 본 연구에서는 IDL을 이용하였으므로, CORBA, JAVA RMI, MS DCOM 등과 IDL을 통해 인터페이스를 정의하는 시스템과의 통합이 용이하다는 장점이 있다. Fig. 9는 사례와 함께 메시지를 교환하는 과정을 보여주고 있다.

에이전트들이 주고 받을 수 있는 인터페이스는 "Content Layer" 언어인 IDL로 정의하며, 이를 서버 측 인터페이스 저장소에 담아 두고 에이전트의 "Belief"로 활용한다. "Belief"는 KQML 에이전트의 정보 저장고로 본 연구에서는 CORBA의 인터페이스 저장소와 같은 역할을 한다. PA 에이전트가 메시지를 통해서 현재 프로세스 상황을 요청하면 WF 에이전트는 이 메시지를 해석하고 인터페이스 저장소에서 해당하는 인터페이스를 찾아 ORB를 통해 워크플로우 객체를 호출하여 프로세스 상황을 반환 받는다. 이때 WF 에이전트는 "Reply" "performative"로 현재의 태스크와 그 상태를 PA 에이전트에게 반환한다.

모든 에이전트들은 Beliefs를 유지하면서 "Tell" "performative"를 통해서 자신이 처리할 수 있는 지식베이스로서의 IDL 인터페이스를 늘려갈 수 있다. 또 에이전트들 간의 언어 통일을 위해 ontology를 저장하여 두고, 모든 에이전트들이 공유한다. Table 1은 ontology의 예를 보여준다.

5.3 에이전트 구현

에이전트 간의 정보교환 실행 과정은 다음과 같다. 먼저 접속 권한이 있는 사용자가 애플릿이 설치된 시스템에 접속하면 애플릿이 실행된다. Fig. 10은 사용자 환경의 기본적인 작업창을 나타내는 것으로 웹을

Table 1. Examples of ontology

Ontology	Vocabulary
DESIGN CHANGE	PART_SHORTAGE, DELIVERY_DATE_CHANGE, DESIGN_CHANGE
WORKFLOW	PROCESS_ID, FIRST_TASK, PROCESS_STATUS
PART	PART_ID, CURRENT_STOCK, WEIGHT
DOCUMENT	DOCUMENT_ID, AUTHOR, VERSION, DOCUMENT_TYPE

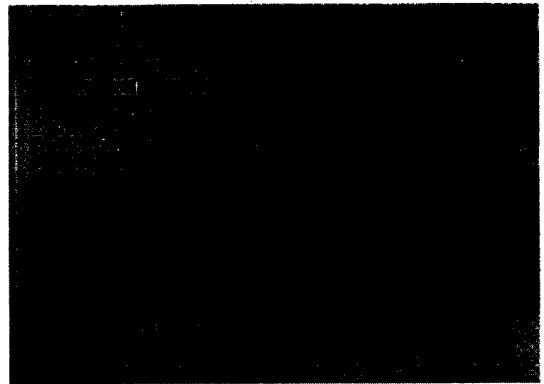


Fig. 10. User interface of agent invocation.

통하여 에이전트에 접근한다.

워크플로우 에이전트를 호출하기 위한 준비 작업으로 워크플로우 에이전트 라우터를 실행시켜 필요 시 워크플로우의 기능을 전달할 수 있게 한다. 클라이언트는 메신저 에이전트를 통하여 메시지를 전달할 수 있는데 특정 에이전트에 접속하기 위하여 수행문에 검색, 수신자에 특정 에이전트, 그리고 실행언어로 KQML을 적는다. 검색에 필요한 키워드를 입력하면 메시지는 네트워크를 통하여 상대방 에이전트(예: WF 에이전트)에게 보내진다. 이제 에이전트는 분산된 데이터베이스를 검색하며, 애플릿의 한계로 인하여 접속한 시스템의 검색만 가능하다. 이때 분산된 데이터베이스를 검색하기 위하여 애플리케이션을 각각의 클라이언트에 설치하는 것이 시스템 사용에 선행되어야 한다. 검색결과를 각 클라이언트에게 보내진다. 메시지를 보내는 과정은 JATLite를 이용하였다.

5.4 에이전트 통신 구조

에이전트들이 분산되어 있고 매우 동적으로 변하기 때문에 해결해야 할 문제가 있다. 인터넷 상에서 널리 쓰이는 클라이언트-서버 아키텍처는 수동적 정보 서비스 위주로 사용된다는 점에서 너무 제한적이다.

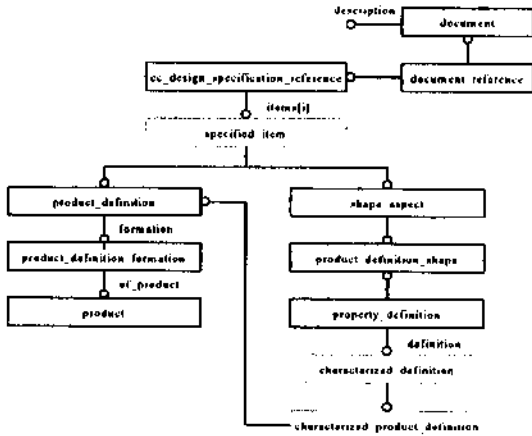


Fig. 12. AP203 schema used for content search.

함하여 6 종류가 있다.

AP203의 제품 모델에서 내용검색과 관련된 부분을 EXPRESS-G로 표현한 것이 Fig. 12이다.

제품의 정의는 형상을 포함하는 shape_aspect와 설계, 재료, 프로세스, 표면처리 등의 규정을 포함하는 document로 이루어진다. 구현한 내용검색 시스템은 사용자로부터 얻은 특성들을 이용하여 해당되는 제품 또는 부품을 검색한다. 입력된 제품 또는 부품에 대한 특성과 데이터베이스에 저장된 document의 description을 비교하여 부분적으로 또는 전체적으로 일치된 경우를 찾는다. 이때, 이러한 속성의 인출은 SDAI의 get_attribute 함수를 이용한다. 검색된 document로부터 cc_design_specification_reference를 통해 파트를 포함하는 product의 정보와 형상을 포함하는 shape_aspect를 제품 데이터베이스에서 추출한다. 내용검색의 결과로서 주어진 특징을 포함하는 부품 리스트와 사용자가 원하는 경우 부품 형상을 화면에 도시한다.

6.2 내용 검색 시스템의 활용

내용검색 시스템은 사용자가 직접 또는 워크플로우 관리시스템이나 에이전트와 같은 응용프로그램이 내용 검색 인터페이스를 통해 STEP 파일이나 DB에 저장된 제품 데이터를 검색한다. 다음은 내용검색시스템의 인터페이스를 위해 정의된 IDL 인터페이스이다.

```
interface GetPart{ //FOR SEARCH.
    short getNextPart(out Product_Data p_data);
    short getResultCount();
    void resetPart();
    void putQuery(in string query);
```

여기서 사용된 putQuery() 함수는 사용자 질의에 포함된 제품을 찾아 linked list에 저장한다. resetPart() 함수는 이 list의 처음을 가리키게 하고, getNextPart() 함수를 사용해 제품에 대한 정보를 가져올 수 있다. 사용자는 AND와 OR를 모두 활용하여 질의문을 작성할 수 있다. 내용검색 서버가 분산환경에서 다수 존재할 때 사용자가 서버의 위치를 일일이 지정하지 않아도 내용검색 미디어이터(mediator)를 설치하여 지능적인 검색을 지원할 수 있다. 이는 STEP 데이터의 위치, 범위, DB 종류, 제품 데이터의 종류 등에 관한 메타 데이터를 이용하여 사용자나 응용프로그램이 입력한 특성을 갖는 제품을 네트워크 환경에서 검색한다.

다음은 사용자가 웹 브라우저를 통하여 내용검색을 실행했을 때의 결과이다. 웹 인터페이스는 사용자가 애플릿이 포함된 웹 페이지에 접속하여 애플릿을 실행함으로써 시작된다. 검색 할 제품의 특징이 입력되면 애플릿은 이를 내용검색 미디어이터에 전달한다. 이 예에서는 "air finger"이며 "angle open/close" 방식의 로봇 그리퍼를 검색한다. 미디어이터는 검색 결과를 사용자에게 전달하여 검색된 부품의 리스트를 열거한다. 사용자는 원하는 부품을 선택해서 더욱 세부적인 명세를 얻을 수 있고, 부품의 형상을 보여주는 있는 뷰어를 선택할 수도 있다. Fig. 13은 이런 과정을 거쳐 선택된 그리퍼 형상을 도시한 화면이다.

7. 결론 및 추후 연구과제

제품정보의 효율적인 관리리는 기업 경쟁력 향상의 중요한 부분이다. 이 논문에서는 제품정보 관리를 위한 분산, 개방, 지능형 제품정보관리시스템의 설계 및 개발 결과를 설명하였다. 제품정보는 주로 제품의 설계 및 생산과 관련된 정보로 이는 분산된 여러 컴퓨팅 환경에서 관리되는 경우가 많다. 또, 각각의 업무

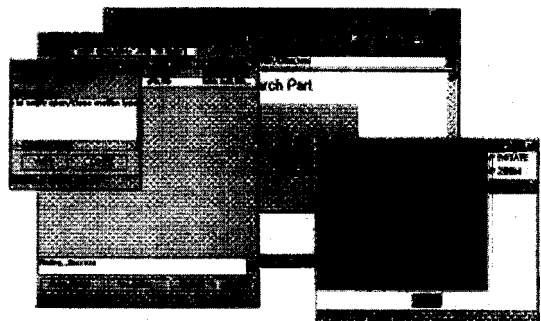


Fig. 13. Gripper design retrieved by content search.

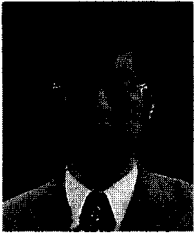
가 독립적이지만 절차를 따르면서 진행되고, 이 절차는 내 외부의 환경 변화와 함께 동적으로 변화될 필요가 있다. 이 논문에서는 이러한 환경에서 PDM 시스템이 가져야 할 요구사항을 정리하고 사례를 통해 설명하였다. 그리고 전체 시스템의 구조를 정보공유 기반구조와 통합 업무수행환경의 두 부분으로 나누어 설계하였으며, 시스템을 구성하는 주요 요소인 워크플로우시스템을 에이전트와 STEP 기반 내용검색을 이용하여 프로세스 관리 기능을 강화하였다.

워크플로우관리시스템에서는 여러 적응성 모델의 도입을 시도하였고, 이를 구현하기 위해 지능적 에이전트를 도입하여 실행 시 프로세스의 동적 변경이 가능하도록 하였다. 에이전트는 또 내용검색 서버와 연결되어 STEP의 장점을 최대한 활용할 수 있는 내용검색 기능을 이용할 수 있다.

추후 과제로 일정관리 에이전트의 스케줄링 기능의 보완과 상용 GT 시스템의 에이전트화 및 통합이 현재 진행되고 있다. SGML을 이용한 실시간 공동문서 작업환경 구축과 공동작업 참여자 간의 컨트롤 이전 모델도 구상 중이다. 한편 에이전트 기술에 있어서는 상충하는 에이전트의 조정 문제와 적용대상과 환경에 따라 새로운 에이전트를 손쉽게 추가하고, 삭제하는 방안도 필요하다.

참고문헌

1. Chorafas, D. N., *The Engineering Database*, Butterworths, 1988.
2. Fischer, L., *The Workflow Paradigm*, Future Strategies, Inc., Alameda, CA, 2nd ed., 1995.
3. Grudin, J., "Why CSCW Applications Fail: Problems in Design and Evaluation of Organizational Interfaces", In *CSCW 88 Proc.*, ACM, pp. 85-93, 1988.
4. Grudin, J. and Palen, L., "Why Groupware Succeeds: Discretion or Mandate?", In *Proc. of the 4th European Conf. on CSCW*, Kluwer Academic Publishers, pp. 263-278, 1995.
5. ISO, *STEP Part I: Overview and Fundamental Principles*, Geneva, 1994.
6. ISO, *STEP Part II: The EXPRESS Language Reference Manual*, Geneva, 1994.
7. ISO, *STEP Part 21: Clear Text Encoding of the Exchange Structure*, Geneva, 1994.
8. ISO, *STEP Part 203: Application Protocols: Configuration-Controlled Design*, Geneva, 1994.
9. ISO, *STEP Part 26: Interface definition language binding to the standard data access interface (CD)*, Geneva, 1997.
10. ISO, *STEP Part 22: Standard Data Access Interface Specification (DIS)*, Geneva, 1998.
11. Meijler, T. D., Kessels, H., Vuijst, C. and Le Comte, R., "Realizing Run-time Adaptable Workflow by means of Reflection in the Baan Workflow Engine", In *CSCW 98 Proc.*, ACM, 1998.
12. Mohan, C., "State of the Art in Workflow Management System Research and Products", *Workflow Systems & Interoperability*, IBM Almaden Research Center, Turkey, 1997.
13. Mowbray, T. J. and Zahavi, R., *The Essential CORBA: Systems Integration Using Distributed Objects*, John Wiley & Sons, 1995.
14. PDM Information Center-Understanding PDM (<http://www.pdmic.com/undrstud.html#brief>).
15. PIF Working Group, *Process Interchange Format Working Group* (<http://ccs.mit.edu/pif/pif-group.htm>), 1998.
16. Rusinkiewicz, M. and Sheth, A., "Specification and Execution of Transactional Workflows", In *the Modern Database Systems*, W. Kim (Ed.), Addison-Wesley, 1994.
17. Schlenoff, C., Knutilla A. and Ray, S., *Unified Process Specification Language: Requirements for Modeling Process*, NIST, U.S. Dept. of Commerce, Sep. 1996.
18. OMG, *What is CORBA?*, OMG Homepage (<http://www.omg.org/corba/>).
19. WfMC, *Interface 1: Process Definition Interchange*, pp. 7-14, 1998.
20. JATLite, <http://java.stanford.edu/>.
21. Finin, T., McKay, D. and Fritzson, R., *An Overview of KQML*, Technical Report, Computer Science Department, U. of Maryland, 1992.
22. Cutkosky, M., Genesereth, M., et al., "PACT: An Experiment in Integrating Concurrent Engineering Systems", *IEEE Computer Special Issue on Computer Supported Concurrent Engineering*, Vol. 26, pp. 28-37, Jan. 1993.
23. Petrie, C., Jeon, H. and Cukosky, M., "Combining Constraint Propagation and Backtracking for Distributed Engineering", *Working Notes of the ECAI-96 Workshop on Non-Standard Constraint Processing*, Aug. 1996.
24. Frost, H. and Cutkosky, M., "Design for Manufacturability via Agent Interaction", In *Proc. of the 1996 ASME Computers in Engineering Conf.*, pp. 1-8, Aug. 1996.
25. Ahn, S.J. and Lee, S.H., "Agent-based Collaborative Design Environment using WWW", *Trans. of the Society of CAD/CAM Engineers*, Vol. 3, No. 1, pp. 31-39, Mar. 1998.



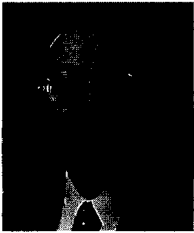
김 영 호

1985년 서울대학교 산업공학과 학사
 1987년 서울대학교 산업공학과 석사
 1988년~1990년 한국과학기술연구원 연구원
 1993년 North Carolina 주립대 산업공학과 공학박사
 1993년~1995년 전북대학교 산업공학과 전임강사
 1995년~현재 서울대학교 산업공학과 조교수
 관심분야: 정보시스템, 동시공학, 인터넷 응용, Internet 기반 PDM개발, Work flow 관리 시스템 개발, 의료정보 시스템 등



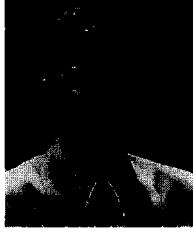
이 수 홍

1981년 서울대 기계공학과 학사
 1983년 서울대 기계공학과 석사
 1991년 Stanford 대학 Design Division Concurrent Engineering 전공, 박사
 1991년~1992년 Lockheed Missile and Space Co. Cable Harness Design System개발 Post-Doc.
 1983년~1994년 KIMM CAD/CAM실, 선임 연구원
 1994년~현재 연세대학교 기계공학과, 부교수
 관심분야: 동시공학설계, 지식기반시스템 설계, DFM



강 석 호

1970년 서울대학교 물리학과 학사
 1972년 미국 Univ. of Washington 산업공학과 공학석사
 1976년 미국 Texas A&M Univ. 산업공학과 교수
 1976년~현재 서울대학교 산업공학과 교수
 관심분야: 생산정보시스템의 설계와 운용, 지능형 생산 시스템, 경영정보 시스템, 생산 계획 및 통제



유 상 봉

1982년 서울대학교 제어계측공학과 학사
 1986년 Arizona 주립대학교 전기및컴퓨터공학과 석사
 1990년 Purdue대학교 전기및컴퓨터공학과 박사
 1989년 AT&T Bell 연구소 연구원
 1990년 삼성전자 컴퓨터부문 선임연구원
 1991년~현재 인하대학교 자동화공학과 부교수
 관심분야: 공학 데이터베이스, STEP, CALS/EC, PDM, Internet Computing, 동시공학