XML 기반 엔지니어링 프로세스 자동화 도구 및 편집기 개발

Development of XML-based System for the Automation of Engineering Process

*#이재경, 박성환, 방제성, 차무현, 남소정

*#J. K. Lee(jkleece@kimm.re.kr), S. W. Park, J. S. Bang, M. H. Cha, S. J. Nam

한국기계연구원 e-엔지니어링연구센터

Key words : Engineering Process Automation, XML-based Process Automation, Engineering Framework

1. 서론

제품 개발기간의 단축 및 품질 향상을 위하여 제품 개발에 필요한 엔지니어링 프로세스를 자동화하고 통합하는 것이 필수적이며 이를 바탕으로 동시공학을 통한 협업, 설계/해석 프로세스 및 시스템 통합, 데이터의 체계적인 관리가 가능하게 된다[1,2].

일반적으로 프로세스란 개인, 부서, 관련 조직들 사이에 이루어지는 의사결정에 필요한 일련의 과정인 비즈니스 프로세스를 의미하며, 엔지니어링 프로세스는 제품 개발에 필요한 협업적인 설계/해석 업무에 필요한 절차로 정의할 수 있다. 현재 널리 쓰이고 있는 PLM(Product Life-cycle Management)[3] 시스템에서는 일반적인 프로세스 및 제품 정보는 관리가 되고 있으나 제품개발 과정의 엔지니어링 정보 및 엔지니어링 프로세스는 관리되지 못하고 있으며, 엔지니어링 통합 시스템인 미국 Phoenix사의 ModelCenter[4]와 Engineous Software사의 iSIGHT[5]에서 관리되고 있으며, 국내 개발 사례로는 한국기계연구원의 멀티 에이전트 기반 엔지니어링 통합 프레임워크[1]를 들 수 있다.

본 논문에서는 멀티 에이전트 기반 엔지니어링 통합 프레임워크(이하 통합 프레임워크)에서 엔지니어링 프로세스 자동화를 위하여 사용한 방법론 및 XML 기반 자동화 도구와 자동화 프로세스 편집기를 소개한다.

2. 엔지니어링 프로세스 자동화 방법론

통합 프레임워크는 멀티 에이전트 기반 시스템으로 시스템 내의 여러 에이전트들의 의사소통을 기반으로 제품개발 프로세스를 협업적으로 수행한다. 시스템 구성 에이전트 중에서 ES(Engineering Server) 에이전트는 제품개발 프로세스를 협업적으로 수행하기 위한 워크플로우 기반의 협업 모델을 처리하고 실제 엔지니어링 프로세스, 즉 전문가가 엔지니어링 지식(설계/해석 지식)을 바탕으로 엔지니어링 도구를 사용하여 설계/해석 문제를 해결하는 에이전트는 Process/Analysis Server (PAS) 에이전트이다[1,2]. PAS 에이전트는 실제 엔지니어링 프로그램들(CAD, CAE 등)과 연동되는 XML 기반 프로세스 자동화 도구인 Resource Wrapper와 저장된 엔지니어링 지식과 입출력 데이터를 활용하여 Resource Wrapper를 구동하는 PAS Process를 가지고 있다. Figure 1은 PAS 에이전트의 구조도 이다. 현재 엔지니어링 지식은 프로세스 자동화 정의파일, 작업이력 등이 관리되고 있으며, 향후 범위를 확대시켜 에이전트 지능화에 사용할 예정이다. 통합 프레임워크는 제품 개발 워크플로우에 따라 엔지니어링 프로세스를 진행시키며, 해당 프로세스에서 필요한 입력 데이터를 EDM(Engineering Data Management)에서 추출하여 입력 XML 파일(ParaInput.XML)로 변환하고 PAS 에이전트에 전달한다. PAS 에이전트는 PAS Process를 실행시켜 입력 XML 파일과 프로세스 자동화 정의파일(Wrapper.XML)을 Resource Wrapper를 전달하고 구동시킨다. Resource Wrapper는 Wrapper.XML을 처리하여 엔지니어링 프로그램을 구동하고 수행결과 및 출력 XML 파일(ParaOutput.XML)을 PAS Process에 전달한다. PAS Process는 수행결과 파일 및 출력 XML 파일을 Local EDM에 저장하고 PAS 에이전트에게 수행결과를 통보한다. PAS 에이전트는 수행결과를 통합 프레임워크의 EDM에 반영하고 통합 프레임워크에 메시지를 전달한다.

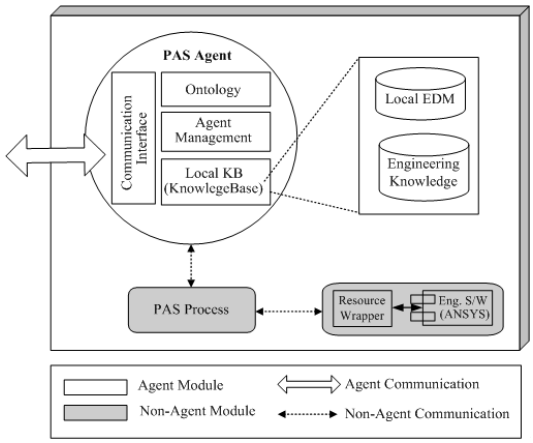

Fig. 1 Architecture of Process/Analysis Server Agent

PAS 에이전트가 모델링한 엔지니어링 프로세스에서 사용되는 설계/해석 소프트웨어는 각각의 독립적인 환경 하에서 운영되므로, 이를 타 서비스나 시스템에 연동하기 위해서는 소프트웨어 랩퍼(software wrapper) 기법을 사용하여야 한다. 소프트웨어 랩퍼는 타 응용 프로그램과 연결하기 위하여 입출력 데이터를 표준형식으로 변환하고 이를 연동하는 소프트웨어 아답터로 정의된다. 소프트웨어 공학에서는 특정 클래스의 인터페이스를 클라이언트가 기대하는 다른 인터페이스로 전환시키는 절차를 아답터(adapter) 패턴으로 정의하며[6], 인터페이스가 호환되지 않아 상호작용할 수 없는 경우에 아답터를 이용하여 클래스 사이의 인터페이스의 호환성을 보장할 수 있음을 나타낸다. 본 논문에서 제안한 엔지니어링 프로세스 자동화 방법론에서는 입출력 데이터의 표준형식으로 XML을 사용하였으며, XML 기반 입출력 데이터를 엔지니어링 소프트웨어가 처리하기 위한 XML 태크 기반 명령어 처리기인 자동화 도구를 개발하였다.

한편, 통합 프레임워크는 멀티 에이전트 기반 시스템이기 때문에 멀티 에이전트 호환성 관련 국제표준인 FIPA(Foundation for Intelligent Physical Agents)을 준수하여 설계 및 개발되었으며, FIPA 표준에서는 멀티에이전트 시스템에서 에이전트가 아닌 소프트웨어와의 연동방안을 정의하고 있다[7]. 에이전트 시스템이 아닌 소프트웨어를 FIPA 호환 에이전트 시스템에서 사용하고자 할 때에는 해당 소프트웨어에 대한 정보, 서비스 정보를 ARB(Agent Resource Broker) 에이전트에 등록하고 소프트웨어를 사용할 Client Agent는 ARB를 통하여 해당 소프트웨어와의 연동을 제공하는 Wrapper Agent를 찾아서 해당 소프트웨어를 이용한다는 것을 정의하고 있다. 통합 프레임워크에서는 DF 에이전트가 ARB역할을 PAS 에이전트가 Wrapper Agent 역할을 수행한다.

본 연구에서 제안한 엔지니어링 프로세스 자동화 방법론은 전술한 아답터 패턴을 통한 레거시(legacy) 시스템 연동과 FIPA (Foundation of Physical Agent)에서 정의한 소프트웨어 연동 표준을 고려하여 설계 되었으며, XML 기반 프로세스 자동화 도구는 Resource Wrapper로 엔지니어링 프로세스 자동화 정의를 생성하고 수정할 수 있는 편집기로 Task Builder를 구현하였다. 개발 된 시스템은 모두 JAVA를 기반으로 하기 때문에 운영체제와 상관없이 수행이 가능한 장점이 있다.

3. 엔지니어링 프로세스 자동화 도구 - Resource Wrapper

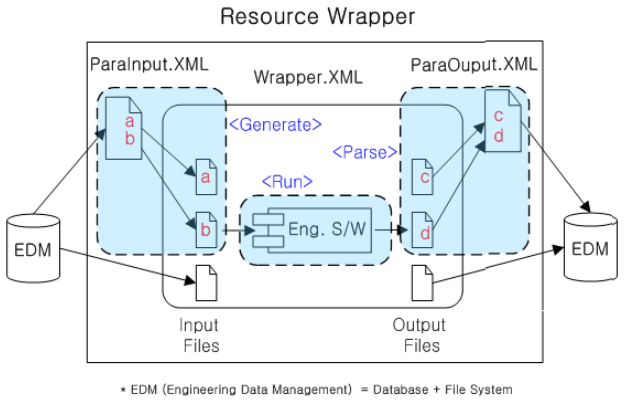

Fig. 2 Structure of Resource Wrapper

개발된 Resource Wrapper는 프로그래밍 언어로 JAVA 1.5를 사용하였으며, Fig. 2와 같이 세 가지 종류의 XML 파일을 기반으로 구성되었다. ParaInput.xml, ParaOutput.xml을 통하여 엔지니어링 프로세스의 입출력 데이터를 처리하며 프로세스 자동화 정의 파일인 Wrapper.xml을 읽어 들여 실제 엔지니어링 프로세스 자동화를 처리한다. Wrapper.xml은 <Generate>, <Run>, <Parse> 구조로 되어 있으며 Resource Wrapper는 이를 순차적으로 처리하여 프로세스 자동화를 수행한다. <Generate> 태그는 엔지니어링 프로그램이 필요로 하는 입력파일을 생성하는 과정으로 엔지니어링 프레임워크에서 전달된 입력 설계변수들이 저장된 ParaInput.xml을 Resource Wrapper가 미리 정의된 입력 템플릿 파일과 바인딩 하여 생성한다. 생성된 입력파일을 기반으로 <Run> 태그부분에 정의된 batch process 기반의 엔지니어링 프로그램이 실행되며, Resource Wrapper는 해당 엔지니어링 프로그램을 구동시키고, 이를 모니터링하여 수행 결과에 대한 정보를 수집한다. <Parse> 태그는 생성된 결과파일에서 출력 데이터를 추출하기 위한 정보를 가지고 있으며, Resource Wrapper는 이에 따라 추출된 출력 설계변수를 ParaOutput.xml에 저장한다.

4. 엔지니어링 프로세스 편집기 - Task Builder

설계/해석 전문가가 PAS 에이전트를 생성하기 위해서는 자신이 사용하는 엔지니어링 프로그램을 자동화 정의 파일로 생성하고 이를 통합 프레임워크 내의 EDM에 저장하는 절차가 필요하다. 설계/해석에 필요한 공학 소프트웨어명과 필요한 입/출력 데이터를 XML(Wrapper.xml) 파일로 정의하고 EDM에 반영하는 이런 일련의 과정을 매번 수작업으로 수행하는 것은 많은 작업 시간이 소요된다. 또한 XML 파일 생성과정에서 사용자는 태그 작성에 어려움이 있고 데이터의 정확성을 보장할 수 없다. 이런 점을 보완하고 수작업으로 진행되던 일련의 과정을 자동화하기 위해 Task Builder를 설계 및 구현하였다. Task Builder는 프로그래밍 언어로 JAVA 1.5를 사용하였으며, 엔지니어링 프로세스 정의 생성 및 수정 기능, 트리뷰 및 미리보기 기능, Wrapper.xml 파일 생성 기능, 통합 프레임워크 EDM 등록 기능들을 제공한다.

사용자는 Wrapper.xml 파일 정의에 사용되는 구성 태그들인 <taskName>, <Header>, <inputFile>, <outputFile>, <generate>, <parse>, <run> 태그별로 입력화면을 구성하고, 정의된 프로세스 자동화 내용은 트리뷰나 미리보기 기능을 추가하여 Wrapper.xml 파일 생성 전에 정의된 항목들을 확인할 수 있게 설계하였다. 또한, 작성된 Wrapper.xml 파일을 이용하여 EDM에 반영하거나 기존의 Wrapper.xml을 수정하고 반영할 수 있다. Table 1은 Wrapper.xml의 구성 태그와 Task Builder의 화면 구성과의 관계를 보여주고 있다. Task Builder의 화면구성은 Fig. 3과 같다.

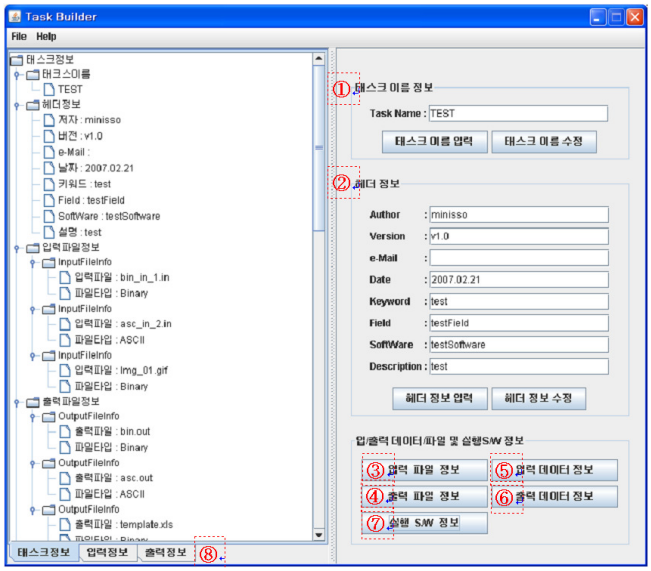

Fig. 3 User Interface of Task Builder

Table 1. Mapping XML <tag> information into Task Builder's GUI

Tag Name	Descriptions	Mapping to GUI
<taskName>	Task Name	①
<Header>	Header	②
<inputFile>	Input File	③
<outputFile>	Output File	④
<generate>	Input Data	⑤
<parse>	Output Data	⑥
<run>	S/W Command	⑦

5. 결론

본 논문에서는 엔지니어링 프로세스 자동화를 위한 XML 기반 자동화 도구 및 편집기를 개발하였다. 개발된 편집기는 전문가의 엔지니어링 지식을 체계화한 XML 파일로 생성하고 엔지니어링 프레임워크에 해당 엔지니어링 서비스를 저장하며, 자동화 도구는 제품 개발 프로세스의 흐름에 따라 적절한 엔지니어링 서비스를 제공하기 위하여 엔지니어링 에이전트에 의해 실행된다. 개발된 자동화 도구 및 편집기는 자동차 모듈 설계용 e-엔지니어링 프레임워크에 적용하여 사용 중에 있으며, 사용자 편의성 증대 및 기능 확대를 수행할 예정이다.

후기

본 논문은 과학기술부 특정연구개발사업 과제인 “자동차 모듈 설계용 e-엔지니어링 프레임웍 개발”의 일부이며, 연구수행에 지원해 주신 관계자 여러분께 감사 드립니다.

참고문헌

1. S.W. Park, J.K. Lee, J.S. Bang, and B.C. Shin, 2005, “Development of an e-Engineering Environment for automotive Module Design,” LNCS 3865, 264-273, 2006.
2. 이재경, 박성환, 이종원, 한승호, 한형석, "멀티 에이전트 기반의 통합설계 시스템 개발". 한국정밀공학회논문집, 제 22권 제 1호, 12-18, 2005.
3. http://www-903.ibm.com/kr/solutions/plm
4. http://www.phoenix-int.com
5. http://www.engineous.com
6. Gamma, Erich., et al., Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley, 1995.
7. FIPA, “Agent Software Integration Specification”, http://www.fipa.org/specs/fipa00079/index.html, 2001