

# 형강 절단 로봇 시스템의 객체 지향 설계<sup>†</sup>

한국과학기술원 전상욱 · 이희진 · 배두환

## 1. 서 론

객체 지향 기법의 등장 이후 많은 소프트웨어 시스템들이 객체 지향 기법으로 설계되고 있다. 또한 OMG (Object Management Group)에서 객체 지향 모델링 언어인 UML(The Unified Modeling Language) [1, 2, 3]을 발표하면서 객체 지향 설계 기법은 더욱 더 많은 시스템의 개발에 적용되고 있다. 이러한 설계 기법은 IT 도메인 뿐 아니라 자동차, 항공기 등과 같은 내장형 시스템의 개발 분야에서도 널리 적용되고 있다[4, 5].

본 논문에서는 **형강 절단 로봇 시스템의 객체 지향 설계와 이 때 적용한 소프트웨어공학 원리 및 기대효과에 대해 기술한다.** 형강 절단 로봇 시스템은 선박 건조에 사용될 형강을 CAD등의 전산화된 모델링을 통하여 형강 제조 데이터를 생성하고, 이를 바탕으로 자동화된 설비라인을 이용하여 최종 형강이 자동으로 생성될 수 있도록 하는 시스템이다. 본 연구에서 사용된 개발 방법론은 Rational Unified Process(RUP)[6]를 기반으로 하고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구의 설계 대상인 형강 절단 로봇 시스템의 초기 요구사항 및 요구사항 분석 결과에 대해 기술하고 3장에서는 요구사항을 바탕으로 시스템 설계에 적용될 수 있는 소프트웨어 공학 원리에 대해 기술한다. 4장에서는 요구사항과 소프트웨어 공학 원리를 바탕으로 한 시스템의 상세 설계에 대해 기술한다. 5장에서는 형강 절단 로봇 시스템을 객체 지향 개념을 기반으로 설계함으로써 얻을 수 있는 기대효과들에 대해 설명한다. 마지막으로 6장에서 결론을 맺고 향후 연구에 대해 기술한다.

## 2. 요구사항 분석

본 장에서는 클라이언트와의 회의를 통해 도출된 초기 요구사항에 및 분석 결과에 대해 기술한다. 형상 제

작 자동화 시스템의 기능 리스트는 다음과 같다.

- 형강 부재를 절단용 로봇과 주변 기기를 이용하여 자동화하기 위한 통합 운영 시스템을 개발한다.
- 사용자는 형강 부재 정보를 셀 제어기에 등록한다.
- 셀 제어기는 레이저 센서로부터 측정 데이터를 읽어 들인다.
- 셀 제어기는 로봇 제어기로 측정 데이터를 전송한다.
- 셀 제어기는 구동 프로그램을 생성하여 로봇 제어기로 전송한다.
- 구동 프로그램은 부재별 하나이 매크로 인스트럭션을 포함하고 있으며, 매크로 인스트럭션은 절단 장치, 측정 장치, 레이블링 장치용으로 구분되어진다.
- 셀 제어기와 로봇 제어기간에는 명령, 응답 형태로 단계별 운전되어진다.
- 셀 제어기는 레이블링 데이터를 생성하여 레이블링 장치로 전송한다.
- 셀 제어기는 PLC로부터 필요한 데이터의 상태를 감시한다.
- 셀 제어기가 작업 내용을 관리한다.

그림 1은 형강 절단 로봇 시스템의 도메인 모델을 보여주고 있다. 시스템은 컨베이어를 이용하여 시스템의 구성 요소들 간의 형강 부재 이동을 담당하는 운송 시스템(Transportation System), 형강 부재 가공을 담당하는 절단소(Cutting Station), 부속 처리 장치(Associated Processing Station)와 이를 중심에서 제어하는 로봇 제어기(Robot Controller)로 구성되며 셀 제어기(Cell Controller)는 사용자의 다양한 요구를 입력받아 다양한 형태의 형강 제작을 자동화 할 수 있도록 지원한다. 로봇 제어기는 이미 구현되어 있는 상태이기 때문에 이를 상위 수준에서 제어할 수 있는 셀 제어기의 역할이 중요하다. 셀 제어기는 일반적으로 메인 프레임 위에서 수행되며 이는 근거리 통신망을 통하여 로봇 제어기와 통신하고 이를 제어하게 된다. 셀 제어기는 형강 제작 작업의 수행 내용이 기록되어

<sup>†</sup> 본 연구는 2006년도 경남도 지역산업기술개발사업의 연구비 지원으로 수행하였습니다.



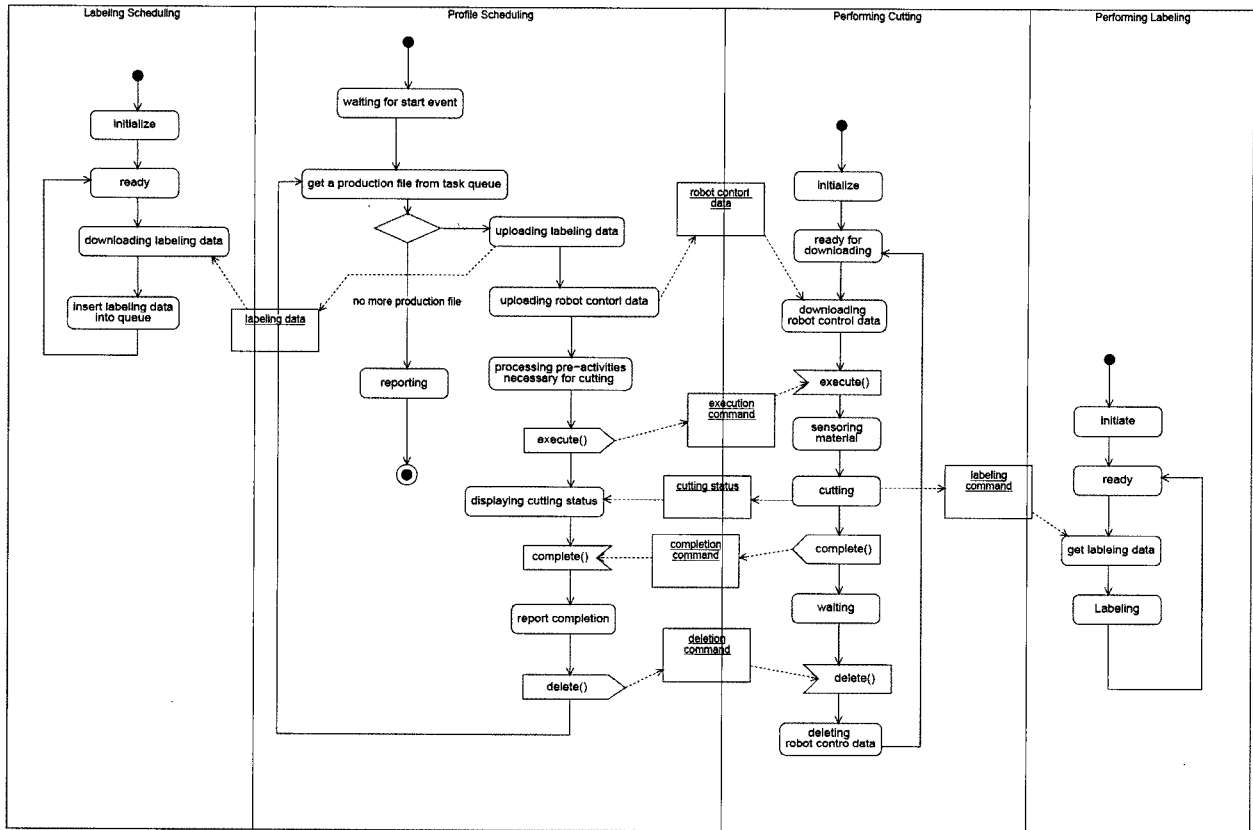


그림 3 Profile control 유즈케이스에 대한 액티비티 다이어그램

스와 이를 실행시키면 작업 계획에 따라 순차적으로 형강 부재를 가공하는 'perform cutting' 유즈케이스, 그리고 가공과 동시에 해당 부재에 정의된 레이블링을 수행하는 'labeling control' 유즈케이스를 포함하고 있다.

### 3. 시스템 설계에 적용된 소프트웨어 공학 원리(Principle)

본 장에서는 앞에서 분석된 요구사항을 검토한 후 형강 절단 로봇 시스템 설계에 적용될 수 있는 소프트웨어 공학 원리에 대해 기술한다.

#### 3.1 데이터 처리와 제어 처리의 구분(Separation of Concern)

셀 제어기는 이미 저장된 production file을 읽어서 그 안의 작업 내용을 번역하여 로봇 제어기와 레이블링 장치로 전송하게 된다. 이 때 한 부분에서 작업 내용의 번역과 다른 부속 장치의 제어를 모두 담당하게 되는 경우 두 작업 간에 필요 이상의 종속성(dependency)을 가지게 된다. 또한 클라이언트의 요구사항을 분석하면서 사용자가 지정한 작업 내용을 표현 및 저장하는 방법이 확정된 것이 아니라 언제든지 변경될 수 있다는 것을 알게 되었다. 따라서 production file의 내용

을 번역하는 작업과 번역된 내용을 로봇 제어기를 비롯한 주변 장치들에게 전송하여 이들을 제어하는 작업을 각각 'translator'와 'task controller'라는 클래스에 할당하였다. 이와 같이 데이터를 처리하는 부분을 주변 장치를 제어하는 부분과 구분함으로써 데이터를 처리하는 방법이 변경되었을 경우 이것이 'task controller'에게 영향을 받지 않는 효과를 얻을 수 있다.

#### 3.2 변화의 예측(Anticipation of Change)

본 시스템의 핵심이 되는 형강 제작 작업의 내용은 시스템이 운영되면서 사용자가 실시간으로 지정하는 것이 아니라 미리 production file의 형태로 저장된 작업 목록을 읽어서 이를 일괄 처리하는 방식으로 이루어진다. 따라서 형강 절단 로봇 시스템의 작업이 시작되기 전에 사용자는 항상 저장된 production file의 목록을 열람하고 이 중 하나를 선택하게 된다. 따라서 production file을 읽고 또한 사용자가 입력한 작업의 내용을 production file의 형태로 저장하는 부분이 중요하게 되는데, 앞에서 기술한 바와 같이 사용자가 지정한 작업 내용을 저장하는 방법이 언제든지 변경될 수 있음을 알게 되었다. 이러한 요구사항을 바탕으로 production file의 목록 및 내용을 읽고 저장하는 부분이 다른 부분에 비해 수시로 변경될 수 있음을 파악

할 수 있었다. 이러한 예측을 기반으로 하여 File Reader 클래스와 File Writer 클래스 등 production file을 취급하는 클래스를 별도의 서브시스템으로 구분하였다.

#### 4. 시스템 설계

본 장에서는 앞에서 기술한 요구사항 분석과 시스템 분석을 거쳐 도출된 행강 절단 로봇 시스템의 최종 설계에 대해 기술한다.

시스템 분석을 통해 식별된 분석 수준 패키지를 기반으로 설계 수준의 서브시스템(subsystem) 개요를 도출해 내고 분석 수준의 클래스를 통해 설계 수준의 클래스 개요를 도출해낸다. 이를 기반으로 설계 모델 내에서 각 유즈케이스가 설계 클래스와 객체들 내에서 수행되고 구체화되는 방법을 나타내기 위한 설계 수준의 유즈케이스 실현(Use Case realization - design)을 작성한다. 그림 4와 그림 5는 시퀀스 다이어그램(sequence diagram)으로 표현한 'profile control' 유즈케이스 실현의 첫 번째와 두 번째 부분을 각각 보여주고 있다. 사용자는 UI를 통해 작업 내용이 담긴 파일들의 목록을 열람하고 이 중 한 파일을 선택하면 Scheduler handler가 이 파일의 내용을 읽어서 새로

운 Scheduler를 생성하게 된다. 사용자가 작업의 시작을 요청하면 Scheduler Handler가 이를 Task Controller에게 알리고 Task Controller는 해당 Scheduler로부터 파일의 내용을 읽어서 production file의 형태로 변환한다. 이 production file로부터 추출된 레이블링 데이터는 Labeling Task Handler에게 보내어 Label Control File을 생성하고 이를 레이블링 장치에 등록하게 하고 로봇 제어 데이터는 Profile Task Handler에게 보내 Robot Control File을 생성하여 로봇에 등록하게 한다. 이후 Profile Task Handler와 Labeling Task Handler는 각각 로봇과 레이블링 장치와 통신하며 작업을 수행하게 된다. 작업 진행 상황은 Profile-Status Displayer를 통해 UI를 거쳐 사용자에게 보여지게 된다.

이러한 설계 수준의 유즈케이스 실현과 앞에서 작성한 서브시스템 개요를 통해 각 클래스의 속성과 메소드를 추가함으로써 그림 6과 같은 완전한 설계 수준의 클래스가 완성된다. 총 30개의 클래스가 도출되었으며 유즈케이스 실현에서 도출된 메소드들과 그 외에 부가적으로 필요하다고 판단되는 속성들을 추가하였다. 그림 7은 설계 단계의 마지막 산출물인 완전한 설계 수준의 서브시스템을 보여준다. 이는 앞에서 작성된 서브시스템 개요를 기반으로 완전한 설계 수준의 클래스와

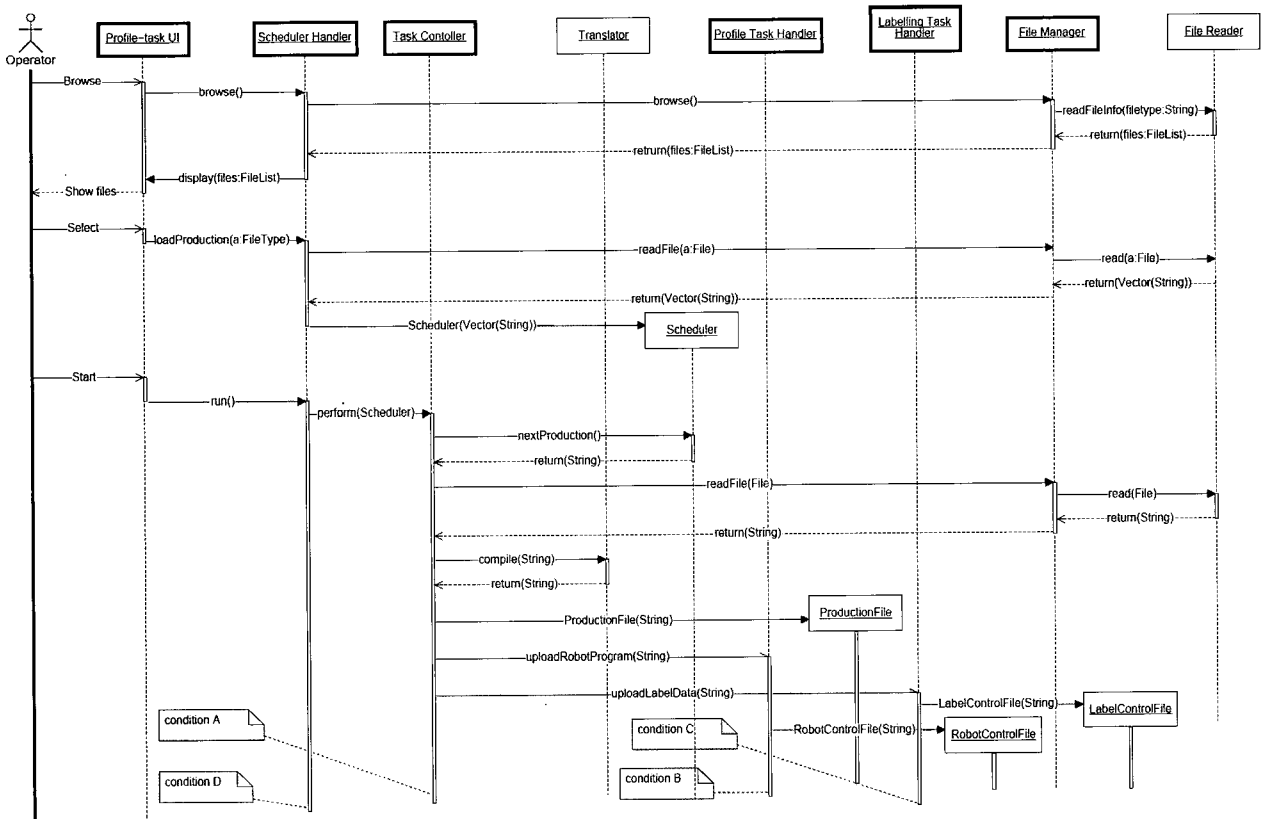


그림 4 설계 수준의 Profile control 유즈케이스 실현(1)

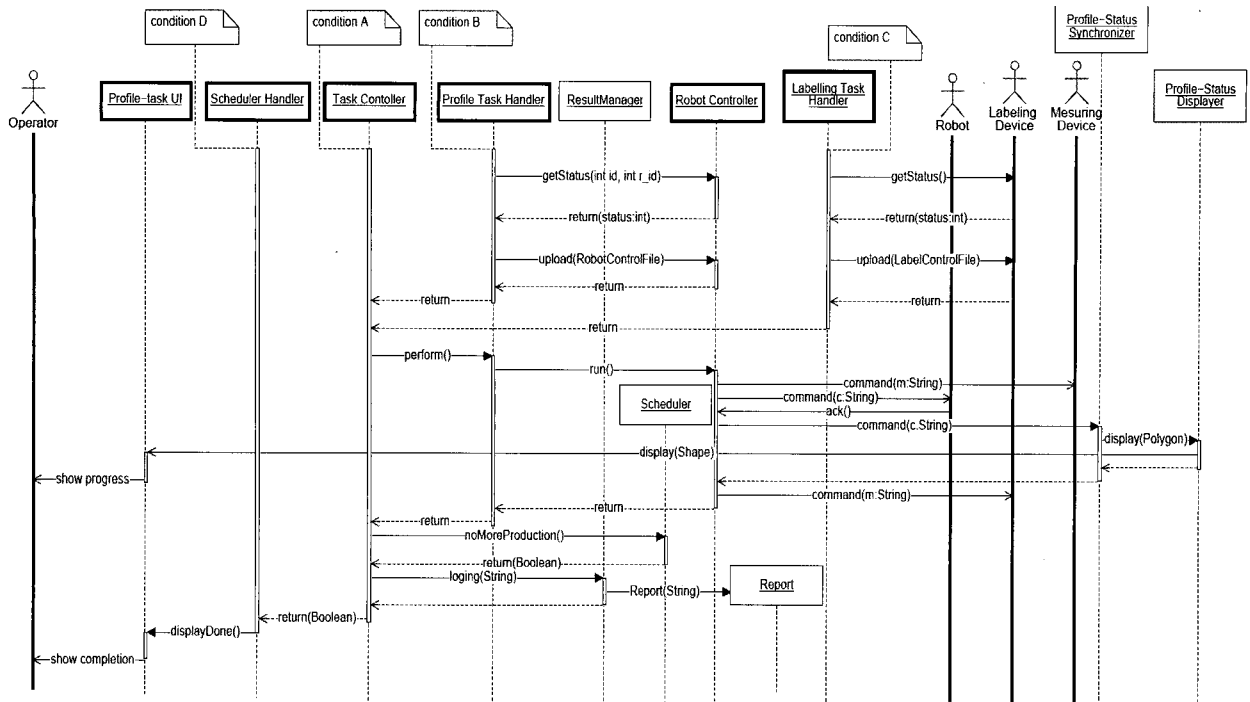


그림 5 설계 수준의 Profile control 유즈케이스 실현(2)

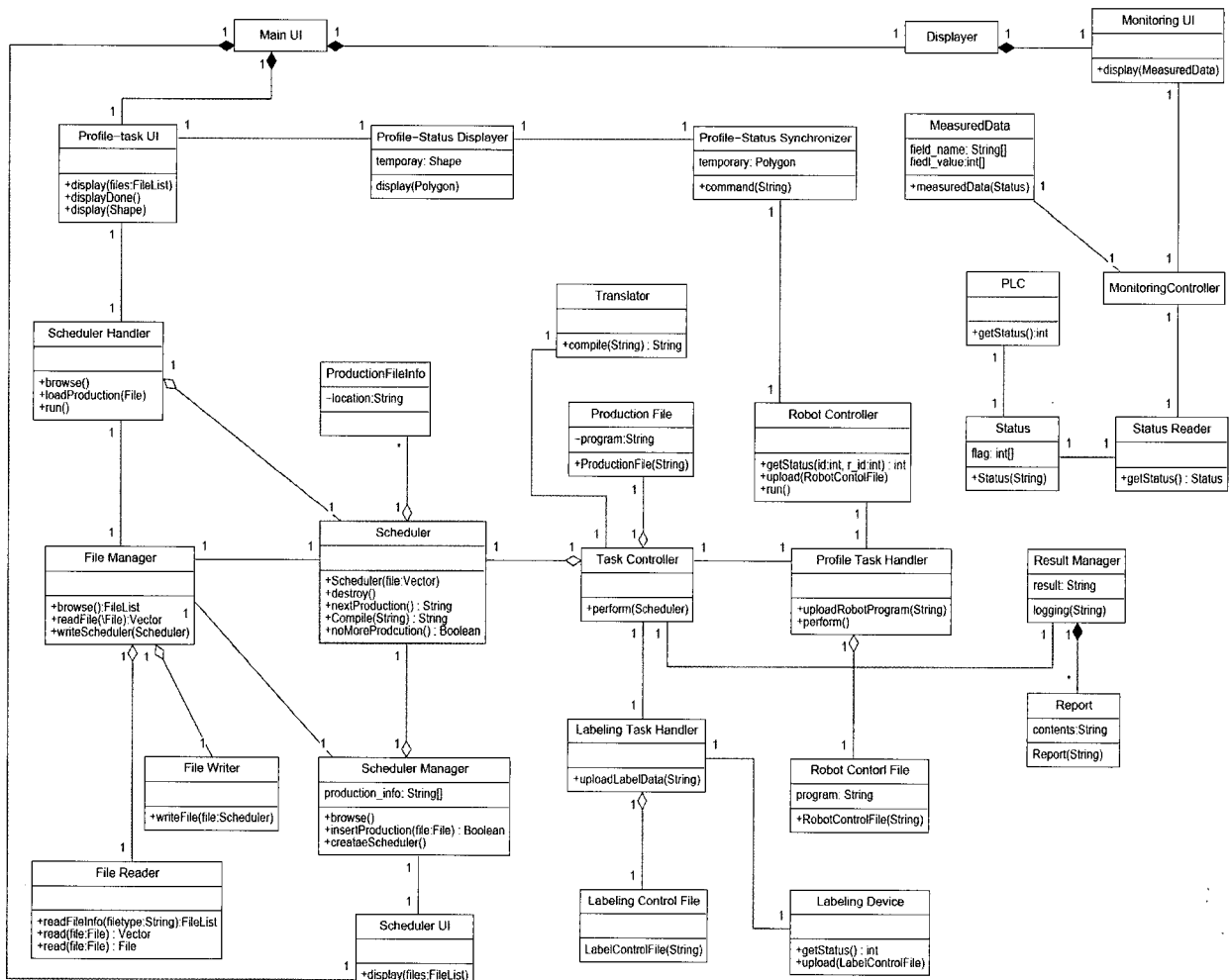


그림 6 설계 수준의 최종 클래스

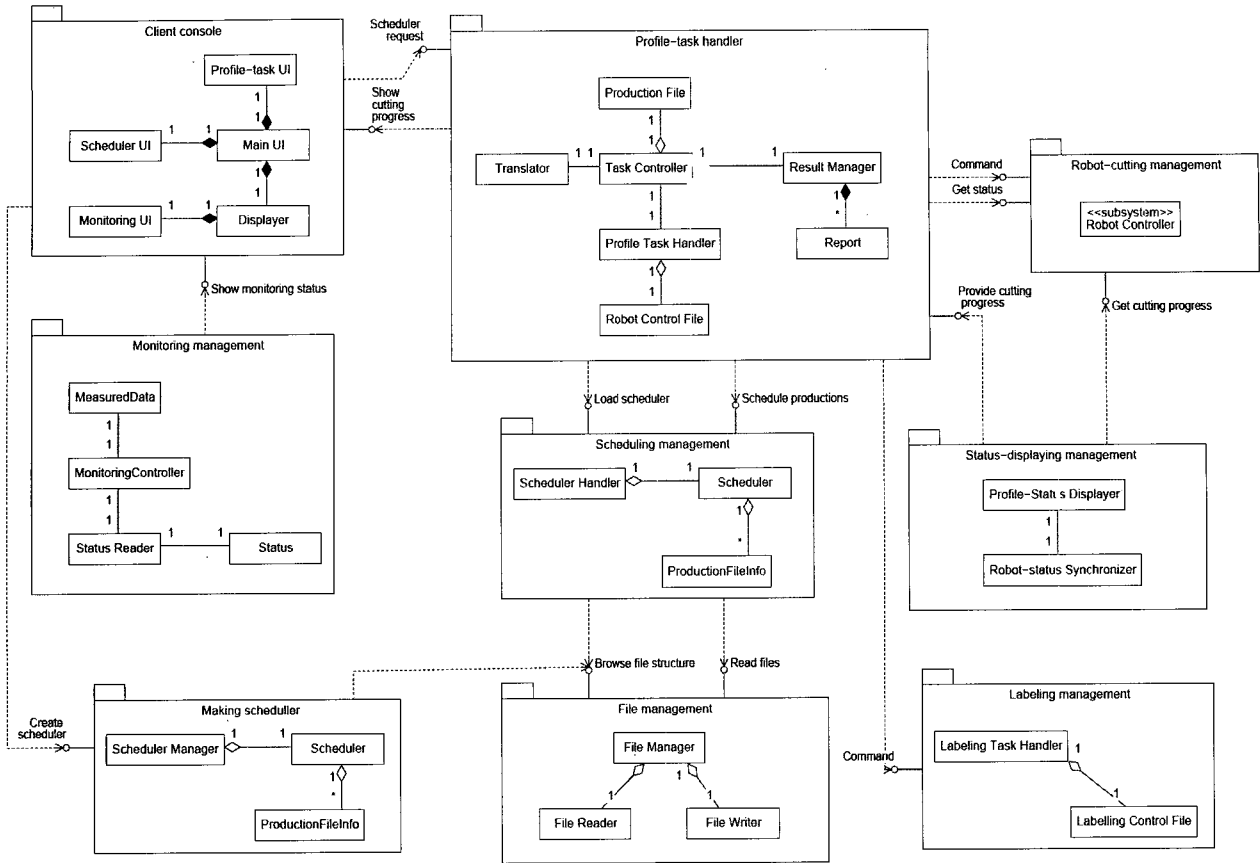


그림 7 설계 수준의 최종 서브시스템

식별된 인터페이스를 통하여 작성되었다. Profile-Task Handler와 Labeling Management 등 총 9개의 서브시스템이 식별되었다.

## 5. 기대효과

본 장에서는 객체 지향 개념을 바탕으로 설계된 형강 절단 로봇 시스템을 사용함으로써 얻을 수 있는 기대효과에 대해 기술한다.

### 5.1 클라이언트와의 의사소통 원할

산업체에서 널리 사용되고 있는 UML을 이용함으로써 클라이언트와의 쉽고 명확한 의사소통이 가능했다. 도메인 모델을 작성함으로써 형강 절단 로봇 시스템의 도메인에 어떤 구성 요소들이 있는지 명확히 할 수 있었다. 또한 유즈케이스 모델을 작성함으로써 설계하고자 하는 시스템의 경계를 정확히 파악할 수 있었다.

### 5.2 유지보수성 증대

앞에서 기술한 바와 같이 형강 제작 작업 내용이 기록되어 있는 production file의 표현 및 저장 형태가 확정된 것이 아니라 수시로 변경될 수 있는 것이었다.

시스템 설계에서 이에 대한 부분과 다른 부분들과의 종속성을 최소화함으로써 변경된 요구사항에 맞추어 설계를 변경할 때 다른 부분에 큰 영향을 끼치지 않아 결과적으로 유지 보수비용이 감소하였다. 또한 형강 절단 로봇 시스템의 유지보수에 대한 담당자가 변경되었는데 이때의 인수인계 비용이 감소했음을 확인할 수 있었다.

### 5.3 모델 재사용성 증대

본 연구의 대상이 되는 형강 절단 로봇 시스템의 설계 및 구현 후 이를 실제 하드웨어에 탑재하여 운영하였다. 얼마 후 탑재되는 하드웨어에 약간의 변경사항이 발생하였다. 따라서 이에 맞는 시스템을 새로 설계할 필요가 생겼다. 변경된 하드웨어에 탑재될 소프트웨어의 요구사항을 분석한 결과 기존에 사용하고 있던 소프트웨어 설계의 일부만을 변경함으로써 가능하다고 판단되었다. 이전에 사용되던 C와 같은 순차적 언어로 이에 대한 비용이 새로 개발하는 비용과 크게 다르지 않았지만 객체 지향 개념을 기반으로 설계된 소프트웨어는 이에 비해 모델 및 코드의 재사용성을 높여 개발 비용을 줄일 수 있었다.

## 6. 결 론

본 논문에서는 형강 제작 자동화 제어 시스템을 객체 지향 개발 기법으로 설계한 결과를 제시하였다. 형강 제작 자동화 제어 시스템의 요구사항을 기술하였고 이를 바탕으로 RUP 프로세스에 기반한 시스템 분석 및 상세 설계를 UML을 이용해 작성하였다. 클라이언트로부터의 요구사항을 분석하여 시스템에 적용할 수 있는 두 가지 소프트웨어 공학 원리를 파악하고 이를 시스템 설계에 반영하였다. 시스템을 객체 지향 기법으로 설계함으로써 클라이언트가 모델을 쉽게 이해하도록 하였다. 또한 설계된 시스템이 실제 시스템에 탑재됨으로서 모델의 유지보수성 및 재사용성이 향상되었음을 직접 확인할 수 있었다.

본 논문의 향후 연구는 다음과 같다. 첫 번째로 하드웨어 변경에 따른 시스템 설계의 변경이다. 앞에서 기술한 바와 같이 클라이언트가 보유한 하드웨어가 변경됨에 따라 기존에 설계한 소프트웨어를 수정할 필요가 생겼다. 형강 부재를 가공하는 로봇이 하나가 아닌 두 개가 되어 병렬 작업이 가능해져 생산성을 높일 수 있게 되었다. 따라서 production file에 대한 정보를 두 로봇에 각각 전달할 수 있도록 기존의 설계를 변경하여야 한다. 두 번째는 디자인 패턴[7]의 적용을 통한 소프트웨어의 품질 향상이다. 이미 설계된 소프트웨어에 디자인 패턴을 적용함으로써 유연성이나 유지보수성 등을 증가시킴으로서 소프트웨어의 설계 변경 비용을 줄일 수 있다. 마지막으로 선박 건조 도메인에서의 디자인 패턴을 추출하는 것이다. 본 연구의 대상인 형강 제작 자동화 제어 시스템과 유사한 시스템의 설계를 통해 공통적으로 발생하는 문제를 찾아 이에 대한 해결 방안으로 디자인 패턴을 개발하게 되면 향후 같은 도메인의 설계에서 발생하는 유사한 문제들을 쉽게 해결할 수 있을 것이다.

## 참고문헌

- [1] The Unified Modeling Language(UML), <http://www.omg.org/uml>
- [2] G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide. Addison-Wesley, 1999.
- [3] M. Fowler, UML Distilled 3rd Edition. Addison-Wesley, 2003.
- [4] B. Selic, "Using UML for Modeling Complex Real-Time Systems," Proceedings of the ACM SIGPLAN Workshop on Languages,

Compilers, and Tools for Embedded Systems, 1998.

- [5] Kirsten Berkenkotter, "Using UML 2.0 in Real-Time Development: A Critical Review," In Proceedings of SVERTS workshop, 2003.
- [6] I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process. Addison-Wesley Longman, 1999.
- [7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

### 전 상 욱



2001 한국과학기술원 전산학과(공학사)  
2003 한국과학기술원 전산학과(석사)  
2003~현재 한국과학기술원 전산학과 박사과정.  
관심분야: 임베디드 소프트웨어 설계, 디자인 패턴, 리팩토링, 소프트웨어 유지보수  
E-mail : sujeon@se.kaist.ac.kr

### 이 희 진



2004 한국과학기술원 전산학과(공학사)  
2006 한국과학기술원 전산학과(석사)  
2006~현재 한국과학기술원 전산학과 박사과정.  
관심분야: 임베디드 소프트웨어 설계, 객체지향 모델링, 모델 검증 기법, 소프트웨어 유지보수  
E-mail : leehj@se.kaist.ac.kr

### 배 두 환



1980 서울대학교 조선공학(공학사)  
1997 미국 Univ. Of Wisconsin - Milwaukee 전산학(석사)  
1992 미국 Univ. Of Florida 전산학(박사)  
1995~현재 한국과학기술원 전산학과 교수.  
관심분야: 소프트웨어 프로세스, 객체지향 프로그래밍, 컴포넌트 기반 프로그래밍, 임베디드 소프트웨어 설계, 원격 지향 프로그래밍  
E-mail : bae@se.kaist.ac.kr