

UML 모델링 도구를 이용한 ATO 차상 소프트웨어의 설계 및 구현

Design and Implementation of ATO On-board Software Using UML Modeling Tool

윤영환*
Yoon, Yeong-hwan

방용**
Bang, Yung

엄정규*
Um, Jung-Kyou

조용기*
Cho, Yong-Gi

ABSTRACT

In this paper, we propose a UML modeling of ATO on-board software. An automatic train operation (ATO) system is a real-time control system, which operates a train without a manual operation by a driver. For the safe and comfortable service, real-time embedded software for ATO on-board equipment should have both of high performance and reliability. UML-based object-oriented modeling technique is introduced and used widely to design software that satisfies this requirement. We used Rhapsody, which is a modeling tool for real-time embedded software, to model the construction and the behavior of ATO on-board equipment. As a result, ATO on-board software which performs the profile calculation and the real-time speed control is designed and implemented. The brief modeling result including behavioral characteristics and the simulation results are presented.

1. 서 론

ATO(automatic train operation) 차상장치는 열차 자동 제어 시스템의 하위시스템으로 운전자의 수동 조작 없이 실시간으로 열차를 운행하는 역할을 수행하며, 이를 위해 속도 프로파일 생성 및 주행 속도 제어, 자동 정밀 정차, 열차문 제어 등의 하위기능을 가진다. ATO 차상장치는 ATP 장치와 같이 열차의 안전과 직결되는 기능을 담당하지는 않으나, 정상적인 상황에서 ATP에 의한 열차 방호가 일어나지 않도록 안전하게 주행하면서 뛰어난 승차감을 제공하려면 ATO 차상장치를 동작시키는 소프트웨어가 높은 성능과 신뢰성을 보일 필요가 있다.

지금까지 실시간 임베디드 소프트웨어를 분석하고 설계하는 것에 구조적 프로그래밍에 뿌리를 둔 구조적 방법론(structured methodology)이 주로 사용되었다. 하지만 구조적 방법론은 프로세스와 데이터를 분리하여 처리하기 때문에 이들을 분석하여 설계와 구현으로 연결시키는 데 복잡한 기법들이 필요하며 결과가 복잡하다는 문제점을 지니고 있다. 이런 문제점을 해결하기 위해 객체지향 프로그래밍을 기반으로 하는 객체지향 방법론(object-oriented methodology)이 제안되었다. 소프트웨어를 분석하고 설계할 때 객체

* LS산전(주), 중앙연구소, 비회원

E-mail : yhyoon1@lss.biz

TEL : (031)450-7270 FAX : (031)450-7599

** LS산전(주), 철도시스템기술팀

지향 방법론을 사용하면 프로세스와 데이터를 객체 단위로 처리하면서 전술한 문제점들을 해결할 수 있고 시스템을 좀 더 이해하기 쉽게 설계할 수 있다. 점점 많은 프로젝트에서 객체지향 방법론이 적용되고 있는 추세이며, 특히 객체지향 모델링 언어 표준으로 제안된 UML(unified modeling language)를 기반으로 하는 다양한 CASE 도구들이 개발되어 활용되고 있다.

본 논문에서는 UML 모델링 기법을 ATO 차상 소프트웨어의 분석 및 설계에 적용한다. 우선 실시간 임베디드 소프트웨어 모델링 도구를 이용하여 ATO 차상장치의 구성 및 동작을 모델링하며, 이를 바탕으로 주행 프로파일 생성 기능 및 실시간 속도 제어 기능을 구현한 소프트웨어를 보인다. 결과로는 동작 특성을 포함하는 모델링 결과 요약 및 시뮬레이션 결과를 제시한다.

2. 본 론

객체지향 방법론을 활용한 소프트웨어 분석 및 설계에서는 (1) 요구사항을 분석하여 유스케이스(usecase)와 시나리오(scenario)를 도출하고 (2) 핵심적인 객체(object)와 클래스(class) 및 관계를 추출하며 (3) 각 객체들의 동작(behavior)을 분석하고 (4) 이와 같은 분석 과정을 통해 도출된 분석 모델을 기반으로 시스템을 여러 수준에서 설계하는 등의 작업을 필요로 한다. UML에서는 객체지향 방법론 기반의 소프트웨어 분석 및 설계 과정에서 시스템을 모델링하고 표현하기 위한 도구로 여러 가지 다이어그램을 제공하고 있다. 대표적인 것으로는 유스케이스 다이어그램, 클래스 다이어그램, 컴포넌트 다이어그램, 액티비티(activity) 다이어그램, 시퀀스(sequence) 다이어그램, 상태기계(state machine) 다이어그램 등이 있다.

본 논문에서는 UML 모델링 CASE 도구로는 Telelogic 사의 Rhapsody를 이용하였으며, 유스케이스 다이어그램, 클래스 다이어그램, 상태기계 다이어그램을 통해 ATO 차상장치를 모델링한 결과를 보인다. 그리고 Rhapsody와 같은 도구들은 분석 및 설계 결과를 기반으로 도출한 모델을 개발 템플릿으로 이용할 수 있도록 기본적인 코드를 생성하는 일까지 수행하는데, 이를 이용하여 프로파일 생성 및 속도 제어 알고리즘을 추가하여 소프트웨어를 구현한 시뮬레이션 결과를 보인다.

2.1 ATO 차상장치의 UML 모델링

2.1.1 ATO 차상장치의 유스케이스 다이어그램

ATO 차상장치가 열차 주행을 관리할 때 담당하는 핵심적인 작업은 (1) 열차 주행을 자동으로 제어하고 (2) 열차를 정위치에 정차시키며 (3) 플랫폼에서 출입문을 제어하는 것을 들 수 있다. (그림 1)에서와 같이 시스템이 수행하는 각각의 작업 활동은 적당한 이름을 붙인 유스케이스로 표현될 수 있으며, 각각의 유스케이스는 그 중 일부의 활동을 명세화하는 여러 유스케이스를 포함(include)할 수 있다.

유스케이스 다이어그램은 이후의 모델링에 대한 기본적인 정보와 방향을 표시하는 데 활용할 수 있다. 위의 경우와 같이 다이어그램을 이용하면 시스템의 어떤 유스케이스와 어떤 액터(actor) 간에 관계가 있는지 간략하게 시각적으로 표현할 수 있다. 하지만 각 유스케이스의 자세한 정보 및 액터와의 상호작용

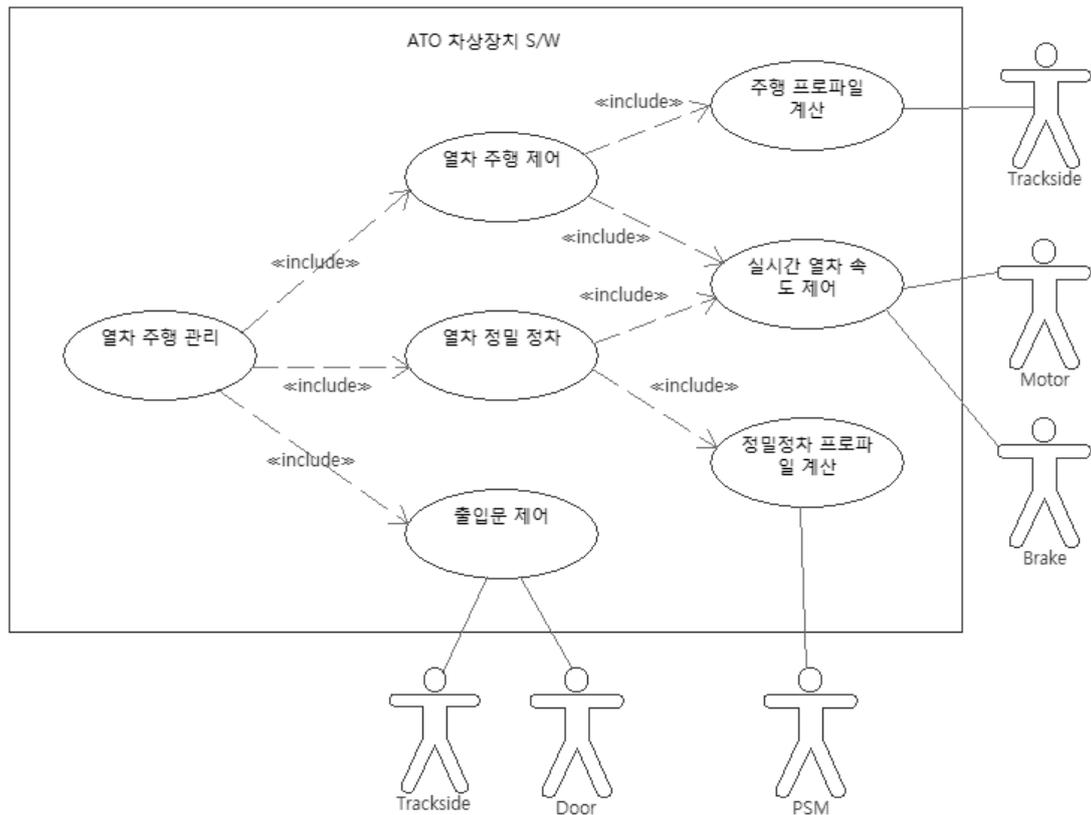


그림1. ATO 차상장치의 유스케이스 다이어그램의 예

용 모두를 다이어그램에 표시할 수는 없다. 유스케이스 다이어그램은 간략하게 유지하되, 추가적인 정보를 따로 기술함으로써 유스케이스 다이어그램에서 표현하기 힘든 정보를 보완할 수 있다.

2.1.2 ATO 차상장치의 클래스 다이어그램

클래스 다이어그램은 시스템을 구성하는 여러 요소들(entities)로 어떤 것이 있으며 이들이 어떻게 관계를 맺고 있는가에 대한 정적인(static) 구조 정보를 표현한다. 요구사항 분석으로부터 유스케이스가 도출되면, 각 유스케이스를 구현하기 위해 시스템을 구성하는 클래스, 그리고 각 클래스가 가지는 속성(attribute) 및 동작(operation)을 추출해야 한다. 그리고 추출한 클래스 사이의 연관관계(association)를 도출해야 한다. 이러한 분석 결과는 (그림 2)의 예와 같은 클래스 다이어그램으로 나타낼 수 있다. 그림에서는 각 클래스의 속성 및 동작 정보는 생략하였다.

시스템을 어떻게 분석하는가에 따라 서로 다른 클래스 다이어그램이 도출될 수 있다. 본 논문에서는 전체 시스템을 관리하는 클래스와 통신을 담당하는 클래스, 그리고 각 유스케이스에 해당하는 주요 하위 작업을 수행하는 클래스로 시스템을 분석하였다. 프로그램 상에서 각각의 클래스는 객체화(instantiation)되어 다른 객체의 함수를 호출하거나 메시지를 주고받으면서 동작한다. 각 클래스가 가지는 상태(state) 및 동작(behavior) 모델링에 대한 정보는 상태기계 다이어그램을 이용하여 표현하게 된다.

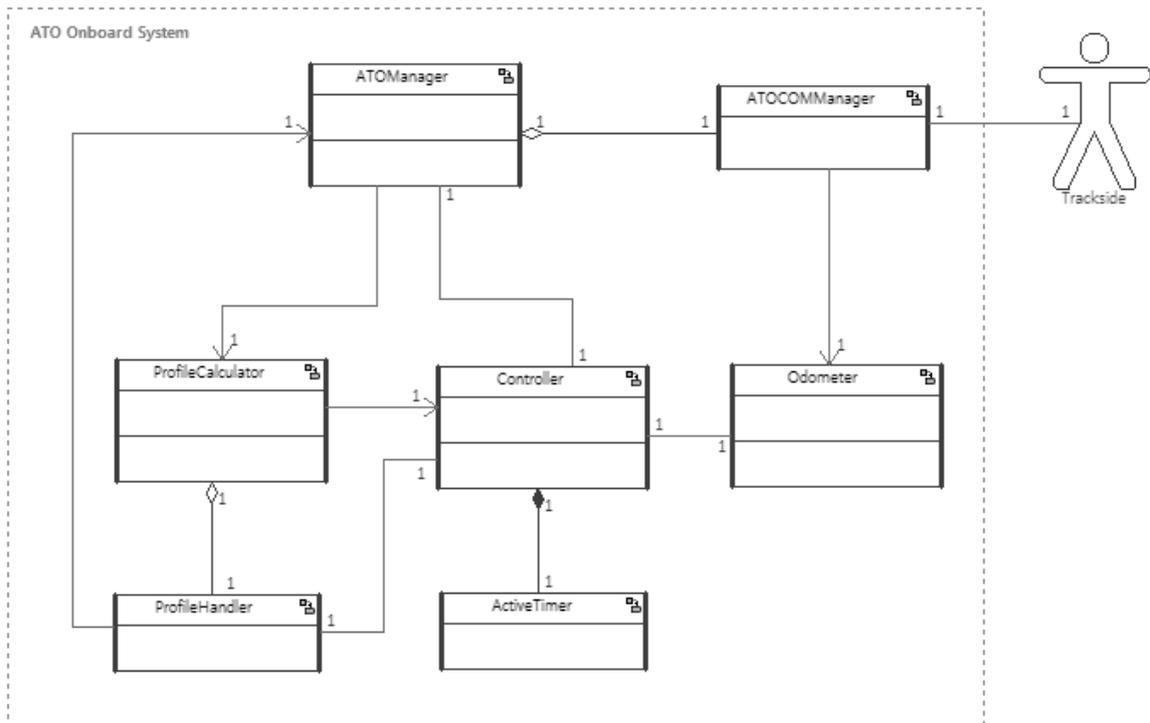


그림2. ATO 차상장치의 클래스 다이어그램의 예

2.1.3 ATO 차상장치의 상태기계 다이어그램

앞에서 도출한 클래스의 행위(behavior)를 분석하고 정의하기 위해 상태기계 다이어그램을 사용할 수 있다. 상태기계 다이어그램에서, 클래스의 객체는 한 시점에서 여러 가지 상태(states) 중 한 가지 상태를 가지게 되며 활동하는 중에 다른 객체와 상호작용을 하면서 상태를 바꿀 수 있다.

객체간의 상호작용은 이벤트(event)를 통해 이루어진다. 한 객체에서 다른 객체로 이벤트를 전달하게 되면, 이벤트를 수신한 객체는 현재의 상태에 따라 그에 대한 반응적인 행동을 취한다. (그림 3)은 ATO 차상장치에서 운영 흐름을 관장하는 역할을 하는 클래스의 상태기계 다이어그램의 예이다. 사각형은 상태를, 화살표는 상태의 전이(transition)를 나타낸다. 화살표에는 한 상태에서 다른 상태로의 전이가 가능한 조건(guarding condition)과 전이를 촉발하는 이벤트(triggering event), 전이할 때 객체가 취하는 행동(action)을 ‘이벤트 [조건] / 행동’의 형태로 명기할 수 있다(각각의 요소들이 필수적인 것은 아니다). 객체는 현재 상태에서부터 가능한 여러 전이를 감시하면서, 특정 화살표에 대해 명기된 조건이 만족되고 자신의 이벤트 큐(event queue)에 적절한 이벤트가 들어오면 그에 따른 행동을 취하면서 화살표 방향으로 상태 전이를 하게 된다. 상태 전이 중의 행동에는 자신의 멤버 함수 호출이나 연관관계를 맺고 있는 다른 객체의 함수 호출, 또는 다른 객체로의 이벤트 전달 등이 포함될 수 있다.

상태기계 다이어그램은 상호작용하는 객체들의 순차적인(sequential) 또는 반복적인 작업 흐름이나 모드(mode)의 변환 등을 표현하는 데 유용하게 쓰일 수 있다. 아래의 상태기계 다이어그램에서 상태 전이는 ATO 차상장치가 보이는 일련의 동작을 기반으로 하고 있다. ATO 차상장치는 역에서 다음 역까지의 운행을 계획하고, 열차를 출발시키며, 속도를 적절히 제어하여 다음 역에 열차를 정차시키는 순서로 사

이클을 돌게 된다. 이 과정에서 이 클래스의 객체는 운영 흐름에 따라 속도제어기 등의 다른 객체들에 적절한 행동을 하도록 이벤트를 보낸다. 그리고 각 객체가 작업이 끝나면 보내오는 확인 이벤트를 수신하여 자신의 상태를 바꾸면서 다른 객체에 이벤트를 통해 명령을 내린다.

Rhapsody와 같은 CASE 도구를 사용하면 상태기계 다이어그램으로부터 각 클래스의 동작에 대한 기본적인 소스코드를 생성할 수 있다. 다만 다이어그램으로부터 도출되는 시스템의 동작에 정형성(formality)이 보장되지 않는 경우가 있으므로 이에 유의할 필요가 있다.

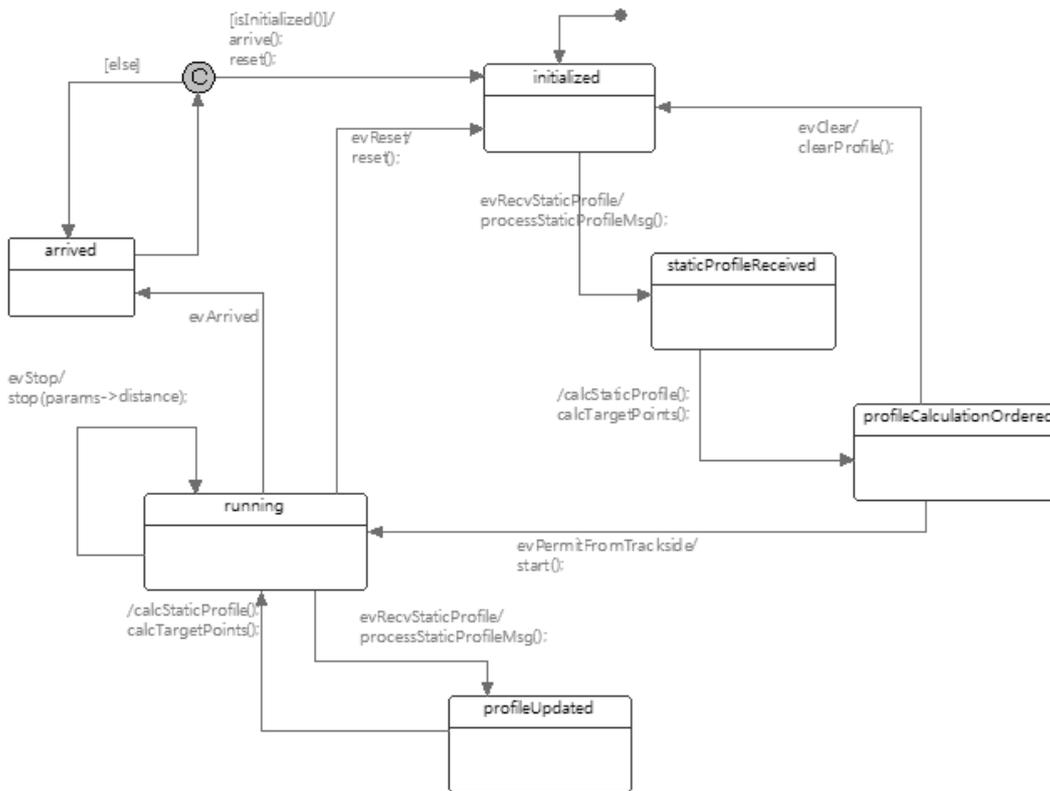


그림3. ATO 차상장치의 상태기계(state machine) 다이어그램의 예

2.2 CASE 도구를 활용한 ATO 차상장치의 구현

전술한 다이어그램 이외에도 액티비티 다이어그램, 시퀀스 다이어그램 등 여러 다이어그램을 사용하여 ATO 차상장치와 같은 시스템을 분석하고 설계할 수 있다. Rhapsody와 같은 UML 기반의 CASE 도구를 사용하면 여러 다이어그램을 통해 시스템을 모델링할 수 있을 뿐 아니라, 도출된 모델의 시뮬레이션을 통해 분석/설계의 결과와 CASE에서의 모델이 보이는 동작이 일치하는지를 확인해 볼 수 있다.

또한, CASE 도구는 모델을 이용하여 기본적인 소스코드를 생성하는 역할까지 수행할 수 있다. 여기서 생성하는 소스코드는 모델에서 명기된 클래스/객체 및 그 동작을 구현하는 일종의 소프트웨어 템플릿으로, 개발자는 핵심적인 알고리즘 부분만을 따로 작성함으로써 소프트웨어를 구현할 수 있고 알고리즘

부분을 교체함으로써 같은 동작을 보이면서 다른 기능이나 성능을 보이는 소프트웨어를 쉽게 만들 수 있다. 이를테면 개발자는 속도제어기 클래스에서 속도 제어 알고리즘만을 따로 작성하여 제어기를 구현할 수 있으며, 알고리즘을 교체하여 쉽게 (동일한 역할을 하는) 다른 제어기를 구현할 수 있다.

본 논문에서는 모델에 속도 프로파일 생성 및 속도 제어 알고리즘을 추가하여 ATO 차상장치의 기본적인 주행관리 기능을 구현하였다. (그림 4)는 모델에 알고리즘을 탑재하여 ATO 차상장치 소프트웨어를 간략히 구현한 후 시뮬레이터를 주행시킨 테스트 결과이다. Rhapsody 상에서 추가한 알고리즘을 교체하면 모델을 고치지 않고도 주행 성능 개선 작업을 수행할 수 있다.

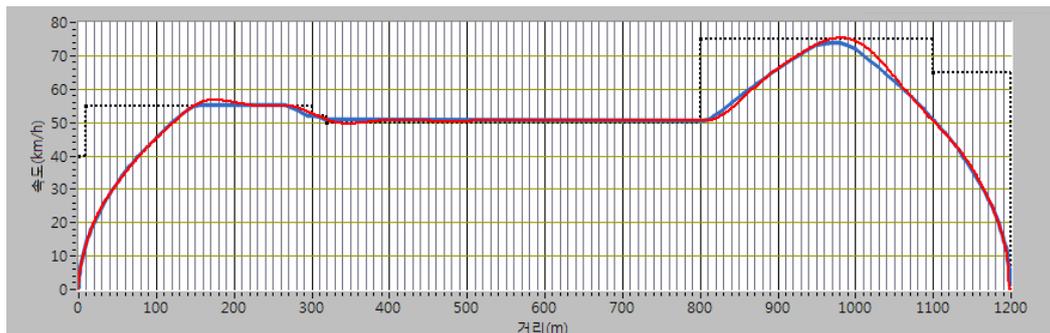


그림4. ATO 차상장치의 프로파일 생성 및 속도추종 시뮬레이션 결과 예

3. 결 론

본 논문에서는 UML 모델링 도구를 이용하여 ATO 차상장치의 소프트웨어를 분석/설계하고 구현한 예를 보였다. UML 기반의 CASE 도구인 Rhapsody를 사용함으로써 시스템을 쉽게 분석하고 설계할 수 있었으며, 모델을 이용하여 기본적인 코드를 생성하고 주요 알고리즘을 추가하는 과정을 통해 소프트웨어를 구현할 수 있었다. 이렇게 구현한 소프트웨어는 CASE 도구에서의 시뮬레이션을 통해 정확한 동작이 일어나는지를 확인해 볼 수 있으며, 알고리즘의 교체나 개선 또한 수월하게 이루어질 수 있다.

객체지향 방법론에 따라 실시간 임베디드 소프트웨어를 개발할 수 있는 UML 기반 모델링 도구를 이용하면, 철도 관련 소프트웨어 개발을 더욱 수월하게 진행할 수 있다. 다만 철도에서는 소프트웨어의 신뢰성과 안전성이 매우 중요한 만큼, 객체지향 방법론을 적용하였을 때의 검증 기법이나 여러 성공 사례를 발굴할 필요가 있다. 향후에는 객체지향 방법론을 통해 개발하는 ATO 차상장치의 신뢰성을 검증하기 위한 방법에 대한 연구가 필요할 것이다.

참고문헌

- [1] "UML 2.0 Superstructure Specification," OMG(2004)
- [2] "IEEE standard for CBTC Performance and Functional Requirements," IEEE(1999)
- [3] "C++ Tutorial for Rhapsody", Telelogic(2007)
- [4] B.P.Douglass(2000), "Real-Time UML," 2/ed, Addison Wesley Longman
- [5] B.P.Douglass(2007), "Real-Time UML Workshop for Embedded Systems", Elsevier