

자율주행 시스템의 시뮬레이션을 위한 소프트웨어 아키텍처 설계 연구

허윤아
건국대학교

hyoona1202@naver.com

A Study on Designing a Software Architecture for Simulation of Autonomous Driving System

Yoona Heo
Konkuk University

요 약

시뮬레이션 기술은 실제 시스템을 그대로, 혹은 간소화하여 모방하는 기술이다. 자율주행 시스템을 실제 도로에서 테스트 하는 것은 많은 비용을 요구하고 큰 위험 부담이 있으므로, 이때 시뮬레이션 기술은 비용 문제와 안전 관련 문제에 효과적인 해결책이 될 수 있다. 본 논문에서는 자율주행 시스템의 시뮬레이션을 위한 소프트웨어 시스템을 개발하기에 앞서, 자율주행 시스템에 대한 소프트웨어 아키텍처를 설계한다. 이를 위하여 사전에 일부 기능 및 특성이 간소화된 소프트웨어 요구사항을 도출하였으며, 이 요구사항 문서를 기반으로 하여 소프트웨어 아키텍처를 설계한다. 또한 설계한 아키텍처를 기존의 자율주행 시스템의 아키텍처와 비교하여 고찰한다.

I. 서 론

자율주행 시스템 (Autonomous Driving System, ADS)은 최근 10 년 이상의 기간동안 많은 사람들의 관심을 받으며 성장해 왔다 [1]. 자율주행 시스템을 실제 도로에서 테스트 하는 것은 위험 부담이 크고 많은 비용이 든다. 이때, 시뮬레이션 기술을 사용하는 것은 이러한 안전과 관련된 문제와 비용 문제에 효과적인 해결책이 될 수 있다. 시뮬레이션은 실제 시스템을 그대로, 혹은 간소화하여 모방하는 기술 [2]로, 다양한 시나리오를 빠르게, 더 적은 비용으로, 동일한 조건에서 반복적으로 테스트할 수 있도록 한다.

자율주행 시스템의 시뮬레이션을 위한 소프트웨어 시스템의 개발을 위해, 본 논문에서는 개발하고자 하는 시뮬레이션 시스템에 포함될 자율주행 시스템에 적합한 아키텍처를 설계한다. 이 시스템의 요구사항 분석, 설계 및 구현은 추후 시뮬레이션을 위한 응용 소프트웨어에서 활용하기 위해 수행한다. 요구사항 작성 시에는 시스템의 일부 기능 및 특성을 단순화하여 진행하였다.

다음 절에서는 분석한 요구사항 중 일부를 설명하고, 아키텍처 설계에 거친 과정을 표현하고, 다이어그램으로 그 결과를 보인다. 마지막으로, 설계한 아키텍처를 기존의 자율주행 시스템의 아키텍처와 비교하여 고찰한 결과를 보인다.

II. 본 론

1) 요구사항의 분석

자율주행 시스템의 시뮬레이션을 위하여, 우선적으로 소프트웨어 요구사항 문서를 작성하는 과정이 필요하다. 이때 요구사항 문서는 관련 표준 양식 [3]에 따라 작성되었

으며, 이 중 자율주행 시스템이 만족해야 하는 품질 요구사항은 다음과 같다:

- 실시간성: 시스템은 센서, GPS 장비와 V2X 통신을 통해 실시간으로 환경 정보를 수집하여 경로 상에 교통사고 등의 비상 상황이 발생했음을 확인하고, 즉시 경로를 변경하는 등의 대응을 할 수 있어야 한다.
- 보안성: 시스템이 V2X 통신을 통해 도로 인프라 또는 다른 자율주행 차량과 통신할 때 외부 시스템 또는 사용자로부터 공격이 들어오면 이를 감지하고 차단할 수 있어야 한다.
- 안전성: 시스템은 일부 기능에 장애가 발생하거나 부품이 고장 나더라도 정상적으로 안전하게 동작할 수 있어야 한다.
- 신뢰성: 시스템은 시동이 완전히 꺼지기 전까지 일관적으로, 고장이나 오류 없이 동작해야 한다.
- 정확성: 주행 중에 수집한 데이터의 인식과 수행하는 제어가 99.99% 이상의 정확성을 가지도록 동작해야 한다.
- 확장성: 시스템은 추후 새로운 기능이 생겼을 때 다른 기능에 영향을 주지 않고 이를 추가할 수 있어야 한다.

2) 아키텍처 설계

본 논문에서는 [4]와 [5]에서 설명하는 Attribute-Driven Design (ADD) 3.0 을 따라 아키텍처를 설계하나, 분량의 문제로 일부 내용과 과정을 생략한다. 품질 요구사항 이외에 기능 및 비기능 요구사항과 설계 목적 또한 아키텍처 설계에 영향을 미칠 수 있으나 (Architectural Driver, AD), 본 논문에서는 분량 상의 이유로 품질 요구

사항을 가장 우선시하여 진행하여 나머지 AD 에 대한 내용은 포함하지 않는다.

아키텍처 설계 시에는 설계 컨셉을 하나 혹은 그 이상을 선택할 수 있고, 설계 컨셉의 종류로는 참조 아키텍처 (Reference Architecture), 아키텍처 설계 패턴, 배포 패턴 (Deployment Pattern), 전술 (Tactics), 그리고 외부에서 개발된 컴포넌트 등이 있다. 본 논문에서는 참조 아키텍처와 배포 패턴을 적용하였다.

참조 아키텍처의 예시로는 계층형 아키텍처, 모듈형 아키텍처, 서비스 지향 아키텍처, 분산형 아키텍처, 이벤트 기반 (Event-Driven) 아키텍처 등이 있고, 여러 참조 아키텍처를 결합하여 사용하는 하이브리드 아키텍처도 있다. 본 논문에서는 실시간성, 안전성, 확장성, 신뢰성, 보안성의 다섯 가지 품질 요구사항들을 고려하여 모듈형 아키텍처, 이벤트 기반 아키텍처, 그리고 분산형 아키텍처의 세 가지 참조 아키텍처를 결합한 하이브리드 아키텍처를 선택하였다.

모듈형 아키텍처는 독립적인 기능 단위로 시스템을 나누어 각 모듈이 역할을 수행하도록 하는 아키텍처로, 확장성을 만족시킬 수 있다. 다음으로 이벤트 기반 아키텍처는 이벤트를 중심으로 하여 이벤트가 발생할 때마다 이를 처리하는 컴포넌트가 작동할 수 있도록 하는 아키텍처로, 실시간성과 확장성을 만족시킬 수 있다. 마지막으로 분산형 아키텍처는 시스템의 기능이 여러 노드에 분산되어 있어 각 노드가 독립적으로 작동하면서 상호 협력할 수 있도록 하는 아키텍처로, 신뢰성과 확장성, 그리고 보안성을 확보할 수 있다.

다음으로 배포 패턴을 설정할 수 있는데, 이는 인프라 요소를 식별하고 명세하기 위한 패턴을 의미한다. 본 논문에서는 정확성과 안전성을 만족시키기 위하여 그 중 Active-Active 배포 패턴을 선택하였다. 이 패턴은 두 개 이상의 인스턴스에서 동시에 서비스를 운영하여 고가용성을 유지하는 방식으로, 한 인스턴스에 문제가 발생하더라도 다른 인스턴스가 즉시 트래픽을 처리할 수 있기 때문에 정확성 뿐 아니라 안전성까지도 보장할 수 있다.

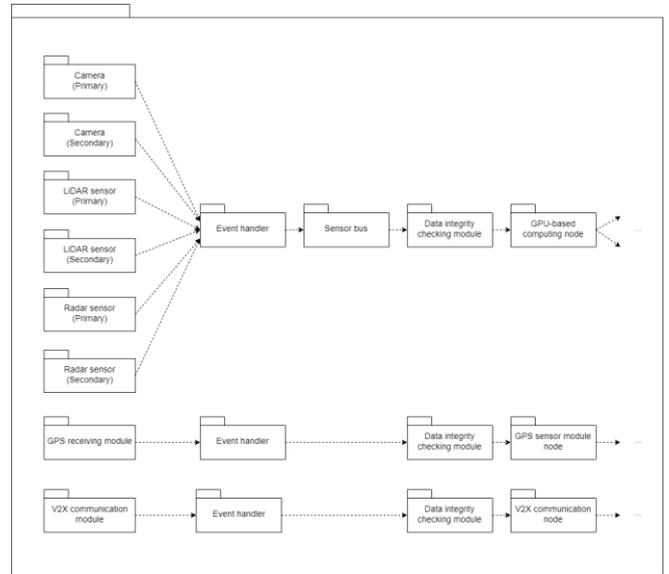
UML (Unified Modeling Language) 형식을 빌려 앞서 언급한 아키텍처들을 결합하여 그린 전체 아키텍처의 모듈 뷰 (view) 중 일부를 표현하면 <그림 1>과 같은 다이어그램으로 나타낼 수 있다. <그림 1>에서 센서 단에는 Event handler 과 Sensor bus 를 두어 각 센서로부터 값을 입력 받아 이벤트가 발생하는 경우 event handler 가 동작하도록 한다. 센서 값들이 sensor bus 로 입력되면, 입력 받은 센서 값들의 데이터 무결성을 확인한 후 컴퓨팅 노드로 입력한다. 이외에 GPS 수신 모듈과 V2X 통신 모듈에 대해서도 각각 event handler 를 사용하고 데이터 무결성을 확인하여 이를 관련된 노드에서 처리할 수 있도록 한다. 이외의 아키텍처에 대한 다이어그램은 분량의 문제로 생략한다.

III. 결론

본 논문에서는 자율주행 시스템의 시뮬레이션을 위한 응용 소프트웨어를 개발하기 위해, 사전에 작성한 요구사항을 활용하여 자율주행 시스템을 간소화한 시스템의 아키텍처를 설계하였다. 설계한 아키텍처는 모듈형, 분산형, 이벤트 기반의 아키텍처이며, 중복성을 가진다.

일반적인 경우에, 자율주행 시스템의 아키텍처는 크게 인식, 계획, 제어를 위한 부분으로 구분되며, 환경을 인식하기 위한 센서, GPS 등의 모듈에서 수집한 데이터를 가지고 계획을 위한 부분으로 직접적으로 전송한다. 계획을 세우기 위한 부분은 이를 일괄적으로 처리하여 제어를 위한 부분으로 제공하고, 조향, 가속, 감속, 제동 등을 위한 모듈들을 제어하여 차량을 운행할 수 있도록 한다.

본 논문에서 진행한 아키텍처의 설계는 분산형 아키텍처,



<그림 1> Portion of Module View

이벤트 기반 아키텍처, 모듈형 아키텍처, 그리고 active-active 배포 패턴의 도입을 통해 요구사항 분석 시에 도출된 품질요구사항을 만족시켰다. 자율주행 시스템은 처리하는 데이터의 양이 매우 방대하고, 실시간으로 처리하고 제어해야 하는 등의 특성이 중요한데, 이와 같은 아키텍처의 설계는 이러한 측면을 잘 고려했다고 볼 수 있다. 단, 여러가지 AD 중 품질 요구사항만을 고려하였다는 점에서 소프트웨어의 전체를 잘 다루었다고 하기에는 어렵다. 연구의 확장을 통해 아키텍처 설계의 과정을 반복하여 소프트웨어의 전반적인 부분을 고려한 아키텍처를 설계할 필요가 있다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-학·석사연계 ICT 핵심인재양성 지원을 받아 수행된 연구임 (IITP-2024-RS-2020-II201834)

참고 문헌

- [1] J. Zhao, et al., "Autonomous driving system: A comprehensive survey," Expert Systems with Applications, vol. 242, no. 122836, 2024.
- [2] P. Bratley, et al., "A Guide to Simulation," Springer Science & Business Media, 2011.
- [3] IEEE, "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998, pp.1-40, 1998.
- [4] L. Bass, et al., "Software Architecture in Practice," Addison-Wesley Professional, 4th ed.002C 2021.
- [5] H. Cervantes, R. Kazman, "Designing Software Architectures: A Practical Approach," Addison-Wesley Professional, 2nd ed., 2024.