

# 위성 탑재 소프트웨어를 위한 Reconfigurable Software Architecture

신현규, 천이진  
한국항공우주연구원  
e-mail : hkshin@kari.re.kr

## Reconfigurable Software Architecture for Satellite Flight Software

Hyun-Kyu Shin, Yee-Jin Cheon  
KARI

### 요 약

위성 탑재 소프트웨어는 기본적인 위성의 상태 데이터 획득에서부터 지상국과의 송수신 및 위성 고유의 임무 수행의 전 과정을 담당하고 있다. 이러한 위성 탑재 소프트웨어에 있어 무엇보다 강조되는 점이 신뢰성이며, 이를 위한 많은 연구가 진행되어 왔다. 위성 탑재 소프트웨어의 개발 과정에서 여러 단계의 검증 및 테스트가 수반되게 되며, 이러한 위성 탑재 소프트웨어의 개발에는 많은 시간과 노력이 요구된다. 또한, 위성의 소프트웨어는 그 특성상 위성 발사 후 탑재 소프트웨어에 대한 수정 및 개선에 많은 어려움이 따르게 된다. 본 연구에서는 위성 탑재 소프트웨어 개발 과정에서 재사용성을 높이고 소프트웨어 및 위성 임무 변경에 보다 용이하게 대응할 수 있는 Reconfigurable Software Architecture 를 제안한다.

### 1. 서론

위성 탑재 소프트웨어는 위성 전체 시스템의 동작을 제어하며, 지상으로부터의 명령 수신 및 위성 데이터의 전송, 위성 임무 수행에 관여한다. 위성 소프트웨어에 문제가 발생하게 되면, 최악의 경우 위성의 본연의 임무를 수행하지 못하여 존재 가치를 잃을 수도 있다. 이렇듯 막중한 임무를 가지고 있는 위성 소프트웨어의 개발에는 가장 기초적인 단위 테스트에서부터 전체 시스템 테스트에 이르기까지 여러 단계의 검증 및 테스트를 수행하게 된다. 이는 혹시 있을지 모를 소프트웨어의 문제점을 발견하고 소프트웨어의 정상 동작을 검증하는 것으로 이에 많은 시간과 노력이 요구된다.

본 연구에서는 위성 탑재 소프트웨어 개발 과정에 있어 소프트웨어의 재사용성을 높이고, 궤도상에서의 임무 변경에 보다 쉽게 대응할 수 있는 Reconfigurable Software Architecture 에 대해 알아본다.

### 2. 위성 탑재 소프트웨어의 변경

위성에 탑재되는 소프트웨어는 일단 발사가 되면 소프트웨어의 수정이 매우 어려운 것이 현실이다. 코드 패치 등의 방법을 통해 위성의 소프트웨어를 변경하거나, 소프트웨어 동작에 관련된 여러 매개 변수의 값을 재설정하기도 한다. 이러한 방법들은 발사 전에 이른바 “하드 코딩”된 내용을 제한적으로 수정하는데 그치게 된다.

일반적으로 임베디드 시스템은 프로세서의 성능,

메모리의 용량 등 하드웨어적인 제약이 일반 소프트웨어 작동 환경에 비해 심하게 된다. 특히 위성 소프트웨어가 동작하는 위성 탑재 컴퓨터의 경우, 우주 환경적인 특성에 따른 요구 사항에 부합하기 위해 더 많은 제약을 가지게 된다.

따라서 지금까지의 위성 소프트웨어는 이러한 특수성으로 인해 제한된 환경에서 주어진 요구 사항을 구현하기 위한 형태로 개발되어 왔다.

### 3. 위성 개발 환경 및 임무의 변화

기술의 발전을 통해 위성에 탑재되는 탑재컴퓨터 역시 그 성능이 비약적으로 향상되고 있다. 이는 위성 소프트웨어의 개발에 있어 보다 많은 선택의 기회를 주게 된다.

더불어 위성에 요구되는 임무 역시 더욱 복잡해지고 있다. 위성에 탑재되는 탑재체의 종류가 다양화되고, 한 위성에 여러 종류의 탑재체가 탑재되는 다목적 또는 복합 위성은 보다 다양한 임무를 지니게 되고, 이를 지원하기 위한 소프트웨어 역시 복잡해지게 된다.

### 4. OBCP

일부 위성의 경우, OBCP(On-board Control Procedure)를 통해 이러한 변화에 대응하고 있다. 일반적으로 OBCP 는 위성 임무 수행의 자동화를 위해 탑재 컴퓨터에서 동작하는 소규모의 프로그램을 의미한다.

OBCP의 기본 개념은 그림 1과 같다.<sup>[1]</sup>

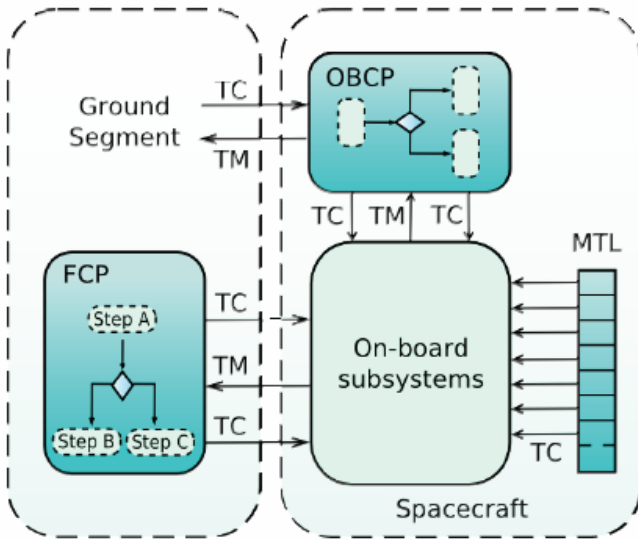


그림 1 OBCP의 기본 개념

지상에서 컴파일되어 탑재 컴퓨터로 전송된 OBCP는 탑재 소프트웨어에 의해 관리되며, 탑재 소프트웨어는 OBCP의 로드, 스케줄링, 실행 및 모니터링을 담당한다. 이러한 처리 과정은 동시에 여러 개의 OBCP를 비동기적으로 처리하는 것이 가능하며, 대부분의 OBCP는 지상 명령이나 저장된 임무 명령(Stored Command)에 의해 시작되게 되며, 일부 OBCP는 다른 OBCP나 탑재 소프트웨어의 처리 과정의 일부로 시작되기도 한다.

이러한 OBCP는 다음과 같은 장점을 갖는다.<sup>[1]</sup>

- 지상국과의 교신이 없는 상태에서도 임무 수행을 위한 향상된 기능을 제공
- 위성의 문제 발생시 보다 즉각적인 대처가 가능
- 복잡도를 감소시키고 위성 운영의 효율성을 증대
- 지상국과의 교신 범위 외에서도 안전한 임무 수행이 가능
- 탑재 소프트웨어 자체의 변경 없이 탑재 소프트웨어의 기능을 변경할 수 있는 메커니즘을 제공

### 5. Reconfigurable Software Architecture

본 연구에서 제안하는 Reconfigurable Software Architecture(이하 'RSA')는 앞에서 알아본 OBCP를 보다 확장한 개념이다. 일반적으로 위성 소프트웨어는 위성의 안전 및 운용에 있어 가장 기본이 되는 부분을 포함하는 버스 파트와 탑재장치를 통한 임무 수행을 위한 미션 부분으로 나누어 볼 수 있다. 앞의 OBCP는 탑재 소프트웨어의 주요 기능을 모두 구현하고, 여기에 OBCP를 해석하고 관리하는 기능을 추가한 형태로, 주로 위성의 페이로드를 통한 임무 수행에 쓰이게 된다. RSA는 이를 좀더 넓게 적용하여 위성 탑재 소프트웨어의 많은 부분을 동적으로 관리

하고자 한다.

또한 OBCP는 OBCP를 제외한 나머지 부분에 대한 변경에 대해서는 기존의 소프트웨어 같은 제약을 갖게 된다. 이에 RSA는 ACP에 의해 참조되는 기본 기능에 대한 변경 가능성을 고려하였다.

RSA의 기본 구조는 그림 2와 같다.

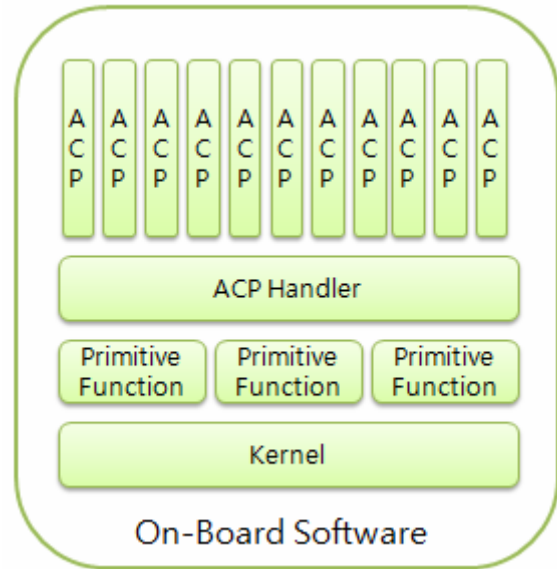


그림 2 RSA의 기본 구조

- Kernel  
위성 탑재 소프트웨어의 가장 기본이 되는 부분으로 위성 소프트웨어의 초기 기동 및 Task 관리를 수행한다.
- Primitive Function  
ACP에 의해 불러지는 기능 단위로 이 역시 추가 될 수 있다.
- ACP Handler  
ACP를 관리하는 역할을 수행하며, ACP의 로드, 스케줄링, 실행 및 모니터링을 담당한다.
- ACP  
탑재 소프트웨어의 실제적인 수행을 위한 코드들로서 Primitive Function들의 조합으로 이루어진다.

이러한 유연한 RSA의 구조는 위성 탑재 소프트웨어의 개발에 있어, 재사용성을 향상시킬 수 있다. RSA 구조를 효율적으로 적용하기 위해서는 무엇보다 먼저 Primitive Function을 정의해야 하는데, 잘 정의된 Primitive Function은 소프트웨어 개발을 용이하게 한다. 더불어 소프트웨어 전체의 신뢰도 향상도 기대할 수 있다. 이는 RSA가 철저하게 테스트된 기본 기능 위에서 이를 이용한 Application Logic의 구현을 전제로 하고 있기 때문이다.

또한 RSA는 Primitive Function의 관리 기능을 포함한다. ASP의 관리에 비해 보다 엄격한 룰이 적용되나 Primitive Function에 대해서도 동적인 추가를 가능케

함으로써 OBCP 에 비해 위성 탑재 소프트웨어의 변경을 용이하게 하였다.

## 6. 장단점

RSA 는 신뢰도 높은 모듈(Primitive Function)을 이용하여 동적으로 동작하는 ASP 를 도입함으로써 기존 OBCP 가 가지고 있는 장점을 모두 흡수, 확장하였다. 또한 Primitive Function 에 대한 관리 부분을 추가하여 거의 대부분의 위성 탑재 소프트웨어에 대한 접근 가능성을 열어두고 있다.

신뢰도 높은 모듈 및 이를 이용한 ASP, 그리고 RSA 의 형태는 이후의 위성 소프트웨어 개발에 재사용될 수 있다. 이러한 신뢰도 높은 모듈의 재사용은 소프트웨어 개발 과정에 있어 구현 및 테스트 비용을 절감할 수 있다.

RSA 는 동적으로 작동하기 때문에 기존의 탑재 소프트웨어에 비해 동작 속도가 저하될 수 있으나 점차 향상되고 있는 탑재 컴퓨터의 성능을 고려하면 크게 문제가 되지는 않는다. 다만, RSA 가 성공하기 위해서는 신뢰도 높은 Primitive Function 을 갖추는 것이 전제되어야 한다.

## 7. 결론

기존의 위성 탑재 소프트웨어에도 궤도 상에서 소프트웨어를 변경하기 위한 기능 및 방법이 존재한다. 특히 OBCP 는 보다 동적으로 위성의 임무 수행을 제어할 수 있는 방안을 제공하고 있다. RSA 는 이러한 기능을 보다 확장하여 위성 탑재 소프트웨어의 기능을 분리, 잘 정의되고 테스트 된 기본 기능을 기초로 하여, 임무 수행을 위한 ASP 를 작성, 활용토록 한다. 이를 통해 발사 후 발생될 수 있는 소프트웨어의 변경에 대응하며, 보다 유연한 임무 수행을 처리할 수 있다. 또한 이러한 시도를 통해 소프트웨어 개발 비용을 절감하고, 재사용성을 증가시킬 수 있다.

## 참고문헌

- [1] F.Trifin, "Simplifying On-Board Control Procedure Development: A Generic Tool Based on ESOC Experience"
- [2] W. Heinen, "Automation through On-Board Control Procedures: Operational Concepts and Tools"
- [3] Stuart James Cater, "Satellite Test and Operation Cost Reduction Through Standardization"
- [4] Andrew Monham, "Reducing Risks and Costs in Automated Operations"
- [5] C.Steiger, "OBSM Operations Automation through the use of On-board Control Procedures"