



Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

<http://www.kkits.or.kr>

A Universal Middleware-based Small Satellite Module for Dynamic Service Platform

Hae-Jun Lee*

Satellite Technology Research Center, Korea Advanced Institute Science and Technology

ABSTRACT

Recently, In the wake of the satellite service industry research has been spreading to convergence science. As a representative field of cross-platform technology lead to user driven approach about space application services. Universal Middleware could be make pre-design review step of small satellite unit development. It could be develop to easy integration for user application services between internal interface module and 3rd party's information services. In this study presents extensible way between each module that consist of dynamic and correlation application services for small satellite development model unit. This model could be adopt to the on-board computer interfaces such as a satellite control data field unit, define function and etc. This interface module transfer a data and service process for applying to each unit of intelligent requirement and development parameter. The condition of requirements are proposed with public protocol and private interface module with variable simulation way. Consequently, look into the extensible and dynamic module of three structural model for a small satellite system to the on-board computer. Firstly, the shared instance model of dynamic module support between service registry and inventory that consist of Satellite Application Platform. It suggested interoperability process of connect to tracking and event handling. Secondly, it should be make sure about the way of dynamic module that private message model. It could be make figure out the function of each relation. Thirdly, it take a real time application load model with application launcher via extended public protocol and network inventory.

© 2019 KKITS All rights reserved

KEYWORDS : Universal middleware, Small-sat platform, Reusable service, Satellite simulation, Dynamic module design

ARTICLE INFO: Received 1 October 2019, Revised 27 October 2019, Accepted 7 December 2019.

*Corresponding author is with the SaTReC, KAIST, 291 Daehak-ro Yuseong-gu Daejeon, 35408, KOREA.

E-mail address: haejun@kaist.ac.kr

1. 서론

위성개발 과정은 국방기술과의 연관성으로 기술의 공유 및 개발비용의 부담이 큰 편이다. 이 때문에 상업적 목적을 위한 재활용 관련 연구의 활성화가 부족하다. 그러나 최근 미국의 NASA, 유럽의 ESA*를 중심으로 ICT기술을 적용하여 개발비용을 최소화 하면서 사용자중심 개발방식과 Application Platform(이하, AP) 재활용 소프트웨어 연구가 증가하고 있다[1]. 이는 위성의 설계, 개발과정, 운용단계에 대해 각각 상업적 활용을 위한 ICBMS**기술과의 융합으로 발전하고 있다[2]. 위성 서비스 상업화를 주도하고 있는 미국과 유럽의 위성기술은 웹과 모바일 서비스 영역까지 확산되는 추세이다.

유니버설미들웨어는 기술적으로 복잡한 모듈간의 통합, 인터페이스 증첩에 의한 개발비용을 절감시킨다. 통신서비스를 위한 다중 프로토콜의 지원과 이기종 시스템 동적 모듈의 Eco-System을 제공한다. AP로 유니버설미들웨어를 적용하면 융합이 용이하며 서비스공급자와 수요자가 유기적으로 연결된다. 이는 위성개발의 목적이 상업적으로 분화하고 확장하며 지속 가능한 서비스를 제공하는 방향성과 부합한다. 상업적 시스템에 독립적인 사용자 주도 접근법으로 서비스모델을 제시할 수 있다[3]. 이와 관련된 유사 연구로는 미국과 유럽에서 추진하고 있는 ESA의 SNAP***과 NASA의 OneWeb**** 프로젝트가 있다. 이들은 위성 영상 어플리케이션인 SAR*****이미지에 대한 동적 서비스를 구축하

여 통신과 응용서비스를 제3의 서비스 수요자에게 동적으로 제공하는 방식이다.

사용자서비스는 수요자 중심으로 위성 내부모듈과 외부모듈로 구분하였다. 내부와 외부 서비스를 통합하고 재활용을 위한 단위모듈을 표준화하여 통신서비스 및 데이터를 Bundle Context로 객체화하였다. 각각의 모듈과 요구 서비스는 수요대상인 사용자와의 연관성을 정의하여 동적으로 처리하였다.

단위모듈은 통신 및 응용서비스와 연동되는 하드웨어의 기능으로부터 정의한다. 핵심모듈을 규격화기 위하여 실제 구현에서는 Dynamic Java에서 지원하는 Context Bundle로 제작하였다. 단위모듈은 인스턴스방식으로 공유하며 다른 모듈에서 재사용할 수 있도록 전용 채널과 네트워크를 통해 실시간으로 처리할 수 있다[4].

논문의 구성으로는 2장에서 본론인 동적모듈화를 위한 공유모델을 설명하였고, 모듈 검증을 위한 Application Client와 AP 구현방법을 설명하였다. 3장에서는 Application Instance에서 시스템자원의 처리과정을 통해 복잡한 시스템의 경우 Application Module간의 Instance 공유모델 구성을 제시하였다. 즉, Instance 종류별로 최종 수요자의 목적에 따라 동적 분류와 네트워크상의 Application Inventory에의 구성 방법을 설명하였다. 제3장에서는 결론을 기술 하였다.

2. 통합 서비스플랫폼 시스템 구조

2.1 시스템 구조

유니버설미들웨어를 적용한 AP는 위성시스템 체계종합에서 분류한 단위모듈을 기준으로 Bundle Context간의 상호작용 프로세스를 기초로 한다. 이때 객체인 Bundle Context에서 Interface를 XML로

* ESA ; European Space Agency, 유럽의 각국이 공동으로 우주개발을 위해 설립한 국제기구

** ICBMS ; 4차산업핵심기술융어로 IoT, Cloud, Bigdata, Mobile, Security를 말함

*** SNAP ; ESA의 SantiNel Application Platform으로 지구관측자원 공유 프로젝트

**** OneWeb ; OneWeb satellite constellation으로 2021년부터 위성인터넷 광대역 서비스 제공예정인 프로젝트

***** SAR ; 합성개구레이더(Synthetic Aperture Radar)로 대기의 기상상황과 상관없이 위성영상이미지를 측정하는 레이더

가상화하여 실행주체인 Activator로 동작시킨다. 유니버설미들웨어상의 Bundle Context와 Activator는 이기종 시스템간의 확장성을 제공하고 위성개발의 복잡한 시스템체계를 사용자 중심의 개념별 Layer로 분리할 수 있다. 하드웨어 Layer에서 드라이버를 Bundle Context화하여 재활용 할 수 있다.

Activator는 응용서비스의 조합과 융합방식으로 구성된 시스템인터페이스, 통신프로토콜의 표준화와 모듈화의 실행 주체이다[5]. 즉, 시스템인터페이스와 통신 프로토콜을 동적으로 모듈화 하는 방식의 가상화가 가능하다. 이 과정에서 위성시스템 내부의 표준인터페이스를 Message로 처리하면 모듈화가 용이하다. Message 처리방식은 AP상에서 외부의 응용서비스와 통신 프로토콜간의 전송 과정에서 외부시스템의 모듈과 상호 공유한다.

<그림 1>은 Dynamic Module로 동작하는 공유모듈인 Shared Module간의 상호작용을 나타내고 있다. AP상의 공유 형태에 따라서 내부 및 외부의 Application을 통합할 때 필요한 서비스 명세서 규칙을 3rd Party와의 규칙을 정의하고 처리방식을 설계해야 한다. 관련하여 <표 1>에서는 위성서비스에 적용한 명세서의 사례로 AP에 영향을 미치는 주요 파라미터들이다.

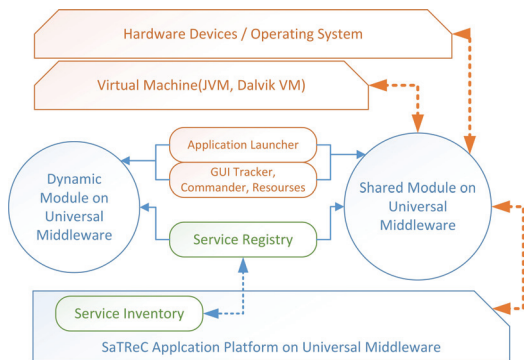


그림 1. 유니버설미들웨어 응용플랫폼 Use-Case 구조

Figure 1. Application Platform Use-Case on Universal Middleware Architecture

표 1. 응용플랫폼의 외부파라미터

Table 1. External Parameter of Application Platform

Application Item	Function of Application Platform Requirement			Workload
Standard	Device Define	Registry	Operation	TBD
Extension	Tracking	Command	GUI Launch	TBD
Manage	Handler	Container	Descriptor	3 rd Party

이 명세서를 서비스모듈에 탑재하기 위해서는 응용서비스 관리가 필요하다. Application Launcher는 표준서비스와 인터페이스를 통해 관리와 추적을 수행한다. Bundle Context에 등록된 저장소인 Registry 정보를 참조하여 Tracker와[6] 내부인터페이스의 명령어 송수신 Commander와 모듈의 자원 관리 정보를 제공한다.

이 Tracker 모듈은 AP에서 Tracking 기능으로 전체시스템의 하드웨어와 서비스 모듈의 이력을 알 수 있다. 이를 응용하면 모듈에 구성된 부품의 도입 과정 뿐만 아니라 개발모듈의 연관성, 단계별 테스트 정보, 위성 발사 과정의 데이터 취득정보를 알 수 있다[7].

<그림 2>는 응용서비스를 탑재한 AP인 SAP는 기초적인 톨로 구현하여 적용한 실행화면을 나타내고 있다.



그림 2. 테스트플랫폼(SAP) 로드 화면

Figure 2. Display of SAP Loading

SAP는 향후 수요자의 요구 서비스의 사례를 받

굴하여 테스트하고 평가할 있도록 구성된 프로그램이다. 수요자 요구서비스 발굴은 플랫폼상의 Service Registry에 등록된 정보에 따른 Bundle Context의 시스템 자원을 재활용 할 수 있도록 서비스를 깨우거나 중지시키는 방식에 기초를 두고 있다.

2.1 위성서비스의 Dynamic Module과 Shared Module의 설계

위성사전개발과 발사 후 운영을 위한 데이터 분석의 경우 표준화 모듈인 Dynamic Module과 Instance를 공유한 Shared Module로 이기종간의 동적 자원을 공유하며 유기적으로 처리할 수 있다[8]. <그림 3>에서는 동적 모듈의 Instance에 대한 Registry, Inventory의 상호작용을 나타내고 있다. 이 구조는 위성 내부 하드웨어 모듈과 응용서비스를 구성하는데 필요하다.

Dynamic Module의 상호작용으로 서비스의 Register, Connection, Find & Install과 Share & Management로 모듈의 연관성을 부여한 Activator 서비스를 적용할 수 있다.

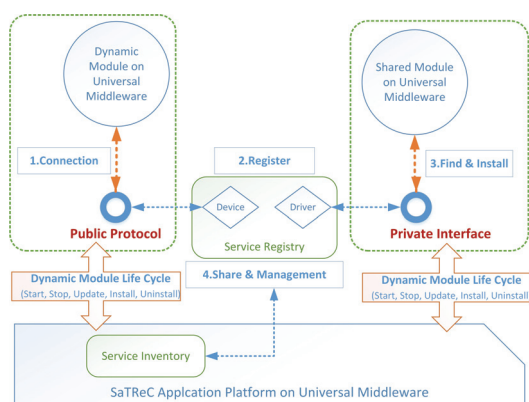


그림 3. 유니버설미들웨어상의 동적모듈 설계

Figure 3. Dynamic Module Design on Universal Middleware

또, Dynamic Module은 유니버설미들웨어에서 동작하는 기본 단위로서 그 기능은 Service Registry와 해당 Registry에 등록된 서비스를 분류하게 된다. Registry에 축적된 서비스목록은 Inventory간의 시나리오 기반의 상호작용 정책의 정의에 필요하다. 새로운 채널을 개폐하고, Inventory 분류를 위해 XML을 통해 외부시스템과의 채널 개폐방식을 정리하여 하드웨어기기와 모듈간의 Instance를 공유한다.

모듈 Instance는 Private Interface방식의 하나인 Private Message로 Application Launcher를 동작 정책에 표준화된 GUI Application 확인을 위한 Tracker와 Commander를 제공한다. AP에 탑재된 대상에게 Resource 변경사항을 적용할 수 있도록 처리한다. Event Handler*와 Tracking Agent는 Mark Up Language에 작성된 명세서를 참조하여 서비스를 공급한다.

이러한 방식으로 복잡한 하드웨어설계과정과 성능테스트, 개발단계별로 다른 환경에서 수행되는 실시간 시뮬레이션, 인터페이스 적용결과, 시스템의 통합과 모듈의 확장성을 제공하여 개발과 운영과정에서의 효율성을 증가시킨다. 또한, 운영목적에 따른 구현 방법은 유니버설 미들웨어상의 동적모듈인 API** 설계를 통해 개발 자원의 배포와 중복사용을 제거하고 시스템자원을 재활용할 수 있다.

결과적으로 동적서비스모듈화와 공유모듈설계 요구사항을 다음과 같이 정리할 수 있다. 첫째, 위성시스템의 다양한 하드웨어기반의 Interface들을 표준화하고 Protocol을 배포 가능한 단위모듈로 설계하여야 한다. 둘째, 위성시스템에서 제공하는 기존의 기능에 대해 유니버설미들웨어의생명주기 형태로 동적모듈 각각의 Instance에 적용요소를 서비스목적에 따라 분류하여야 한다.

* Event Handler ; OSGi Framework의 Event시 발생 분산처리 Listener

** API ; Application Programming Interface, 외부시스템 구현을 위한 인터페이스

Dynamic Module의 경우에는 응답시간에 따라 호출간격을 제어해야 하며 (2)와 같이 Think Time을 가변적으로 적용한다. 호출간격은 절댓값을 산정하여 측정하는데 기준시간에 대한 임의지정상수인 30을 ReqDuration로 지정하였다.

$$TPS = \frac{NL}{ReqDuration} \quad (1)$$

$$(ReqDuration = RespTime + ThinkTime)$$

$$ReqDuration(Max) \geq RespTime + ThinkTime \quad (2)$$

$$\frac{NL(1000)}{ReqDuration}, \quad (3)$$

$$(30 - (RunApplication + RunAppPreview)),$$

$$(30 - (30 - (15 + 1))) = 62.5 \text{ TPS}$$

결과적으로 식(1)과 (2)를 토대로 <표 2>의 데이터를 참고하여 ReqDuration을 기준으로 Run AP Preview에서의 Run Application Time을 추가한 값을 계산한다. 따라서, TPS는 (3)에서와 같이 62.5가 된다.

표. 3 명령어 모듈 실행결과
Table. 3 Result of Application Commander Module

Elapsed time	CPU	Memory	Swap
granularity/Isec	1×10^9	1×10^9	
00.01	14.58	60.746	8.10E+09
00.13(Run Application)	29.365	60.764	8.12E+09
00.24(Run Application)	31.657	60.735	8.11E+09
00.27	33.96	60.716	8.11E+09

<그림 5>와 <표 3>은 Private Message를 적용한 Application Commander방식의 공유모듈 실행 측정결과를 나타내고 있다. 또한, <표 3>의 데이터 TPS는 (4)에서 52.6 TPS로 측정되었다.

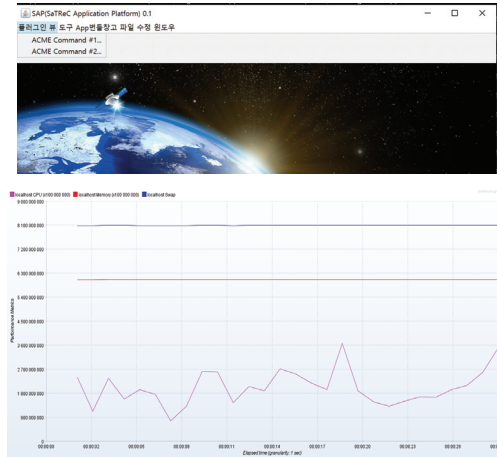


그림. 5 명령어 모듈
Figure 5. Application Commander Module

$$\frac{NL(1000)}{ReqDuration}, \quad (4)$$

$$(30 - (RunApplicationTime)),$$

$$(30 - (24 - 13)) = 52.6 \text{ TPS}$$

2.3 External Application Platform의 시스템 구성

외부 AP와 연동은 위성시스템의 서비스에서 제공하는 데이터의 정확성과 분석단계의 복잡성으로 구분하였다. 외부에서 제공하는 취득데이터를 실시간으로 해석하고 분석단계와 직접 적용할 경우 개발시간을 단축할 수 있다. 또, 데이터의 해석결과에 대해 다수의 응용서비스를 대상으로 모델링하고 비교한다면 분석 결과의 정확성을 향상시킬 수 있다. OnTology*와 연계할 경우 외부시스템을 네트워크로 연동하여 분석하는데 효과적이다. 이는 DAP의 모듈과 Instance를 선택적으로 운영할 수 있다는 점이 중요하다[11].

외부시스템 구성은 다양한 자원을 공급하고 다수의 사용자가 서비스를 요구할 경우에 대응할 수

* OnTology : 개념과 사용상의 제약조건들을 명시적으로 정의한 기술로 시맨틱 웹으로 구현함, RDF, OWL, SWRL 등의 언어를 이용

있는 시스템으로 발전할 수 있다. 그 사례로 SAR*의 영상데이터 분석, 천문데이터, 우주의 관측데이터로 제3의 클라우드 서버와 실시간 연동으로 결과물의 신뢰성과 시간을 절약할 수 있다[12]. 데이터 맵과 형태와 구성법은 요구정보에 대해 데이터 교환규칙을 정의하여 Dynamic Web 3.0기반의 XML을 통해 외부서버의 Module과 연동하는 방식이다.

$$TPS = \frac{NL(1000)}{ReqDuration}, \quad (5)$$

$$(ReqDuration - (RunOnLine)) \times TrafficRatio$$

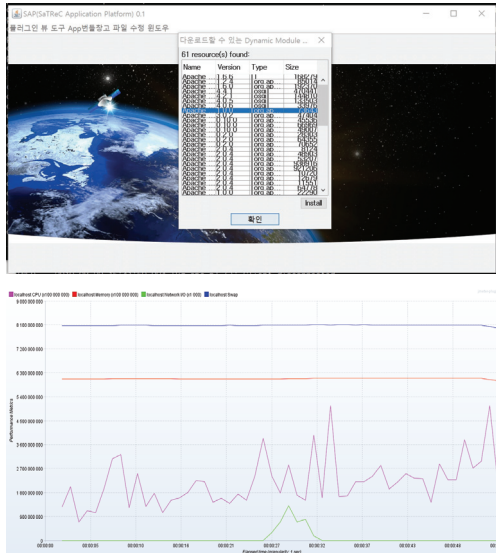


그림 6. 네트워크를 통한 어플리케이션 설치 모니터링
Figure 6. Application Install from Network Inventory by Public Protocol

<그림 6>과 식(5)에서는 제3의 외부시스템으로부터 연동할 수 있는 Application을 로드하여 연동대상이 되는 서버의 Bundle Context 목록을 나타내고 있다. <표 4>에서는 선택된 Bundle Context에 대한

네트워크 트래픽처리와 TPS 사용자원 측정결과를 나타내고 있다.

<표 5>는 서론에 제시한 DAP으로 설계할 경우에 적용모듈에 대한 파라미터이다. 앞서 설명했던 Public Protocol을 사용한 <표 1>의 서비스 명세서이면서 파라미터의 고려사항을 나타내고 있다.

표 4. 네트워크설치 모듈 결과
Table 4. Result of Network Inventory Module

Elapsed time	CPU	Memory	Network
granularity/1sec	1×10^9	1×10^9	
00.00	38.453	60.845	0
00.01(Run On-Line)	24.218	60.855	318780
00.02(Run On-Line)	17.991	60.864	710066
00.03(Run On-Line)	28.514	60.936	1324750
00.04(Run On-Line)	17.154	60.953	699468
00.05(Run On-Line)	15.263	60.958	805448
00.06(Run On-Line)	39.68	61.125	175259
00.07	16.029	61.126	0

그 외에도 Private Interface방식을 활용하여 지원하는 ACME** 커맨드를 통해 커널 시스템 내부의 명령어와 Application을 관리하도록 하여 시스템의 안정성을 제공하였다[13].

표 5. 소형위성 응용플랫폼 파라미터결과
Table 5. Result of Small-Sat Application Platform Parameter

Applications	Function of Application Platform Requirement	Workload
Standard	Private Interface	$n \times NoInt.$ (Interface) 62.5 TPS
	Public Protocol	$n \times NoMsg.$ (Message) 52.6 TPS
Extension	Public Protocol	$n \times No.Pd.$ (Protocol) Depend on Traffic
Management	Future Constants	3 rd Party 3 rd Party

결과적으로 External Application System 구성을

* SAR ; Synthetic Aperture Radar, 합성개구레이더로 위성영상촬영을 위한 탑재체

** ACME : 최상위 명령체계로 시스템관리자가 사용하는 명령어를 의미함.

위한 위성서비스 설계의 시스템요구조건은 Plug-In Command Type을 $n \times No.Msg.$ 의 선형조건으로 나타낼 수 있다. 이에 대한 시스템자원의 안정성은 52.6 TPS를 기준으로 내부시스템인 Internal Application Type에 대해 $n \times No.Int.$ 으로 추측할 수 있다.

$$Traffic = (62.5 - 52.6/62.5 + 52.6) = 0.09 \quad (6)$$

즉, 식(6)에서 시스템자원의 Workload는 62.5 TPS를 기준으로 Dynamic Service Platform에 대해 External Application Type은 $n \times No.Pd.$ 으로 계산할 수 있다. 이 조건을 토대로 파라미터를 적용하지 않을 경우에 비해 선형모델링 Traffic 비용으로 AP를 구성할 경우 $n \times No.Int.$ 대비 $n \times No.Pd.$ 에 대해 0.09로 예측하여 기존 시스템 구축 자원을 재활용하여 설계할 수 있다[14].

3. 결론

본 연구에서는 AP와 Dynamic Module의 상호작용 모델을 활용하여 모듈간의 공유를 통한 재활용 방법에 대해 연구를 수행하였다. 유니버설미들웨어를 적용할 경우 위성 개발에 사용되는 시스템에 비독립적인 모듈간의 복잡한 구조를 간단하게 일원화 할 수 있다. 추가적인 장점으로는 동적서비스 구성과 확장성을 지원하여 Extension Bundle Context 방식으로 설계할 수 있다. 이 방식으로 시스템에 독립적인 AP상에서 각종 데이터를 목적별로 분리하여 서비스의 수요자가 요구하는 방식으로 서비스구성이 가능하다. 이를 위해 위성서비스를 중심으로 외부 시스템에 대해 독립적이며 안정적이며 지속적인 서비스공급의 효율성을 제시하였다. 이때 위성시스템의 하드웨어 파라미터 변수를 응용서비스의 명세서에 적용하였다. 명세서에

OnTology와 같은 사용자 중심 규칙기반 XML로 구성하여 서비스관리와 신규개발이 용이하도록 하였다[15].

본 연구를 통해 상업화 위성 내부시스템과 외부 제공 서비스 구축을 위한 Application간의 유닛화와 표준화 과정이 필요함을 알 수 있었다. 이에 대해 다음과 같이 세 종류의 설계모델을 제시하였다. 첫째, 유니버설미들웨어 기반의 Standard Platform Layer를 구성하였다. 둘째, 확장성을 제공하는 하드웨어의 기능에 대해 Shared Module 라이브러리를 제공하였다. 셋째, 동적라이브러리와 동적서비스를 지원하기 위한 Network Inventory를 XML로 구성하였다. 이러한 구성은 위성본체의 안정성과 AP상의 Payload와 같은 확장성으로 재활용 방식에 효과적이다.

향후 과제로는 AP를 적용하기 위한 실질적인 운영 데이터축적이 필요하다. 예를 들어 SARP 분야의 경우 ESA Santinel 프로젝트에서 적용된 위성 데이터의 AP기반 Use-Case분석의 상업화가 가능한 서비스 발굴이 기대된다. 통신모듈의 경우 SAT5G 망에서 사용가능한 서비스를 대상으로 구체적인 융합 서비스에 대한 연구도 필요하다. 그 외에도 타 위성의 AP와의 연동이나 통합을 위해서는 기존의 복잡한 FPGA Dynamic Module화와 A/D 데이터 변환을 위한 Dynamic Module화가 필요하다. 기존 개발방식을 Open API 방식으로 전환하기 위한 과정이 필요하다. 이를 통해 우주산업분야에서 통신 서비스 산업의 수요자 주도 방식으로 상업적 서비스 발굴을 위한 연구가 확대될 것으로 기대한다.

References

- [1] European space agency, open concurrent design tool, <https://ocdt.esa.int/>, Oct. 2016.

- [2] C. Henning, A. Viehl, B. Kämpgen, and H. Eisenmann, *Ontology-based design of space systems*, International Semantic Web Conference, pp. 308-324, 2016.
- [3] European space agency, esa science & technology, <https://sci.esa.int/rosetta>, Aug. 2015.
- [4] H-J. Lee, C-G. Hwang, and C-P. Yoon, *System visibility of universal middleware pervasive memorial engine*, Journal of the Korea Inst. of Information & Communication Engineering, Vol. 21, No. 11, pp. 2115-2120, 2017.
- [5] OSGi Alliance, initial version of rfc 243 OSGi connect, <https://github.com/osgi/design>, Aug. 2019.
- [6] D. K. Shin, and D. I. Shin, *Single sign-on scheme using XML for multimedia device control in children's game network based on OSGi service platform*, International Journal of Security and Its Applications, Vol. 7 No. 3, pp. 75-84, 2013.
- [7] S. Kim, H-C. Lim, and B. S. Kim, *Conditionals signal-acquisition parameter selection for automated satellite laser ranging system*, Journal of Astronomy and Space Sciences, pp. 97-103, 2019.
- [8] N. H. Hedjazi, and D. Benatia, *Modeling handover in satellite communications*, International Journal of u-and e-Service, Science and Technology, Vol. 7 No. 5, pp. 161-170, 2014.
- [9] H. W. Kim, and Y. S. Ahn, *Modeling handover in application performance evaluation in main memory database system*, Journal of Digital Contents Society, Vol. 15 No. 5, pp. 631-642, Aug. 2011.
- [10] V. Mariappan, M. W. Lee, J. P. Cho, and J. S. Cha, *Onboard vision based object tracking control stabilization using PID controller*, International Journal of Advanced Culture Technology, Vol. 4, No. 4, pp. 81-86, 2016.
- [11] G. P. Shashikala, and S. N. Kini *Survey paper on big data analytics in real time satellite data*, International Journal of Science and Research, Vol. 6, No. 1 pp. 1135-1139 , 2017.
- [12] J. H. Kim, S. M. Lim, and D. H. Cho, *Neural network based on rrbrit cloud coverage estimation technique for intelligence earth observation Nano-satellite*, *Proceeding of Korea Space and Science*, Vol. 2018, No. 11, pp. 400-401, 2018.
- [13] G. Leng, M. Huang, Q. Tang, W. J. Sacks, H. Lei, and L. R. Leung, *Modeling the effects of irrigation on land surface fluxes and states over the conterminous United States: Sensitivity to input data and model parameters*, Journal of Geophysical Research Atmospheres, Vol. 118, pp. 9789-9803, 2013.
- [14] K. Rupnow, Y. Liang, Y. Li, D. Min, M. Do, and D. Chen, *High level synthesis of stereo matching: Productivity, performance, and software constraints*, 2011 International Conference on Field-Programmable Technology, pp. 12-14, 2011.
- [15] W. Zuo, W. Kemmerer, J. Bin Lim, L. Pouchet, A. Ayupov, T. Kim, K. Han, and D. Chen, *A polyhedral-based SystemC modeling and generation framework for effective low-power design space exploration*, 2015 IEEE/ACM International Conference on Computer-Aided Design, 2015.

유니버설미들웨어기반 소형위성 동적모듈 서비스 플랫폼

이해준

한국과학기술원 인공위성연구소 박사후연구원

요 약

최근 위성분야의 상업화로 융합서비스 발굴이 확산되고 있다. 우주서비스 시스템 설계 및 개발 방식이 사용자중심으로 전환되면서 우주산업환경인 우주인터넷망구축, 교통인프라 서비스, 지구관측 및 항법 정보 서비스 분야로 영향을 미치고 있다. 본 연구의 목적은 위성개발에서 요구되는 복잡한 시스템간의 통합에 있어 효과적인 융합 서비스플랫폼을 구성하는 것이다. 특히 소형 규모의 위성시스템을 대상으로 동적 모듈화 설계의 연구와 재활용 가능한 인터페이스 유닛을 설계하는 것이다. 이러한 요구 조건에 대해 복잡한 인터페이스의 분석 및 설계를 위한 위성기술 재활용을 목적의 유니버설미들웨어 기반 Public Protocol과 Private Interface 유닛으로 분리하여 구현하였다. 동적모듈 유닛은 하드웨어 자원의 공유와 재활용을 지원한다. 위성 내부와 외부 시스템 사이에서 데이터를 전송하는 표준모듈로 활용 할 수 있다. 또한, 위성서비스의 모듈간의 데이터 처리 과정에서 요구하는 파라미터를 명세서에 반영하여 실시간으로 적용하고 검증 할 수 있다. 결과적으로 Dynamic Service Platform은 다음과 같은 동적모듈의 규칙을 제공한다. 첫째, 유니버설미들웨어기반의 Application Platform상의 인터페이스 유닛은 Service Registry와 Service Inventory 간의 전송방식에 따른다. 이때, Instance 공유를 위해 Tracking과 Event Handling으로 구현한다. 둘째, 하드웨어모듈의 확장성을 지원하는 사용자 서비스중심 형태인 Private Message로 구성한다. 셋째, 외부 시스템과 내부시스템 연동방식은 Public Protocol, Application Launcher, Network Inventory를 명세서규칙을 반영하도록 각각의 유닛을 구성한다.



Hae Jun Lee received the bachelor's degree in the Department of Computer Science from the HanShin University in 1998. He received the M.S. degree and the Ph.D. degree in the Department of Computer Engineering from Korea PolyTechnic University in 2008 and 2014, respectively. He was a professor in the Department of Convergence Software Engineering at Kyungmin University from 2014 to 2018. He has been a researcher and Post-Doc. in the SaTRC at Korea Advanced Institute of Science and Technology since 2018. From 2004 to 2013, he was the founder venture company of ProSyst Korea Inc. which is venture company of Smart Home/Vehicle/Mobile. His current research interests include Universal Middleware, Satellite Platform, SAR Electric&Electroic Module, Signal Processing & System sets. He is a life member of the KKITS.

E-mail address: haejun@kaist.ac.kr

감사의 글

본 논문은 과학기술정보통신부의 차세대소형위성2호 개발사업의 일환으로 수행되었으며, 한국연구재단의 연구비 지원을 받아 작성되었습니다.