

## 3D 우주환경 가시화 도구 원격 제어 모듈 개발

한상혁\*, 구철희\*\*, 이훈희\*\*\*, 문성태\*\*\*\*

### Development of Remote Control Module on a 3D Universe Visualization

Sang-Hyuck Han\*, Cheol-Hea Koo\*\*, Hoon-Hee Lee\*\*\*, Sung-Tae Moon\*\*\*\*

#### Abstract

Recently, In Aerospace area the interest of open source software is increasing. One of examples is celestia that is used for visualizing space environment with 3D. Celestia that is open source software has many advantages. First is very easy to use, second is that it can extend new features easily with script language. It is very useful to extend with other systems. But, celestia has a few of remote control features from remote site. In this paper I describe design and implementation of remote control module using UDP communication protocol between celestia and GenSim that is satellite simulation software developed by KARI and describe the problem and solution items are found during development duration.

#### 초 록

최근, 항공우주 분야에서 오픈 소스 소프트웨어에 대한 관심이 늘고 있다. 대표적인 사례로 우주환경을 3D로 가시화하는 소프트웨어인 셀레스티아가 있다. 셀레스티아는 오픈 소스 소프트웨어로서 여러 장점이 있다. 첫째 사용하기 쉽고, 둘째 스크립트 언어를 지원하여 필요한 기능을 쉽게 추가할 수 있다. 이런 특징으로, 다른 시스템과의 연계도 용이하다. 그러나, 셀레스티아를 동일 시스템이 아닌 원격시스템에서 제어하는 기능은 부족하다. 본 논문은 위성 시뮬레이션 도구인 GenSim에서 UDP 통신을 이용하여 원격으로 명령을 전송하는 모듈의 설계 및 구현 과정에 대해 기술하고, 개발시 겪었던 문제와 이에 대한 해결방법을 기술한다.

키워드 : 셀레스티아(Celestia), CELX, 루아(Lua), GenSim, UDP, 원격제어(Remote Control)

접수일(2013년 5월 7일), 수정일(1차 : 2013년 6월 14일, 2차 : 2013년 6월 17일, 게재확정일 : 2013년 7월 1일)

\* 융합기술연구팀/shan@kari.re.kr    \*\* 융합기술연구팀/chkoo@kari.re.kr    \*\*\* 융합기술연구팀/lhh@kari.re.kr  
\*\*\*\* 융합기술연구팀/stmoon@kari.re.kr

## 1. 서 론

최근, IT분야 성장동력의 한 축인 오픈 소스 소프트웨어가 항공우주 분야를 포함한 다양한 산업 분야에 확산되고 있다. 오픈 소스 소프트웨어의 주요 장점은 소스 코드의 재활용, 커뮤니티의 활성화, COTS 수준의 품질 제공, 빠른 피드백이 있다. 따라서, 오픈 소스 소프트웨어의 활용은 항공우주를 포함한 전 산업 영역으로 확대 및 활성화가 예상된다. 물론, 오픈 소스 소프트웨어가 여러 장점을 가지고 있지만, 기술을 습득하기 위해 많은 노력이 필요하다.

항공우주 분야에서 가장 대표적인 오픈 소스 소프트웨어로 3D로 우주환경을 가시화하는 소프트웨어인 셀레스티아(Celestia)가 있다. 셀레스티아는 지구를 포함한 우주를 과학적으로 정확하게 묘사하고, 3D로 시각화하며, 루아 스크립트를 작성하여 필요한 기능을 추가할 수 있다[1]. 그러나, 셀레스티아에 기능을 추가하기 위해 루아 스크립트를 사용하는데[2], 셀레스티아를 원격에서 제어하는 기능은 부족하다.

본 논문은 3D 우주환경 가시화 도구인 셀레스티아와 KARI의 시뮬레이션 소프트웨어인 GenSim[3-4]간 UDP 기반 원격 제어 모듈의 설계 및 구현과정에 대해 기술한다. 또한, 셀레스티아를 활용하고자 하는 소프트웨어 개발자가 동일한 문제들을 겪지 않기 위해 개발과정에서 발생했던 문제점들과 해결방안을 기술한다.

본 논문의 구성은 다음과 같다. 2장에서는 3D 우주환경 가시화 소프트웨어인 셀레스티아와 인공위성 시뮬레이션 소프트웨어인 GenSim에 대해 기술하고, 3장에서는 UDP 기반 원격제어 모듈의 개발, 수행결과, 개발 시 발생했던 문제점과 해결방안에 대해 기술한다.

## 2. 셀레스티아와 GenSim

### 2.1 셀레스티아-3D 우주환경 가시화 소프트웨어

셀레스티아는 그림 1과 같이 3D로 우주환경을 가시화하는 실시간 우주 시뮬레이션 소프트웨어이다. 셀레스티아는 지구뿐 아니라 태양계 및 우주공간 전체에 대해 원하는 대상을 위치, 방향, 변화 속도를 임의로 조정 가능한 동적 능력을 제공한다[1].

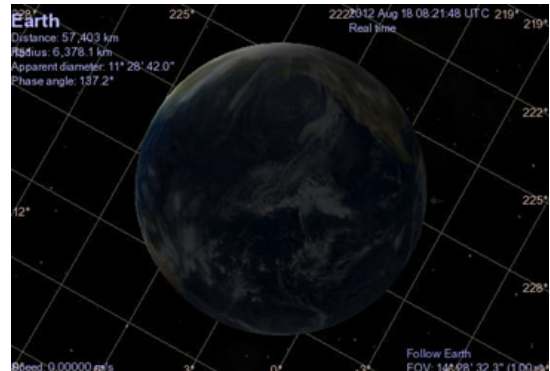


그림 1. 셀레스티아 실행 화면



그림 2. 셀레스티아 상에서 Lua\_edu\_tool 도구 실행화면

또한, 셀레스티아는 교육, 홍보 등 다양한 곳에 활용되고 있으며, 개발자 커뮤니티도 활성화되어 필요한 기능도 다양하게 추가되고 있다. 그림 2는 루아 스크립트로 작성된 GUI 기반 제어 사례를 보여준다. 셀레스티아의 주요 특징은 사용자가 필요한 기능을 추가할 수 있는 것인데, 응용 개발을 위해 CELX 스크립트와 루아 스크립트[2]를 사용한다.

## 2.2 GenSim-시뮬레이션 소프트웨어

GenSim은 Generic Simulator의 약어로 인공 위성과 OBC(On Board Computer)에 포함된 하드웨어 컴포넌트들에 대한 시뮬레이션을 수행할 수 있도록 제공하는 시뮬레이션 프레임워크이다. 하드웨어 컴포넌트란 하나의 기능을 갖는 하드웨어 모듈로 MIL-STD-1553B, RS-232, Ethernet 등이 여기에 포함된다. GenSim은 ESA SMI/SMP 표준에 맞게 개발되었기 때문에, SIMSAT와 EuroSim등과 같은 표준을 따르는 시뮬레이션 도구에 사용되는 모델을 적용 할 수 있다[3]. 그리고 비행 운영 소프트웨어 검증을 위해 VxWorks 및 리눅스와 같은 다양한 RTOS 디버깅 도구 (gdb, WDB)를 사용 할 수 있으므로 스레드/태스크 및 스택 정보 획득이 가능하다. 뿐만 아니라, GUI를 통해 모델의 상태 정보를 실시간으로 모니터링 할 수 있다. 현재 GenSim은 QT로 개발되어 윈도우 및 리눅스에서 사용 가능하다. GenSim은 시뮬레이션 커널의 기능을 제공하는 Infrastructure 부분과 시뮬레이터 모델을 가지고 있는 Simulator 부분으로 구성된다. Infrastructure 는 시뮬레이터 엔진 부분으로 스케줄링, 스크립트 해석, 로그 기록 등 전반적인 기능을 담당한다. 특히 SimNetwork를 통해 분산 처리를 수행하여 성능을 향상 시킬 수 있다[4]. Simulator 부분은 동적 링크 방식으로 개발되었고, 사용자가 개발하여 추가할 수 있도록 설계되었다.

## 3. 원격제어 모듈 개발

### 3.1 원격 제어 모듈 개념도

원격제어 모듈의 개념도는 그림 3과 같다. 원격 컴퓨터에서 제어를 위한 메시지를 UDP 방식으로 보내면, 셀레스티아에 루아, CELX 스크립트로 작성된 응용 프로그램에서 UDP 방식으로 메시지를 받아 필요한 기능을 수행한다.

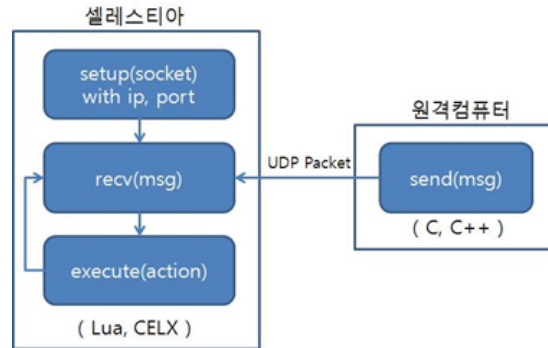


그림 3. 원격제어 모듈 개념도

원격 제어 모듈의 핵심인 UDP 모듈 송수신 모듈 제작시 주의할 사항이 있다. UDP는 connect 모드와 unconnected 모드가 있고, 모드에 따라 사용하는 함수와 소켓의 설정에 차이가 있다. connect 모드는 unconnected에 비해 불필요한 socket 해제/재연결을 수행하지 않음으로 성능상 장점을 가지나, 클라이언트에서 동일한 소켓으로 send/recv를 해야만 메시지 송수신이 되므로 셀레스티아에 다수의 원격 클라이언트를 연결이 안되는 단점을 가진다. 이에, 본 연구에서는 다수의 연결이 가능한 unconnected UDP를 사용하였다. 그림 4는 루아 스크립트로 UDP 소켓을 생성하고 대기하는 모듈의 소스 코드이다.

```
function createudp(celestia_host, celestia_port)
    local socket = require("socket");
    local udp_obj, err = socket.udp();

    if udp_obj == nil then
        celestia:print("Unable to create udp socket: " .. err,1);
    else
        celestia:print("Success to create socket");
    end

    --Bind
    local result, err=udp_obj:setsockname(celestia_host, celestia_port);
    if result == nil then
        celestia:print("Unable to set sock name: " .. err,1);
    else
        celestia:print("Success to set sock name");
    end

    udp_obj:settimeout(0);
    return udp_obj;
end
```

그림 4. 루아로 작성된 UDP 모듈

표 1은 루아로 작성된 UDP 프로그램이다. `createudp` 함수는 UDP 소켓을 만들고, `recvdata` 함수는 생성된 UDP 소켓을 통해 클라이언트로부터 데이터를 수신하는 함수이다.

또한, 셀레스티아는 GenSim으로부터 명령어, 대상객체, 1개 이상의 파라미터로 구성된 메시지를 해석하여, 해당 기능을 수행한다. 원격지로부터 수신되는 명령어 포맷과 사례는 표 1과 같다.

표 1. 원격지로부터 수신되는 명령어 포맷과 사례

항목	설명
명령어 포맷	<code>&lt;CmdType&gt; &lt;Object&gt; [&lt;Param &gt;+]</code> CmdType : 명령어 종류 (1 : ZoomIn/Out, 2: Rotate) Object : Month, Mars 등 객체 Param : 파라미터 * Delimeter 는 Blank 문자 사용
명령어 사례	1 Sol/MARS 25 10.0 5.0 MARS 객체를 5초동안 줌인하는 명령어
명령어 수행방법	1) GenSim . 스크립트 창에서 CelCtl(1, Sol/MARS, 25, 10.0, 5.0) 실행 2) nc 도구사용 . nc -u host port 입력후 1 Sol/MARS 25 10.0 5.0 메시지 입력

호출할 수 있는 기능은 세 가지로 객체에 대한 줌인, 줌아웃, 회전이다.

### 3.2 원격제어 모듈 개발환경

원격제어 모듈 개발환경은 표 2와 같다. S1은 주 개발환경으로 윈도우 XP 운영체제를 사용하고, S2는 원격시스템으로 리눅스 운영체제를 사용한다.

표 2. 원격 제어 모듈 개발 환경

위치	소프트웨어	버전	설명
S1	Celestia	1.6.1	3D 우주환경 가시화 소프트웨어
S1	Lua	5.1	셀레스티아 응용 개발 언어
S1	CELX	-	셀레스티아 응용 개발 언어
S2	nc		원격컴퓨터에서 셀레스티아로 UDP 메시지 전송 소프트웨어
S2	C/C++	-	원격지 메시지 송수신 개발 언어
S2	GenSim	1.1	

프로그램 개발 디렉토리의 구조는 표 3과 같다. 셀레스티아는 최초 호출되는 모듈이 CELX 스크립트로 되어 있고, 나머지는 루아 스크립트로 되어 있다. GenSim은 C++ UDP 통신을 위한 라이브러리를 제공한다.

표 3. 개발 디렉토리 구조

디렉토리명	파일명	설명
celestia	cel_zen_interface.celx	최초호출프로그램
celestia\lua	kari_udp.lua	udp 송수신 모듈
	socket.lua	루아 기본 패키지
	udp.lua	루아 기본 패키지
gensim	udp_send_recv_clib.cpp	c++ UDP 모듈
	CSpacecraft.cpp	자바스크립트 모듈

최초, 작업 디렉토리를 원하는 디렉토리에 설정하기 위해서 루아 스크립트의 `package.path`를 이용하여 패키지의 경로설정이 필요하다. 설정 사례는 표 4와 같다.

표 4. cel\_zen\_interface.celx 파일의 일부

```

...
package.path =
celestia:getscriptpath().."/../lua/?..lua;"
require "socket"
...
    
```

이를 통해 최초 스크립트가 실행되는 폴더 하위의 루아 폴더의 루아 스크립트들을 참조하여 socket이라는 패키지를 찾을 수 있다.

### 3.3 수행결과

원격 제어 모듈 검증을 위한 수행절차는 표 5와 같으며 각각의 절차에 대해서 세부적으로 살펴보면 다음과 같다.

표 5. 원격 제어 모듈 수행 절차

항목	셀레스티아	GenSim
P1	. UDP Bind 용 IP, PORT 스크립트 파일에서 설정	. 셀레스티아가 동작하는 IP, PORT 설정 및 컴파일
P2	. 셀레스티아 구동	-
P3	. 원격제어 모듈 실행 - 파일메뉴에서 cel_zen_interface.celx 파일 선택	-
P4	-	. GenSim 구동
P5	-	. 스크립트창에서 제어명령어 수행

#### ▶ P1. UDP IP, PORT 설정

원격 제어 명령어를 UDP를 통해 전송함으로 UDP BIND를 위한 IP, PORT 정보를 셀레스티아와 GenSim에 설정해야 한다. 셀레스티아의 관련 파일은 cel\_zen\_interface.celx 으로 그림 5와 같이 createudp 함수에서 IP, Port를 설정한다.

```

...
udp_obj = createudp("150.197.20.59",
"61120");
...
    
```

그림 5. cel\_zen\_interface.celx파일에서 UDP IP, PORT 설정부분

GenSim의 관련 파일은 CSpacecraft.cpp 으로 그림 6과 같이 헤더에서 셀레스티아의 IP 및 포트를 설정한다. 설정이 바뀌면 그림 7과 같이 빌드를 수행하여 라이브러리를 갱신한다.

```

...
#define CEL_IP "150.197.20.59" /* PC of S. HAN */
#define GEN_IP "150.197.20.72"
#define CEL_PORT 61120
#define GEN_PORT 61130
#define BUFLen 512
...
    
```

그림 6. CSpacecraft.cpp 파일에서 UDP IP, PORT 설정부분

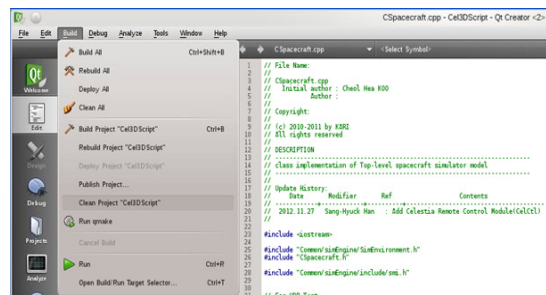


그림 7. QT 빌드하기

빌드가 성공적이면 Cel3DScript 관련 라이브러리가 libCel3DScript.so 이름으로 생성되며, GenSim은 그림 8 같이 사용자 라이브러리의 위치를 설정하여 실행시 동적으로 로딩한다.

```

<?xml version="1.0" encoding="UTF-8"?>
<GenSim type="GenSim Configuration File">
  <General>
    <UserName>Hoon Hee Lee</UserName>
    <ModuleName>Paper</ModuleName>
    <EnableOS>RTEMS</EnableOS>

<LoadImageName>PaperDemo.src</LoadImageName>
    <RamSize>4096</RamSize>
    <RamStart>0x40000000</RamStart>
    <RomSize>2048</RomSize>
    <RomStart>0x0</RomStart>
    <ScriptFile1>JSTest.txt</ScriptFile1>
    <ScriptFile2>dummy.txt</ScriptFile2>
    <ScriptFile3></ScriptFile3>
    <ScriptFile4></ScriptFile4>

<LoadUserLibrary>/home/GenSimFactory/Cel3DScript-build-desktop/libCel3DScript.so</LoadUserLibrary>
  </General>
  
```

그림 8 libCel3DScript.so 패스 환경설정

▶ P2. 셀레스티아 구동

셀레스티아 실행파일을 더블클릭하면 그림 9 같이 셀레스티아 실행화면을 볼 수 있다.

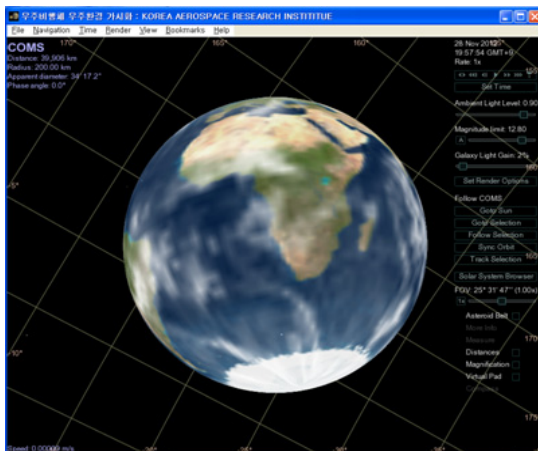


그림 9 셀레스티아 실행화면

▶ P3. 원격제어 모듈 실행 - 셀레스티아  
셀레스티아 실행 후 그림 10과 같이 원격제어 모듈 스크립트인 cel\_zen\_interface.celx 파일을

선택한다. 이 모듈은 첫째로 원격 제어에서 호출될 기능을 차례로 동작하는지 검증하고, 원격지로부터 UDP 통신을 통해 명령어를 수신할 준비를 한다. 준비되면 "data is nil"이라는 메시지가 출력된다.

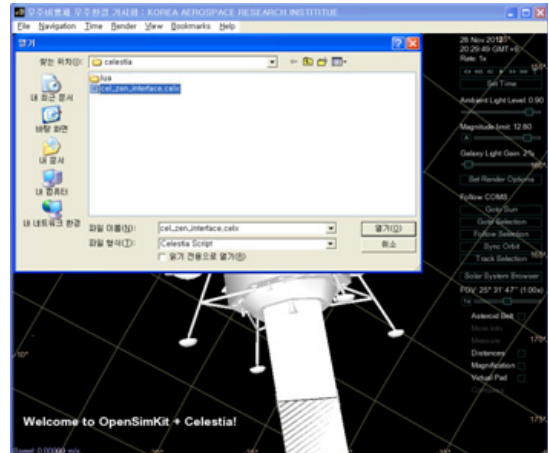


그림 10. 원격제어 모듈 스크립트 선택

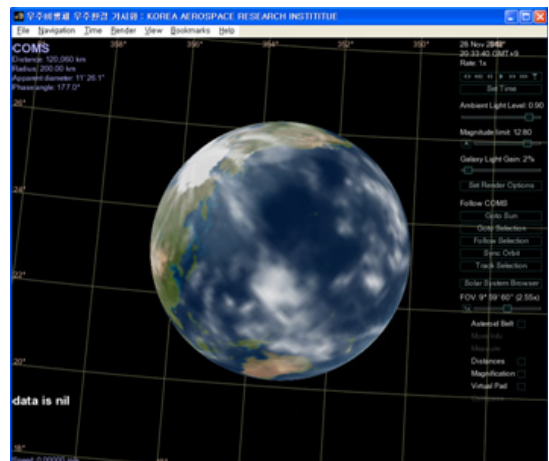


그림 11. 원격제어 모듈 준비 완료 상태

▶ P4. GenSim 구동

콘솔창에서 ./GenSim 명령어를 수행하면 그림 12와 같이 GenSim이 실행된다.

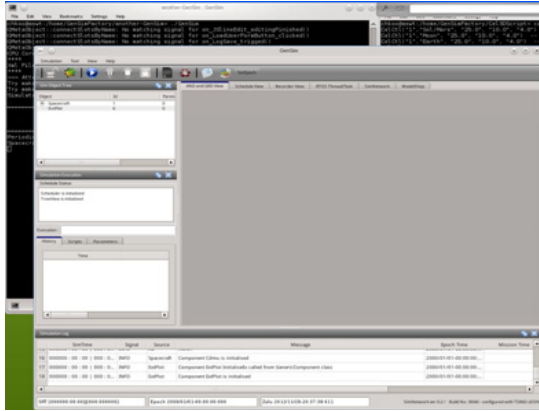


그림 12. GenSim 실행 화면

▶ P5. GenSim 스크립트 창에서 명령어 수행

그림 13은 GenSim에서 수행하기 위한 줌인/줌아웃/로테이트 명령어 사례이다.

```

CelCtl("1", "Sol/Mars", "25.0", "10.0", "4.0") -- ZoomIn
CelCtl("1", "Moon", "25.0", "10.0", "4.0") -- ZoomIn
CelCtl("1", "Earth", "25.0", "10.0", "4.0") -- ZoomIn
CelCtl("1", "COMS", "25.0", "10.0", "4.0") -- ZoomIn
CelCtl("2", "Sol/Mars", "50.0", "4.0") -- Rotate
CelCtl("2", "Moon", "50.0", "4.0") -- Rotate
CelCtl("2", "Earth", "50.0", "4.0") -- Rotate
CelCtl("2", "COMS", "50.0", "4.0") -- Rotate
    
```

그림 13. 원격 제어 명령어 사례

이중 CelCtl("1", "Earth", "25.0", "10.0", "4.0")을 GenSim 스크립트 창에서 수행하면, 셀레스티아는 그림 14와 같이 동작한다.

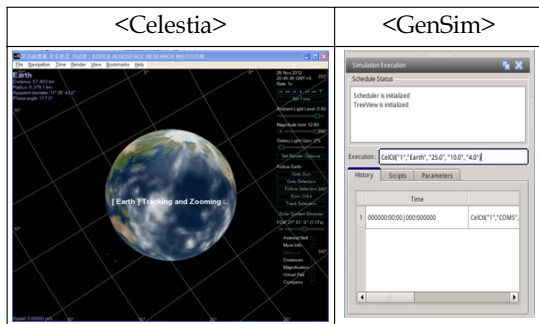


그림 14. 지구(Earth) Zoom In 사례

### 3.4 개발과정에서의 문제점 및 해결방법

표 6은 원격 제어 모듈을 개발하는 과정에서 겪었던 대표적인 문제에 대한 증상/원인/해결방안을 정리한 것이다.

표 6. 개발과정상에서의 문제점 및 해결방법

번호	문제	증상/원인/해결방안
1	셀레스티아 멈춤 현상(1)	(증상) 셀레스티아의 화면이 멈추고 다른 동작으로 넘어가지 않음 (원인) udp:settimeout( seconds )의 seconds 가 0이 아닐 때 발생 (해결) seconds값을 0으로 설정
2	셀레스티아 멈춤 현상(2)	(증상) 제어권이 셀레스티아에 넘어오지 않음 (원인) 루아 while 문 안에 wait() 함수를 사용하지 않아, 제어권이 넘어오지 않았음 (해결) wait(1) 문장을 추가함
3	루아 socket 패키지 찾지 못함	(증상) CELX에서 socket 패키지 및 관련 함수 호출시 socket 패키지 찾을 수 없다는 메시지 출력 (원인) 라이브러리가 패스에 추가되지 않아서 발생함 (해결) package.path에 해당 디렉토리를 추가하여 해결
4	루아 UDP 송수신 오류	(증상) send는 수행되나 recv 함수는 오류가 발생함 (원인) UDP의 설정이 connect로 되어 있고, 원격시스템에서 send, recv가 서로 다른 socket을 사용할 때 발생 (해결) unconnect UDP를 사용하거나, send/recv에서 사용하는 socket을 동일하게 설정

## 4. 결 론

지금까지 GenSim에서 UDP 통신을 이용하여 셀레스티아를 원격으로 제어하는 모듈의 설계 및 개발에 대해 살펴보고, 개발 과정에서 겪었던 여러 문제들의 원인 및 해결과정을 기술하였다. 향후, 원격 제어 모듈에 다양한 명령어를 추가하여 3D 가시화 도구와 연계한 위성의 고장 확인을 쉽게 할 예정이다.

## 참 고 문 헌

1. Celestia User's Guide, "<http://www.shatters.net/celestia/documentation.html>"
2. Lua 5.0 Reference Manual
3. C. H. Koo, H. H. Lee, and Y. J. Cheon, 2010, "SMI Compatible Simulation Scheduler Design for Reuse of Model Complying with SMP Standard", JASS, Vol. 27, No. 4, pp. 407~412.
4. C. H. Koo, H. H. Lee, and Y. J. Cheon, 2011, "Distributed simulation design by using of SimNetwork to overcome speed limit on GenSim", IEEE ICRAST, Vol. XX, No. X, pp. 430~435.