

## Flexible Software Design for Korean WA-DGNSS Reference Station

W.S. Choi, S.S. Chhattan, J.E. Kye & W.Y. Han

*ICT Convergence Autonomous Research Team, Electronics and Telecommunications Research Institute, Korea*

H. Yun & C. Kee

*Mechanical and Aerospace Engineering and Institute of Advanced Aerospace Technology, Seoul National University, Korea*

**ABSTRACT:** In this paper, we describe the software design results of WA-DGNSS reference station that will be constructed in Korea in the near future. Software design of the WRS (Wide area Reference Station) is carried out by applying object oriented software methodology in order to provide flexibilities: easy of model change (namely ionospheric delay model etc) and system addition (Galileo, GLONASS in addition to GPS etc). Software design results include the use case diagrams for the functions to be executed, the architecture diagram showing components and their relationships, the activity diagrams of behaviors and models among them, and class diagrams describing the attribute and operation.

### 1 INTRODUCTION

In Korea, technologies on WA-DGNSS (Wide area differential GNSS) are being jointly developed by a consortium of universities and research organization [1, 2, 3, 4, 5, 6] for the construction of system in the near future.

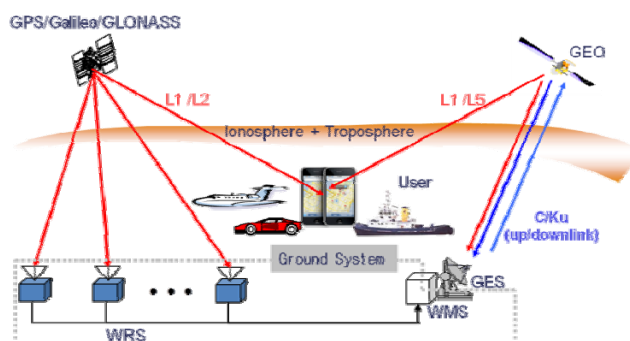


Figure 1. Korean WA-DGNSS Architecture

WA-DGNSS system comprises WRS (Wide area Reference Station), WMS (Wide area Master Station)/GES, GEO satellite and User segment (Fig. 1). WRS performs the quality monitoring of navigation message decoded from the received GNSS (GPS, Galileo etc) satellite, and transmits the monitored and processed data to the WMS via terrestrial communication network.

WMS generates the correction messages by processing the data received from WRSs, and uplinks them to the GEO satellite via GES (Ground Earth Station). GEO satellite transmits the correction message and its own navigation message to the user with the same frequency as the GNSS satellite.

WA-DGNSS compensates for the weakness of GNSS in terms of accuracy, integrity, continuity and availability. SBAS (Satellite Based Augmentation System) broadcasts GPS correction message and its integrity in real time via GEO satellite in wide area. WAAS, EGNOS and MSAS provide services in the U.S., EU and Japan respectively. Also GAGAN,

SDCM etc are being planned operational in India, Russia etc.

This paper describes the detailed design results of the WRS software. Use cases of the WRS are defined and designed in order to fulfill the WRS functions. Architecture design, activity diagram design and class diagram design are followed in order to complete the WRS software design by applying the object oriented software methodology.

## 2 WRS FUNCTIONS

WRS (Wide area Reference Station) collects and processes the GPS navigation data, and calculates errors such as ionospheric delay, tropospheric delay, pseudorange residuals etc. And WRS sends the calculated errors to WMS(Wide area Master Station) to generate the wide area differential data.

The following is a summary of WRS Functions:

- Collect Raw GPS Data
- Determine Satellite Orbits
- Determine Satellite Corrections
- Determine Satellite Integrity
- Calculate Ionospheric Delay
- Calculate Tropospheric Delay
- Calculate Pseudorange
- Estimate Pseudorange Residuals
- Monitor Data Quality
- Determine WRS Integrity
- Perform Data Verification
- Transmit Raw and Pre-processed Data to WMS
- Log data

In order to implement the WRS functions, we take a software approach that provides flexibilities in software architecture of easy model changes and easy system addition or deletion, namely adding GLONASS in addition to GPS. This can be regarded as an advantage of the WRS software design.

## 3 DIRECTIONS IN WRS SOFTWARE DESIGN

WRS Design is carried out by applying a UML based software methodology, and the design results include Use case diagram, Architecture diagram, Activity diagram and Class diagram. It is also designed to provide TCP/IP communication with WMS to send the generated data in WRS.

In order to provide the flexibility of the software, the WRS software is designed in modular concept for easy maintenance: minimizing the effect to other components due to a component changes and system addition.

Through preliminary design of WRS software [6], UML (Unified Modeling Language) based use case diagram is designed, and software architecture design is followed. Input, Processing and output are also defined for each component.

In detailed design, activity diagram required for carrying out WRS functions is designed, and class

diagrams are designed as the final step for WRS software design.

## 4 WRS SOFTWARE DESIGN

### 4.1 Use Case Diagram

Use case diagram comprises with Actors and Use Cases. Actors interacting within the WRS software consist of four components such as Time, WRS Operator, WRS Administrator and WMS. Time triggers an event when time interval expires or new data arrives within the system. WRS Operator performs various tasks related to raw and pre-processed GNSS and WRS data. WRS Administrator is responsible for configuration and maintenance of the WRS software. WMS actor requests the raw or pre-processed data or WRS data from WRS software.

Total 18 use cases are defined in order to fulfill the WRS functions as UC1 Collect Data, UC2 Determine Satellite Orbits etc.

One of the WRS use cases, the UC8-Calculate Pseudorange is shown in Fig.2. This use case calculates the pseudorange, and the actors are Time, WRS Operator and WMS.

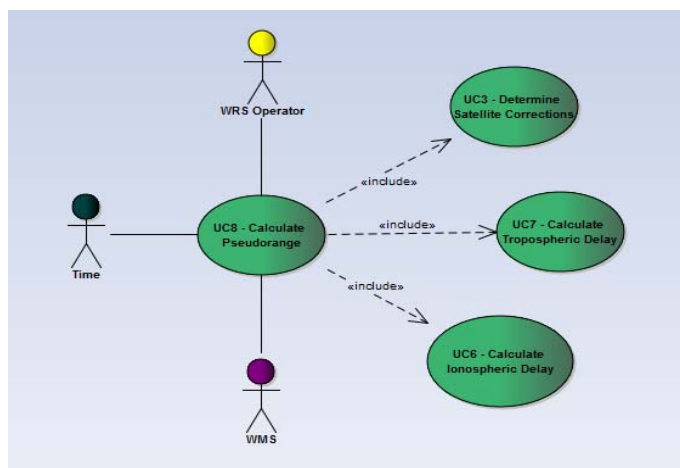


Figure 2. Use Case-Calculate Pseudorange (UC8)

### 4.2 WRS Software Architecture

Based on the Use cases designed, WRS software architecture (Fig. 3) is configured to perform the defined use cases, and some components have sub-components respectively. Input, Processing and Output for each sub-component are defined in order to carry out the use cases designed. In component level, subcomponent can be added or deleted in order to reflect the changes in model or system. The WRS software is configured so that other GNSS such as Galileo can be easily added to the current system architecture. This property provides the flexibility in WRS software system, and it can be regarded as advantages for this software design approach.

The WRS software architecture comprises 19 components such as Navigation Message Receiver, Navigation Message Decoder, Data Handler, Data Quality Monitor, SV Integrity Monitor, WRS Integrity

Monitor, WRS-WMS Network Monitor, Data Processor, Meteorological Station, SV Azimuth and Elevation Calculator, Pseudorange Calculator, WRS Position Calculator, Independent Data Verifier, Raw

and Pre-processed Data Buffer, WRS Information Manager, WRS Operation and Maintenance, Data Logger, Data Display Unit, WRS-WMS Interface Unit.

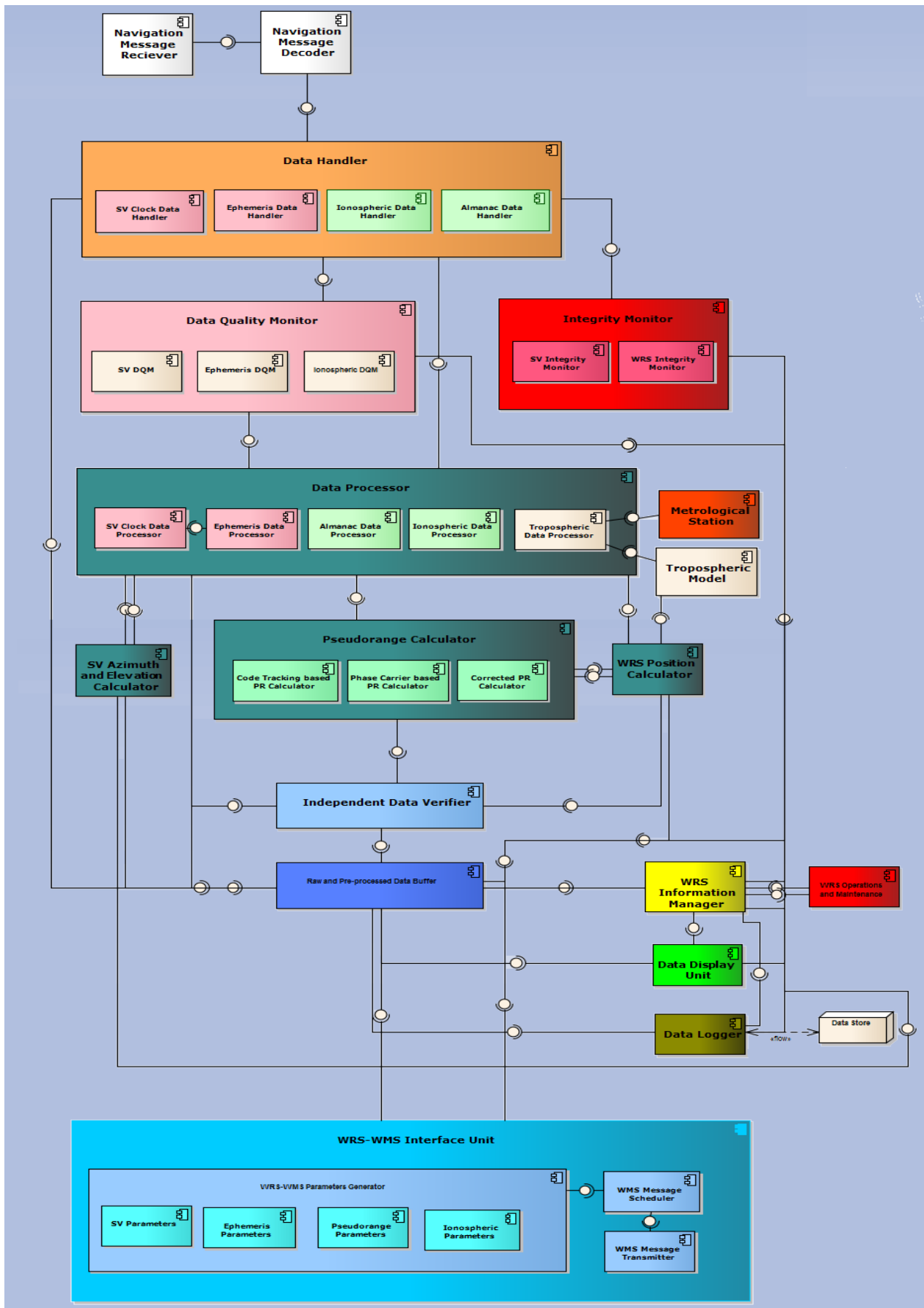


Figure 3. WRS Software Architecture

## 5 WRS SOFTWARE ACTIVITY DIAGRAM

Activity diagram is designed to represent the workflows of stepwise activities and actions of WRS software.

Elements of activity diagram include Start Node, Finish Node, Activity, Action, Control Flow, Decision Node and Fork/Join Nodes.

Major processing actions include calculations of ionospheric delay, tropospheric delay, SV clock correction, pseudorange, pseudorange residuals, data transmission to WMS etc.

### 5.1 WRS Software Class Diagram

Class diagrams are designed to describe the static structure of the WRS software, and each class in the diagram has attribute, operation and relationship with other classes.

Class Diagrams of the WRS software comprise with four categories: Raw and Pre-processed Data Holder Classes, Data Processing Classes, Data Logging and Transmission Task Classes, Parsing RINEX Files Classes.

The following is an example of details for one of the classes, i.e. AlmanacDataProcessor Class (Fig. 4) designed as relationships and operations.

#### Relationships:

Relationship Type	Source	Target
<b>Association</b>		
Source ->	Public	Public
Destination	AlmanacDataProcessor	AlmanacData
<b>Association</b>		
Source ->	Public	Public
Destination	AlmanacDataProcessor	SatellitePosition

#### Operations:

Method	Parameters	Description
<b>Almanac-DataProcessor()</b>	<b>AlmanacData</b> [in]	Constructs AlmanacDataProcessor object and initializes the member variables with provided almanac data.
Void	almanacData	
Public		
correctAlmanacReferenceTime()	<b>double</b> [in] gpsTransmissionTime double	Corrects almanac reference time.
Public	<b>double</b> [in] satelliteClockError <b>AlmanacData</b> [in] almanacData	

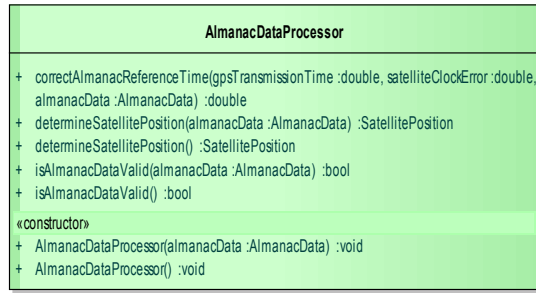


Figure 4. AlmanacDataProcessor Class

## 6 CONCLUSIONS

In this paper, we described the software design results of the Korean WRS. Software design results include use case diagram, architecture diagram, activity diagram and class diagram. The WRS software design also takes a flexible architecture that facilitates software maintenance, namely ease of model change and system addition such as addition of Galileo by applying the object oriented software methodology. This flexible software design approach can be regarded as an advantage of the WRS software design.

Implementation of the WRS software will be followed by validating each component, and it will be integrated with the WMS for AIT (Assembly and Integration Testing). System level comprehensive validation will be performed for completeness of WRS software development.

## ACKNOWLEDGEMENT

This research was a part of the project titled "WA-DGNSS Development" funded by the Ministry of Land, Transport and Maritime Affairs, Korea.

## REFERENCES

- [1] Ho Yun, Changdon Kee, Doyoon Kim, "Availability Performance Analysis of Korean Wide Area Differential GNSS Test Bed", The Korea Navigation Institute, Vol 15, No 4, Aug. 2011
- [2] Ho Yun, Changdon Kee, and Doyoon Kim. "Korean Wide Area Differential Global Positioning System Development Status and Preliminary Test Results", International Journal of Aeronautical and Space Science, Vol. 12, No. 3, 2011, pp. 225-233
- [3] Uzair Ahmad, Choi Wan Sik, Changdon Kee, "Processing of WA-DGNSS Correction Messages: A Functional Perspective", The 18<sup>th</sup> GNSS Workshop, Phoenix Island, Jeju, Nov. 2011
- [4] Yong-Won Ahn, Donghyun Kim, Jason Bond, Wan Sik Choi, "Performance Test of the WAAS Tropospheric Delay Model for the Korean WA-DGNSS", Journal of the Korea Navigation Institute, Vol 15, No 4, Aug. 2011
- [5] Changdon Kee, Ho Yun, Wan Sik Choi, "Transmission Data Specification for GNSS(GPS) Wide area Reference Station", TTA.KO-06.0266, Dec. 2011
- [6] Wan Sik Choi, Shah Sayed Chhattan, Joong Eup Kye, Ho Yun, Changdon Kee, "Data Processing Design of Korean WA-DGNSS Reference Station", enc2012, 25-27 April, Gdansk, Poland