２０１６年度

博士学位論文

API Integration Platform for Agronomic Models

農業関連モデルのための API 統合プラットフォーム

中部大学大学院

工学研究科　建設工学専攻

Rassarin Chinnachodteeranun

Abstract of the Dissertation

This research developed a novel API integration platform which provides essential information, both observation and predictive analytics, for making decisions in agriculture through APIs. The platform harmonized several data sources on climate and soil for operating agronomic models namely crop models and stochastic weather generators. Climatological data API is developed in compliance with Sensor Observation Service (SOS) API, an international standard interface, to secure interoperability between services. Soil API was built in order to provide soil profile, an essential information for determining water condition for crop growth. Users could acquire soil profile in a standard format compatible with crop models. The complex internal structure and diverse protocol of the data sources were hidden behind the APIs. These APIs were seamlessly wired in the API integration platform. Crop model API was developed by embedding a standalone crop model into the platform. It was virtually constructed as the top layer of the platform so that it can internally call other services for inputs. Crop model API provides a service on simulating crop growth processes and yield, and thus considered as a vital component of agricultural decision support systems. Weather generator APIs were developed in this platform in order to provide services on generating synthetic weather realizations of coming cropping season, indispensable inputs for crop models. Collection of APIs for data preparation and operating agronomic models were harmoniously constructed in this platform. Users could acquire data through APIs with no need to learn complicated internal structure of the models. Useful agricultural applications based on modeling can be easily developed by utilizing the services provided by the platform. This API integration platform, which revolutionizes the way to operate agronomic models, is a noticeable foundation for expanding applications and services in the agronomic arena.

Dedication

This research is one of my journeys which I spent years to reach the destination. This magnificent journey provides me with considerable opportunities to fly all over the world, experience several turbulences, meet great researchers across the globe and see the world in broader vision. All mentioned circumstances contributed to this research.

I would like to dedicate this research to people who have supported me throughout the journey.

"Did you ever know that you're my hero?

You're everything I wish I could be.

I could fly higher than an eagle, 'cause you are the wind beneath my wings."

The Wind Beneath My Wings by Jeff Silbar and Larry Henley, 1982

Rassarin Chinnachodteeranun

Acknowledgement

My doctoral research has made a profound impact on me for both academic skill and perspective of life. I would like extend my wholehearted appreciation to those extraordinary people who had generously contributed to my work presented in this dissertation.

Foremost, I would like to express my sincere gratitude to my advisors, Prof. Naoki Matsuo and Prof. Kiyoshi Honda, who have supported me since the starting of this research. With their kind supports, my Ph.D. hood has been an amazing experience and full with great opportunities. Their vast knowledge and thoughtful advice contributed to every part of this research.

Besides my advisors, I would like to thank other advisory members, Prof. Seishi Ninomiya, Dr. Amor V.M. Ines, Prof. Masaru Mizoguchi, Prof. Makoto Takeda and Prof. Takaharu Kameoka, for their insightful suggestions and encouragements.

My sincere thanks also go to my colleagues: Dr. Nguyen Duy Hung, Mr. Junichi Okamura, Mr. Koumei Kuroda, Mr. Iwamasa Nishikado, Mr. Hiroyuki Sakaue, Dr. Eunjin Han and Dr. Apichon Witayangkurn for their supports and collaboration through various exciting projects. I would like to extend my gratitude to FI-Space team in Wageningen University, the Netherlands, for informative meetings at the early stage of my research as well as to DSSAT community for providing me with full support on DSSAT crop modeling.

I highly appreciate the Cabinet Office, Government of Japan, for funding the Cross Ministerial Strategic Innovation Promotion Program (SIP), of which my research is a part.

I am also thankful to my role models, Dr. Thanksin Shinnawatra, former Prime Minister of Thailand and Mr. Phongthep Thepkanjana, former Deputy Prime Minister of Thailand, who have inspired me to achieve my goal.

Lastly, I would like to thank my family: my parents, sister and brother for standing beside me throughout my life, consoling me when I fall, and encouraging me to stand up and work harder.

Curriculum Vitae

Name: Rassarin Chinnachodteeranun

E-mail: ch.rassarin@gmail.com

Research Interest:

- Field sensor network platform and applications
- Crop simulation application
- Web service for agro-system and data interoperability

Academic Record

- Master of Science (Information Management), Computer Science and Information Management Field, Asian Institute of Technology, Pathumthani, Thailand in August 2007
- Bachelor of Computer Science, Computer Science Department, King Mongkut Institute of Technology Ladkabang, Bangkok, Thailand in May 2004
- Science-mathematics major in high school, Nakhon Nayok Wittayakom school, Thailand in February 1999

Award

- Finalist, NIA award (National Innovation Agency) Thailand, 2012
- Silver award of True Innovation Award 2011, AgroSense War Room Project
- The winner of Thailand ICT Award 2010 in Application and Infrastructure Tools Category on Sensor Service GRID
- The winner of Asia Pacific ICT Alliance Award (APICT) 2010 in Tools and Infrastructure Applications on Sensor Service GRID, Malaysia
- Finalist, Thailand Prime Minister's Creative Award, Functionality Category, 2010
- Best Paper Award to 30ACRS2009 on the topic of "UAV-BASED SENOSR WEB MONITORING SYSTEM"

Work Experience

- Visiting Ph.D. scholar, Michigan State University, USA, July 2016 – August 2016
- Advisor to the permanent secretary, Ministry of ICT, Thailand, November 2013 - June 2014
- Coordinator, Nakhon Nayok Smart Province, the road map to smart Thailand project, May 2012– May 2013
- Research Specialist, Asian Institute of Technology, Pathumtani, Thailand, October 2007– July 2012

Publication and Conference

- Designing and Implementing Weather Generator as Web Service: Rassarin Chinnachodteeranun, Nguyen Duy Hung, Kiyoshi Honda, Amor V.M. Ines, Eunjin Han: Future Internet Journal, 8(4): 55, DOI: 10.3390/fi8040055, 15 December 2016
- Sensor Observation Service API for Providing Gridded Climate Data to Agricultural Applications: Rassarin Chinnachodteeranun, Kiyoshi Honda: Future Internet Journal, 8(3): 40, DOI: 10.3390/fi8030040, 8 August 2016
- Designing Crop Simulation Web Service with Service Oriented Architecture Principle: Rassarin Chinnachodteeranun, Nguyen Dug Hung, Kiyoshi Honda: AGU Fall Meeting 2015 (https://agu.confex.com/agu/fm15/meetingapp.cgi/Paper/69005), San Francisco, USA, DOI: 10.13140/RG.2.1.3520.8568, 18 December 2015
- Designing dynamic binding framework for weather generator web service: Rassarin Chinnachodteeranun, Nguyen Dug Hung, Kiyoshi Honda: Ambient intelligence and Sensor Network (ASN) 2015, Tokyo, Japan, November 2015
- Ecosystem of Web Services for crop simulation: prototype and further development: Rassarin Chinnachodteeranun, Nguyen Dug Hung, Kiyoshi Honda: 40th Asia Pacific Advanced Network Meeting, Kuala Lumper, Malaysia, August 2015
- Web-based wheat simulation by DSSAT on sensor observation service standard API: Rassarin Chinnachodteeranun, Kiyoshi Honda, Amor V.M. Ines, Kumpee Teeravech, Apichon Witayangkurn: IEEE Society of Instrument and Control Engineers of Japan (SICE), DOI: 10.1109/SICE.2015.7285540, July 2015
- Agriculture Information Service Built on Geospatial Data Infrastructure and Crop Modeling: Kiyoshi Honda, Amor V. M. Ines, Akihiro Yui, Apichon Witayangkurn, Rassarin Chinnachodteeranun, Kumpee Teeravech: Proceedings of the 2014 International Workshop on Web Intelligence and Smart Sensing, IWWISS '14, Saint Etienne, France, ACM 978-1-4503-2747-3/14/09, doi:10.1145/2637064.2637094, 1-2 September 2014
- Agricultural decision support system by multi-scale sensing and modeling: Kiyoshi Honda, Akihiro YUI, Amor M.V. Ines, Rassarin Chinnachodteeranun, Apichon Witayangkurn: WG5: AGRICULTURE AND FOOD SECURITY (GEO GLAM), the 7th GEOSS Asia-Pacific Symposium, Tokyo, Japan, 26-28 May 2014
- Fieldtouch - Agricultural Information Platform on OGC Web Services: Kiyoshi Honda, Akihiro YUI, Amor M.V. Ines, Rassarin Chinnachodteeranun, Apichon Witayangkurn: OGC TC/PC Meeting - IIT Bombay, India, 2-5 December 2013

- Production Management for Aqua Farm Based on Field Sensor Network Technology and GAP; Rassarin Chinnachodteeranun, Kiyoshi Honda, Apichon Witayangkurn: SICE Annual Conference 2013, Nagoya Japan, September 2013
- Publishing Agriculture Information on SSG: Rassarin Chinnachodteeranun, Honda Kiyoshi, Apichon Witayangkurn: Asian Federation for Information Technology in Agriculture (AFITA), Bogor, Indonesia, 4-7 October 2010
- Sensor service grid as real-time monitoring infrastructure and its application to soil moisture observation in Thailand: Kiyoshi HONDA, Aadit SHRESTHA, Apichon WITAYANGKURN, Rassarin Chinnachodteeranun and Masaru MIZOGUCHI: Journal of the Japanese Society of Soil Physics, pp.25-28, No. 115, 11 July 2010
- UAV-based sensor web monitoring system: Masahiko Nagai, Apichon Witayangkurn, Aadit Shrestha, Rassarin Chinnachodteeranun, Kiyoshi Honda and Ryosuke Shibasaki: 30 ACRS2009, Beijing, China, 18-23 October 2009
- Field Server Data Acquisition using Instant Messaging: Rassarin Chinnachodteeranun. University of Hawaii, Hawaii USA APAN, 24 January 2008
- Sensor Asia Initiative and its application to Landslide Early Warning System: HONDA Kiyoshi, Aadit Shrestha, Rassarin Chinnachodteeranun, Apichon Witayangkurn, Nguyen Duy Hung, Hiroshi Shimamura: Forum on Disaster risk management in mountainous areas – How to reduce vulnerability to flash floods and other related phenomena, Hanoi, Vietnam, 15 December 2008
- Field Sensor Network for Environmental Monitoring - Sensor Asia Initiative and Applications –: Kiyoshi Honda, Rassarin Chinnachodteeranun, Aadit Shrestha, Apichon Witayangkurn, Hung Nguyen Duy, Hiroshi Shimamura: Proceedings of GeoInformatics for Spatial-Infrastructure Development in Earth & Allied Sciences (GIS-IDEAS), Hanoi, Vietnam, 4-6 December 2008
- Spinach Field Monitoring for bridging Thai Producer and Japanese Consumer under Sensor Asia: HONDA Kiyoshi, Aadit Shrestha, Rassarin Chinnachodteeranun, Masaru Mizoguchi, Hiroshi Shimamura, Takaharu Kameoka: SICE Annual Conference 2008. The University Electro-Communications, Japan, pp.2582-2585, 20-22 August 2008
- Landslide Early Warning System for Rural Community as an Application of Sensor Asia: Honda Kiyoshi, Aadit Shrestha, Rassarin Chinnachodteeranun, Nguyen Duy Hung, Hiroshi Shimamura, Teuku Faisal Fathani, Dwikorita Karnawati, Kyoji Sassa: World Conference on Agricultural Information and IT / IAALD AFITA WCCA 2008, Tokyo Japan, 24-27 August 2008

Table of Contents

List of Figure

List of Table

CHAPTER 1: Introduction

Climate variability directly affects the crop growth and production across the globe. Uncertainties in changing climate, resources and economy cause growing needs for data to support decisions by agricultural communities, governments, public sectors and academia. Traditional decisions took place in the farms based on trial and error together with observation and knowledge passed down through generations. In agriculture 4.0, the era of data driven decision making and interoperability, observed data and predictive analytics are playing an important role for better decisions in order to increase productivity, minimize risks and optimize resource usages. Several forms of modern data collection devices such as sensors, Internet of Thing (IoT) and Unmanned aerial vehicle (UAV) are all around us for monitoring and collecting data on environment. We should take advantage of technology and data capabilities to achieve breakthroughs in better decision makings for agriculture. Data are essential for calibrating agronomic models for obtaining accurate results. Weather generator and crop simulation are the key technologies that have been using for predicting future condition based on input. Weather generator is a tool for predicting weather in coming season while crop simulation is one of the fundamental technologies for predicting crop growth, crop development and yield according to weather condition, crop management, soil condition and genetics. From the above reasons, demand on utilizing agronomic models as main engine for decision support systems is accelerating. What-if questions for managements and decision makings can be asked to agronomic models instead of doing trial and error on real situations. Data availability and agronomic models can bring big benefits for precision and sustainable agriculture.

1.1 Problem Statement

Agronomic models were developed by scientists for research purposes, mostly as standalone applications or programs. Operating agronomic models are complicated with hundreds of internal components. Various data from different sources are required as input to agronomic models. Preparation of the data is an iterative and extremely time-consuming process. Moreover, multidisciplinary knowledge is necessary to fully understand and appropriately operate crop simulation and weather generator. Users need to understand the internal mechanism of the models. Various input data and enormous effort are required for calibrating and operating the models. These cause developments of decision support systems complex and only big team of developers can afford to do.

The problems of current situation can be classified as follows:

1. Lack of interoperability between climatological data from different sources. Each climatological data source has proprietary protocol and data format. It means that users need to learn a protocol and data format of each source one by one.

2. Lack of harmonization between climatological data and weather generator. Weather generators were developed in closed environment for operating as standalone applications. Data preparation for climatological data as required input of weather generators needs to be done manually.
3. Intricacy of soil data. Soil data are kept in various formats according to survey teams' preference. To prepare soil data for running agronomic models, soil data need to be formatted into complex hierarchical structure.
4. Complexity of operating agronomic models. Programing languages used to develop agronomic models are suitable for scientific purposes but they do not harmonize with trend of applications or systems development.

1.2 Objective

The objective of this research is to develop an agronomic API integration platform that has a collection of web APIs for providing essential data in both observed and predictive analysis for facilitating development of agricultural decision support system. The platform should simplify procedures to operate agronomic models, as well as eliminate indirect logic tasks such as format conversion of data. In order to achieve the objective, web API frameworks which integrates DSSAT crop simulation model, stochastic weather generators and data preparation are implemented. With reference to the problem statement, the main tasks of this research to answer those classified issues are as follows:

1. Developing a climatological data API that enables interoperability of multi data sources
2. Developing weather generator APIs as parts of the platform for exposing standalone weather generators online and seamlessly connecting to the climatological data API
3. Developing a soil API to provide soil profile in a standard format suitable for DSSAT crop simulation model
4. Developing a crop model API for online accessible DSSAT crop simulation model

Each API shall work independent. Meanwhile, Output from an API shall seamlessly flow to be an input of another API. All APIs shall be connected for facilitating crop models' operation.

1.3 Dissertation organization

Chapter 2 describes background knowledge required for this research as well as a problem analysis. Chapter 3 provides development lesson from previous research which is the inspiration to conduct this research. Chapter 4 provides the overview of this research including designing the API integration platform. Chapters 5 to 8 describes the design and implementation details of each API in the platform: Chapter 5 climatological API, Chapter 6 weather generator API, Chapter 7 Soil API, and Chapter 8 crop model API, respectively. Chapter 9 is the conclusion of the research.

CHAPTER 2: Literature Review and Problem analysis

2.1 Overview of Crop Simulation

Climate, seasonal weather conditions, soil, management and genotype affect crop growth, yield and resources use. Crop simulation models have been recognized as useful tools to simulate impact of environmental factors and management factors on crop growth and yield. The environmental factors can be separated into the aerial environment and the soil environment. The aerial environment information comes from a record of weather conditions such as temperature, solar radiation and precipitation. The soil environment information includes soil water status, the availability of soil nitrogen and other nutrients, the aeration status of the soil and the spatial distribution of the root system. Management factors can be crop scheduling on such as planting date, fertilizer management and harvesting date. The simulation of crop development, growth, and yield involves evaluating stages of crop development, the growth rate and the partitioning of biomass into growing organs. All of these processes are dynamic and are affected by environmental and cultivar specific factors (Tsuji, et al., 1998).

Crop models have been using to simulate effects of the physical medium, climate and soil condition, and crop management schedule variations, e.g. planting date and fertilizing date, on crop production and environment at a farm scale. There are two main fundamental concepts for crop model, including comparing the observed with simulation phenological events and comparing the observed result with simulated crop growth and yield. They can be used to quantify the gaps between potential and actual yield, to evaluate management options and to determine environmental impacts. They can also be utilized for yield predictions before harvesting at a specific season and location. Crop models enables us to quantify variability of crop growth or yield resulting from uncertainty in seasonal weather conditions and to predict the long-term impacts of climate change and land use options (Honda & et.al., 2016). Thus, crop simulation models can potentially provide a unique means of quantifying the potential impacts of climate change on crop performance and water use requirement in the evaluation of adaptation strategies. There are various crop models around the world, of which the ones developed by the three main groups have dominated their evaluations and applications in Asia (Timsina & Connor, 2000).

These groups are listed as follows.

1. Decision support system for agro-technology transfer (DSSAT) and crop estimation through resource and environment synthesis. DSSAT package consists of, such as, DSSAT-CERES, DSSAT-GUMCAS, DSSAT-CANEGRO. They have been developed under the international benchmark sites network for agro-technology transfer (IBSNAT) project in USA (Tsuji, et al., 1998).

2. Dutch-Origin models, especially ORYZA 2000 (Bouman, et al., 2009). This model is jointly developed by the system analysis for rice production (SARP) project, Wageningen University and Research Center (WUR) and International Rice Research Institute (IRRI). It focuses on only rice simulation.
3. Agriculture production system simulator (APSIM) models (APSIM, 2016). They are jointly developed by several organizations in Australia in the name of Agriculture Production Research Unit (APSRU) group.

2.1.1 Data requirements for model operation and calibration

Typical crop simulation models require information on location, weather, soil profile, management and initial conditions for model operation (Tsuji, et al., 1998), although details of the requirements are different. Models operating at a daily time step is the most appropriate for application to crop production and environmental sustainability problems. (Tsuji, et al., 1998) defined a minimum set of data as shown in Table 1 that could be widely for models operation. The defined minimum data set can be a guideline for data preparation. Various data are required for running crop models. Each data source provides data in its proprietary format and protocol. They cause time consuming in iteration of data preparation.

Table 1. Minimum data requirement for model operation

Item	Detail
1. Location	Latitude and longitude
2. Weather	Daily solar radiation, maximum and minimum temperatures, precipitation
3. Soil profile	Basic profile characteristics by soil layer and soil analysis
4. Management	Cultivar name and type, planting information, irrigation and water management, fertilizer application, chemical application
5. Initial condition	Previous cop, root, water, ammonium and nitrate by soil layer

Before operating the model, model calibration need to be performed. Data set which are mentioned in Table 1 and crop performance information shown in Table 2 are required.

Table 2. Crop performance information for model calibration

Item	Detail
Crop performance	Date of emergence
	Date of flowering
	Date of onset of bulking in vegetative storage organ
	Date of physiological maturity
	Leaf area index and canopy dry weight at 3 stages during life cycle
	Canopy height and breadth at maturity
	Yield in dry weight terms
	Canopy dry weight or harvest index

	Harvest product individual dry weight e.g. weight per grain
	Harvest product number per unit at maturity e.g. seeds per spike
	Damage level of pest
	Number of leaves produced on the main stem
	N percentage of economic unit
	N percentage of non-economic parts

2.1.2 Simulation of crop yield

Crop growth simulation consists of three main areas; growth duration, growth rate and stresses which influence the two processes. Growth duration is extremely important in the determination of potential crop yields.

2.2 Decision support system for agrotechnology transfer: DSSAT

DSSAT was developed by an international network of scientists, cooperating in the International Benchmark Sites Network for Agro-technology Transfer project (IBSNAT), for facilitating the application of crop models in a systematic approach to agronomic research (Tsuji, et al., 1998). Knowledge about soil, climate, crops, and management for making better decisions about, e.g., transferring production technology from one location to another where soils and climate are different, are considered. DSSAT aims at creating system for predicting behavior of crops according to environment. It was developed to support decision-makers to overcome limitation in time and human resources required for analyzing complex alternative decisions. Moreover, it is also a framework for researchers to integrate new knowledge and to apply them to research questions. The DSSAT Cropping System Model (CSM) simulates growth and development of a crop over time, as well as the soil water, carbon and nitrogen processes and management practices. The main component of DSSAT CSM is shown in Figure 1.

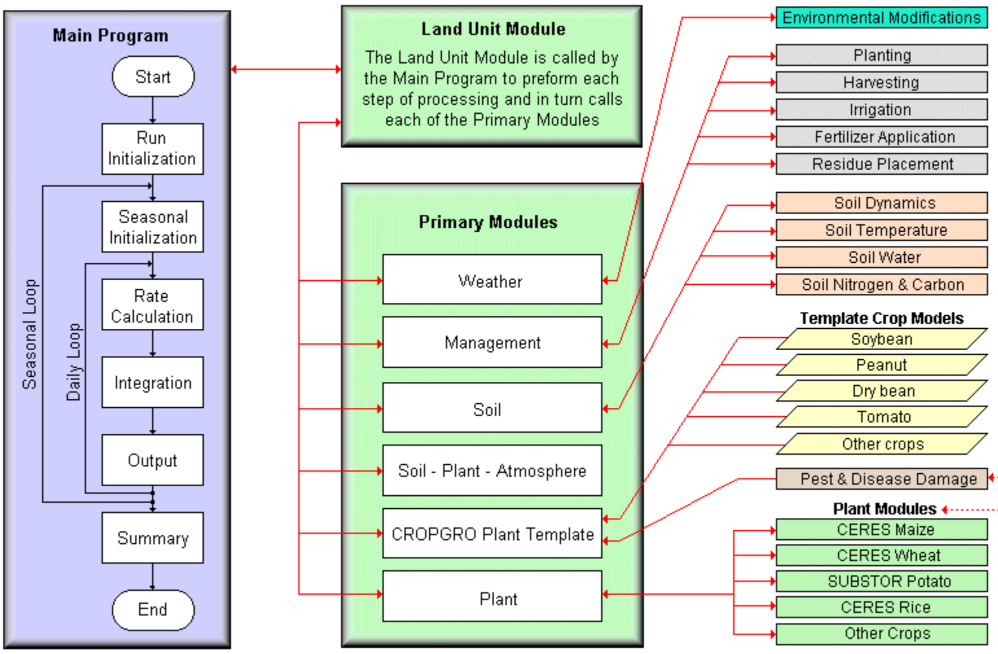


Figure 1. Overview of the components and modular structure of the DSSAT cropping system model Source: (Tsuji, et al., 1998)

DSSAT is composed of various crop models that are executed under one shell. The available DSSAT crop models are the CERES models (for barley, maize, sorghum, millet, rice and wheat); the CROPGRO models (for dry bean, soybean, peanut and chickpea); root crops (for cassava and potato) and other crops (for sugarcane, tomato, sunflower and pasture). The crop model architecture differs from one model to another. Less than one shell (interface), simulation controls and management scenarios can be invoked in the system to simulate the crop growth. The model can simulate seasonal sequential cropping systems and a single cropping (Tsuji, et al., 1998). Crop models are useful but have internal complex structure.

Table 3. Summarizing output files of DSSAT CSM

Output File Name	Type of File	Description
Error.OUT	Detail	Echo of screen errors
ET.OUT	Daily	Soil-Plant-Atmosphere (evaporation and transpiration)
ETPhot.OUT	Daily	Leaf level photosynthesis (CROPGRO crops only)
Evaluate.OUT	Summary	Measured vs. observed values
FloodN.OUT	Daily	Flooded field nitrogen processes
FloodW.OUT	Daily	Flooded field management
List.OUT	Auxiliary	List of unit assignments for output files
OUTPUT.PLIST	Auxiliary	List of output files
Overview.OUT	Summary	Detail seasonal summary of plant growth processes
Pest.OUT	Daily	Pest and disease damage
PlantC.OUT	Daily	Plant Cardon (CROPGRO crops only)
PlantGRO.OUT	Daily	Plant growth
PlantN.OUT	Daily	Plant Nitrogen
PlantNbal.OUT	Summary	Seasonal balance of plant nitrogen processes (CROPGRO crops only)
SoilC.OUT	Daily	Soil carbon
SoilN.OUT	Daily	Soil Nitrogen
SoilNbal.OUT	Summary/Daily	Seasonal and daily balances of soil nitrogen processes
SoilTemp.OUT	Daily	Soil Temperature
SoilWat.OUT	Daily	Soil water
SoilWatbal.OUT	Summary/Daily	Seasonal and daily balances of soil water processes
Somlit1.OUT	Detail/Daily	Soil organic matter detailed by soil layer
SOMLITC.OUT	Daily	Soil organic carbon
SOMLITM.OUT	Daily	Soil organic nitrogen
Summary.OUT	Summary	Seasonal summary of simulation
Warning.OUT	Detail	Warning issued by various modules related to conditions of simulation
Weather.OUT	Daily	Weather
Work.OUT	Detail	Simulation details for CSCERES wheat or barley simulations only

There are several output files generated by DSSAT CSM as describe in Crop models are useful but have internal complex structure.

Table 3. The main output which describe the development stage and amount of yield is Summanry.OUT.

2.3 Weather generator models

Weather generators are statistical models, which are useful for generating synthetic weather sequences or weather realizations (Hanse & Ines, 2005). They are essential for crop yield predictions and risk assessment because of their ability to simulate seasonal climate in advance of the growing season. Sub-seasonal to seasonal (S2S) forecast, from two weeks to seasonal time scales, are regularly released by several international agencies such as World Meteorological Organization (WMO) (WMO, 2012). When linked with crop growth models, weather generators allow simulation of crop growth and distributions of expected yields (Wilks & Wilby, 1999). Together with crop models, they are used for implementing decision support systems for agricultural risk management related to climate and weather uncertainties. The results of crop models can provide expected yield distributions to farmers, e.g., the ones provided by Tomorrow's Crop framework (Teeravech, et al., 2013). The resulting yield distributions represent not only the average value but crop yield's stability and associated risks, which are more useful information for decision making. Daily weather realizations from weather generators have been used not only in agriculture but also in various fields, such as water engineering design, ecosystem and climate change simulations, because observed meteorological data are often inadequate in terms of their temporal and spatial coverages (Wilks & Wilby, 1999). They have been utilized to fill up missing historical data, assess the impact of climate change such as droughts, rainfall pattern change and extreme temperature (Wilks & Wilby, 1999).

Most of the weather generators have been developed as stand-alone programs using programming languages suitable for scientific computations, such as, Fortran and Pascal. However, stand-alone programs developed in these programming languages cannot readily serve clients over the Internet as a service, and cannot be linked with other web services, e.g., web-based simulations. In this regard, the usability of weather generators is limited. Similarly, weather generators' output formats can vary and metadata are not included resulting in their lack of portability and interoperability. Nowadays, some climatological data are available online, e.g. IRI/LDEO Climate Data Library (IRI, 2016), daily Agro-meteorological Grid Square Data System (Chinnachodteeranun & Honda, 2016), among others, that can be retrieved and transferred over the Internet, however, their data formats and APIs are oftentimes different. Weather generators have been developed as standalone applications, while decision support systems (DSS) for farm management are now being developed as web applications. Processing and providing climate and weather data for DSS web applications, however, is still a bottleneck.

2.3.1 Stochastic weather generator

Weather generators are models that use stochastic method for generating synthetic daily weather data (realizations). Long-term climate data are used to calibrate the stochastic model. Weather generators are often used to generate weather scenarios as input to crop models in order to study the impacts of climate variability to crop production (Hanse & Ines, 2005).

Two stochastic weather generators are involved in this research. The first model is used for stochastic disaggregation (temporal downscaling), called, DisAg (Hanse & Ines, 2005). The second model is a parametric downscaling method for probabilistic seasonal climate forecasts (SCF), called, predictWTD (Ines & Han, 2015). The predictWTD embeds the DisAg program inside of its algorithm which expands the capability of DisAg to downscale the full distribution of SCF. The detail of the stochastic weather generators included in this paper are described in (Hanse & Ines, 2005). Some basic concepts are described here, in order to understand better the data requirement and data flow, which are important for designing and implementing them as web services. Figure 2 presents the schematic for DisAg. Long-term climate data that consists of solar radiation (SRAD), maximum temperature (TMAX), minimum temperature (TMIN), and rainfall (RAIN), is required for calibrating the stochastic models.

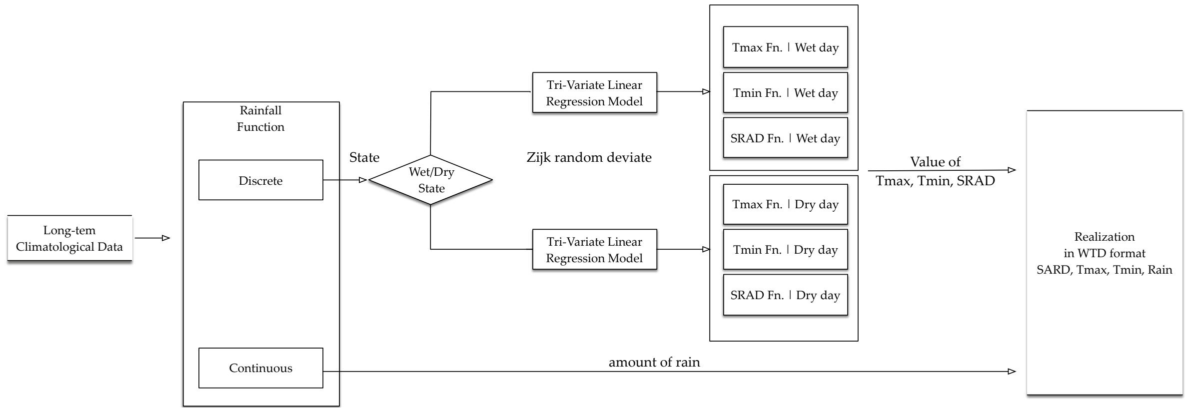


Figure 2. Overview functions of DisAg weather generator

The rainfall model in DisAg is a stochastic model with discrete (rainfall occurrence) and continuous (rainfall intensity) processes, as shown in Equation 1

Equation 1

$$R = f(\alpha_c, \emptyset_c)$$

where:

R is rainfall function

f is an operator that maps discrete and continuous model parameters with rainfall

α_c *is set of discrete model parameters based on climatology*

$\emptyset_c$ *is set of continuous model parameters based on climatology*

The discrete model is a rainfall occurrence model that decides the state of wetness or dryness on a day, based on a two-state Markov chain model. DisAg uses a hybrid two-state Markov model by which, if yesterday was wet, the probability that today is wet is based on 1st order Markov model, while if yesterday was dry, it uses a

2nd order Markov model. The continuous model in Equation 1 is a function for rainfall intensity, a hyper-exponential distribution, as used by DisAg.

There are several methods that users can choose in DisAg to downscale or disaggregate monthly forecasted rainfall by inverting Equation 1:

1. Matching monthly rainfall amount. This is an iterative process that generates stochastic rainfall sequences, which exactly match total rainfall target. The iterative process stops when the total of monthly rainfall is within 95% accuracy, and then rescaled to match exactly the target rainfall amount.
2. Conditioning rainfall intensity. This generates rainfall realizations by adjusting rainfall intensity model parameters of Equation 1 based on target monthly rainfall, while holding rainfall occurrence parameters in their climatological values (derived from long-term climate data).
3. Conditioning rainfall frequency. This generates realizations by adjusting rainfall frequency model parameters of Equation 1, while holding rainfall intensity model parameters in their climatological values.
4. Combinations of 1 to 3.

SRAD, TMAX and TMIN are generated based on a tri-variate normal model conditioned on rainfall (Ines & Hansen, 2006).

The predictWTD enhances DisAg's capability by downscaling the full distribution of seasonal climate forecast. Seasonal climate forecasts are usually issued in tercile probabilities, which are probability of below-normal (BN), near-normal (NN), and above-normal (AN) (Figure 3) (Min, et al., 2016). Climate forecasting can facilitate risk assessment, preparation and adaptation for global decision-making and planning in various sectors such as energy, water and food security (IRI, 1997). The International Research Institute for Climate and Society (IRI) has been issuing seasonal climate forecasts of precipitation and temperature since 1997 (IRI, 1997) using a two-tiered dynamically based multi-model prediction system (Barnston, et al., 2010). It issues 3-month period forecasts, which are updated every month. The predictWTD can generate weather realizations based on climatological distribution, or from actual seasonal climate forecast (see Figure 3) (Ines & Han, 2015).

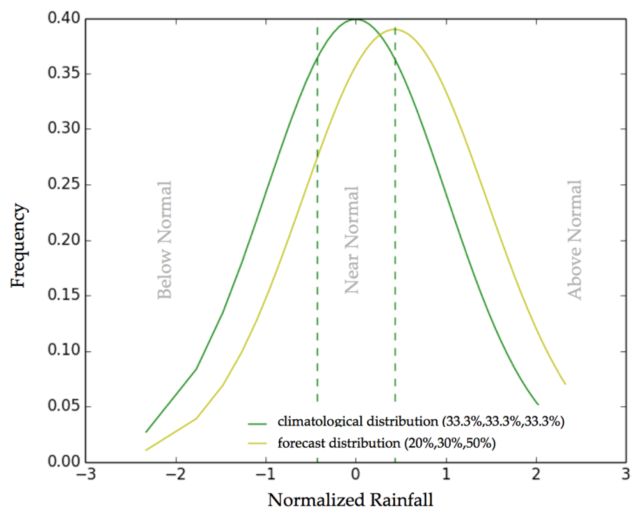


Figure 3. Comparison graph between climatological distribution and forecast distribution

2.4 **Climatological data**

Climate and weather information are essential for farm management and decision support systems in agriculture. Farmers gain their knowledge by looking at how crops respond to historical climate conditions. Information of future weather and climate in the coming cropping season will help farmers make proper plans for farm management. Based on these requirements, agronomists and application developers provide decision support systems, which require historical climate conditions and future weather conditions of the coming season for farmers. Good accessibility to farm-scale climatological data is one of the most important requirements for developing decision making systems, such as DSSAT (http://dssat.net), which is being developed through international collaboration. Gridded climate data are readily available, which provide historical climate and weather data. For example, The National Agriculture and Food Research Organization (NARO) has been providing gridded climatological data for the whole of Japan (Hiroyuki, et al., 2016) (NARO Agric.Rec.Cent., 2016). Gridded data are very useful for regional decision making, such as developing crop suitability maps or estimating regional crop development, because they provide weather and climate data in a spatially-distributed way. Long-term climatological data are vital for analyzing the impact of climate changes on cropping, as well as modeling future weather scenarios. On the other hand, farmers' interests focus more on each individual farm, which is a specific point; thus, decision support systems for farmers are usually made to utilize data sources that have a point-based structure. In general, point-based weather information can be obtained from weather stations, although long-term climate is hardly available in most situations. Official weather stations provide a point-based data source; however, they are often located far away from farms. Fusing gridded data with local point data could provide more useful information for decision support systems for agriculture. The heterogeneity of the data source is increasing as the field sensor network system is expanding to agriculture fields under the concept of the Internet of Things (IoT). In this situation, the standard web API that secures the interoperability on various meteorological data and other agro-environmental data is becoming more important.

2.4.1 Weather realization

Weather realization is synthetic climate data as output of weather generator. In this research, weather realizations are generated and written in WTD or WTDE format. One WTD file represents one weather realization in the specified time span. The number of WTD files, which represents to total number of realizations, is controlled by the parameter file. For example, if number of realizations in the parameter file is 100, then 100 WTD files are generated. The filename of weather scenario is based on the weather station name followed by scenario number, for example, the station name is MEMU then the first scenario will be named MEMU0001.WTD.

2.5 Interoperable climatological data

Many organizations have been working on monitoring and recording weather conditions and then providing those data on the Internet. For the convenience of users, some organizations operate climate data portals while serving data from other organizations. Each data source provides data in their proprietary formats with different protocols. Weather realizations, which are generated from weather generators are in WTD or WTDE format. This growing number of data formats and protocols causes a problem for data exchange and application development. To enable information interoperability from observed and generated data by models, open standard web API can play an important role.

2.5.1 Sensor Observation Service (SOS)

SOS is a standard protocol and application program interfaces (APIs) which was adopted as an OGC standard in the SWE framework for accessing measured sensor observations and sensor metadata registered. Official OGC implementation specification version 2.0 is available on OGC's homepage (Open Geospatial Consortium, 2012). Clients can obtain point weather data through SOS API in the same format and mechanism even though data are from different data source or structure. Once users experience SOS API, they can retrieve data from any source with SOS support. The same program can be reused without modification. SOS API assures interoperability of point-based weather data in time series format.

The main objectives of SOS are accessing to observed data in a standard way and mediator of data exchange and sensor system while hiding heterogeneous structure of sensor data formats and protocols. Three core operations required for SOS are;

1. GetCapabilities: this operation allows clients to access metadata and a list of available operations on SOS server. The request and response format is defined in (Open Geospatial Consortium, 2016).
2. DescribeSensor: it is the operation for retrieving metadata of physical sensors. Sensor Model Language (SensorML) is used for encoding sensor metadata (Open Geospatial Consortium, 2016).
3. GetObservation: this operation is for accessing observed data by allowing spatial, temporal and thematic filtering. The data are returned in O&M document encoding approved as an OGC standard, which is XML format (Open Geospatial Consortium, 2016). O&M is used for encoding data observed by sensors. The observation comprises timestamp, value, observed property and feature of interests.

There are several examples of applications and services, which manage weather data and metadata, using SOS API such as agriculture service application (Honda, et al., 2014) and crop simulation application (Chinnachodteeranun, et al., 2015). These applications have been providing information to farmers in Japan. They retrieve weather data from various kinds of sensors from various sources through the same program. National Oceanic and Atmospheric Administration (NOAA) has been utilizing SOS API for providing weather data such as on (National Oceanic and Atmospheric Administration, 2016).

SOS API is suitable as an intermediate layer for providing weather data to applications. A system can request metadata and data without knowing how the data are stored in the database, but just following the API specification. Applications do not need prepare data access routines for each data source. This reduces the time and cost of programming. Honda, Ines, et al. (2014) demonstrated efficiency of SOS in an agricultural application that accesses several different sensor systems.

Ministry of Internal Affairs and Communications in Japan accepts SOS as one of the standard APIs for managing agro-environmental sensor data in agriculture (Prime Minister of Japan and His Cabinet, 2015).

2.5.2 International Consortium for Agricultural Systems Applications (ICASA) standard

International Benchmark Sites Network for Agrotechnology Transfer (IBSNAT) project (Uehara & Tsuji, 1993) developed ICASA standards. ICASA defined standard data formats for documenting experiments and modeling crop growth and development (White, et al., 2013) . The ICASA standards are implemented in plain text in a hierarchical arrangement with major separations among descriptions of management practices or treatments, environmental conditions (soil and weather data), and measurements of crop responses. The goal of the ICASA standards is to provide a reliable and flexible structure for documenting experiment, detailed information is required on weather, soil, crop cultivars, weeds, diseases, pests, and crop management, along with measurements of crop growth and of dynamic soil characteristic.

Most weather generators and crop models require daily weather data as inputs. ICASA standard format "WTH" provides a basis for writing daily weather data (White, et al., 2013). WTH format is written in plain text, with special syntax and formatting rules (Figure 4). The weather generators used here accept daily weather data in the same format as WTH file (yearly), and longer-term weather files called WTD or WTDE file (multiple years). Weather file in WTH format is used in DSSAT model, one of the most popular crop simulation platforms used today.

```
@  DATE  SRAD  TMAX  TMIN  RAIN
2015001  17.5  26.1  16.1   0.0
2015002  16.2  29.5  14.7   0.0
2015003  17.3  27.9  14.9   0.0
2015004  17.7  32.9  17.7   0.0
2015005  16.0  30.9  15.8   0.0
```

Figure 4. Example of WTH, WTD and WTDE daily weather file, DATE format is YYYYDOY, (DOY is Day of year), SRAD=Solar radiance in MJ/m2-day, TMAX, TMIN are maximum and minimum temperature in Celsius, RAIN is precipitation in mm.

2.5.3 MetBroker

MetBroker (Ninomiya, et al., 2007) was one of the first systems that introduced the concept of interoperability between clients and legacy weather databases. The main concept of MetBroker was to provide client applications with single method to access different weather data sources, by which it absorbs the heterogeneity of each database. MetBroker has single API for clients, while it holds a pool of drivers that interface with each heterogeneous weather database. When a new database is connected to MetBroker, a new driver is developed as plug-in and added to the pool. MetBroker is a collection of Java classes, and clients can easily embed in their code. MetBroker provided web access portal for clients to obtain data from various databases in the same format.

Several applications have been developed on top of MetBroker. For example, (Laurenson & Ninomiya, 2001) developed a simple application for predicting maturity and harvesting date of Maize. (Laurenson, et al., 2001) developed a tool to analyze the risk of extreme climatic events. These applications demonstrated the effectiveness of MetBroker, then the importance of interoperability. MetBroker is rather bound to Java programing language. Recently, applications are developed as web applications in various programming languages such as Python and PHP. To fulfill current trend of development, web API can be applied to support crossing platform development.

2.6 Soil Database

Computer simulation models have recently become more common and acceptable for impact assessment studies and for supporting policy decisions on different temporal and spatial scales. Soil data are one of the main inputs for running simple to complex models such as crop simulation model (Tsuji, et al., 1998). Running crop models, detailed, extensive, quantitative and geo-referenced databases covering small to large regional areas are needed. Unfortunately, often these data- bases are neither in the formats nor in the specific dimensions required for crop and environmental models (Batjes, 2012).

Soil data requirements obviously vary, depending on the processes that are simulated within a particular model, but for many management- orientated models, there are considerable overlaps.

There are many situations in which model users simply do not have access to such soil data. In such cases, there are only limited options, all of which may involve rough estimates and plain guesses. One is to use generic data that are typical for texture of soil; for example, there are several such profiles in the DSSAT soils database, such as "deep silty clay" and "shallow sandy loam". A second option is using a soil map. With sufficient spatial resolution, it is possible to identify the soil class for a particular location, but relatively few of the required variables (Table 4) are included in soil map information. It would be very helpful to have good examples of soil input files for a particular profile that has similar type to the profile that is to be used for a simulation. Soils database includes many profiles from around the world into a format that contains the information typically needed as input to biophysical and ecosystem models (Table 4) and is suitable for operating and running the DSSAT crop simulation models.

The International Soil Reference and Information Centre (ISRIC) developed a detailed geo-referenced soil database, entitled "World Inventory of Soil Emission Potentials" (WISE). Rules were applied to most soil parameters (Table 4) to verify if the value under evaluation fell within a range of values already established in the literature. The established limits for maximum and minimum values and associated references are shown in Table 5.

Table 4. Soil data input requirements for a daily time-step crop simulation model, such as the Cropping System Model (CSM) (Ritchie et al., 1990; Jones et al., 2003)

Parameter name	Meaning	Units
General data		
SLTX	Texture code of surface layer	unitless
SLDP	Soil depth	cm
SLDESCRIP	Soil description or local classification	unitless
COUNTRY	Country	unitless
LAT	Latitude	unitless
LONG	Longitude	unitless
SCSC FAMILY	Soil class	unitless
General and soil surface information		
SCOM	Soil color (Munsell color system)	unitless
SALB	Albedo	unitless
SLU1	Evaporation limit	cm
SLDR	Drainage rate	fraction day^{-1}
SLRO	Runoff curve number	unitless
SLNF	Mineralization factor	0–1 scale
SLPF	Soil fertility factor	0–1 scale
SMHB	pH in buffer determination method	unitless
SMPX	Extractable phosphorus determination code	unitless
SMKE	Potassium determination method	unitless
First tier		
SLB	Depth until base of layer	cm
SLMH	Master horizon	unitless
SLLL	Lower limit of plant extractable soil water	cm^3 cm^{-3}
SDUL	Drained upper limit	cm^3 cm^{-3}
SSAT	Saturated upper limit	cm^3 cm^{-3}
SRGF	Root growth factor	0–1 scale
SSKS	Saturated hydraulic conductivity	cm h^{-1}
SBDM	Bulk density (moist)	g cm^{-3}
SLOC	Soil organic carbon concentration	%
SLCL	Clay (<0.002 mm)	%
SLSI	Silt (0.002–0.05 mm)	%
SLCF	Coarse fraction (>2 mm)	%
SLNI	Total nitrogen concentration	%
SLHW	pH in water	unitless
SLHB	pH in buffer	unitless
SCEC	Soil cation exchange capacity	$cmol^{(+)}$ kg^{-1}
SADC	Soil adsorption coefficient (anion exchange cap.)	0–1 scale
Second tier		
SLPX	Extractable soil phosphorus concentration	mg kg^{-1}
SLPT	Total soil phosphorus as P concentration	mg kg^{-1}
SLPO	Soil organic phosphorus concentration	mg kg^{-1}
$CACO_3$	Soil $CaCO_3$ concentration	%
SLAL	Soil aluminum concentration	mg kg^{-1}
SLFE	Soil iron concentration	mg kg^{-1}
SLMN	Soil manganese concentration	mg kg^{-1}
SLBS	Soil base saturation	%
SLPA	Soil phosphorus isotherm A	mmol kg^{-1}
SLPB	Soil phosphorus isotherm B	mmol kg^{-1}
SLKE	Exchangeable potassium soil concentration	$cmol^{(+)}$ kg^{-1}
SLMG	Exchangeable magnesium concentration	$cmol^{(+)}$ kg^{-1}
SLNA	Exchangeable sodium concentration	$cmol^{(+)}$ kg^{-1}
SLSU	Soil sulfur concentration	$cmol^{(+)}$ kg^{-1}
SLEC	Soil electric conductivity	dS m^{-1}
SLCA	Soil calcium concentration	$cmol^{(+)}$ kg^{-1}

Table 5. Rules to identify potential discrepancies observed in the WISE soil database (Ritchie et al., 1990; Jones et al., 2003)

Soil variable	Range	Reference
Soil color	DSSAT uses brown, red, black, gray, yellow, and yellow-red. Soils that did not have a color code were classified as 'brown'.	Tsuji et al. (1994); Gijsman et al. (2007).
Soil albedo	Albedo is estimated from the soil color or the top layer. Ranges from 0.09 for a black soil to 0.17 for a yellow soil.	Ritchie et al. (1990); Gijsman et al. (2007).
Evaporation limit	Less or equal to 12.0 mm d^{-1}.	FAO (1990).
Clay fraction	0–100%	Gee and Bauder (1986).
Silt fraction	0–100%	Gee and Bauder (1986).
Coarse fraction	0–100%	FAO (2006).
Drainage rate	Seven permeability classes: very poorly drained 0.01), poorly drained (0.05), somewhat poorly drained (0.25), moderately well drained (0.40), well drained (0.60), somewhat excessively drained (0.75), and excessively drained (0.85).	Ritchie et al. (1990).
Runoff curve number	Soils are classified by slope and by hydrologic group, runoff curve number ranges from 61 to 94.	Ritchie et al. (1990).
Mineralization factor	1	Gijsman et al. (2007).
Soil fertility factor	1	Gijsman et al. (2007).
Soil depth until the base of the layer	Use only the lower limit of a soil layer or horizon.	Gijsman et al. (2007).
Hyd. coefficients: Lower limit, drained upper limit, and saturated limit.	Values for lower limit should be less than values for drained upper limit. Values for drained upper limit should be less than values for saturated limit. For missing values pedotransfer equations were used for estimation.	Saxton et al. (1985).
Bulk density	0.5–1.8 g cm^{-3} for most soils except Histosols that can show low values as 0.2 g cm^{-3}.	Brady and Weil (1999); Wild (1993); FAO (2006)
Soil organic carbon	Most soil surface layers (0.2 m) seldom contain more than 5% soil carbon. High values could be associated with Histosols, Andosols, Czernozems and Kastanozems. It can be estimated from C:N = 10.	Buringh (1984); Batjes (1996); Eswaran et al. (1993); Brady (1990).
Total nitrogen concentration	Values should range between 0 and 0.5%. Higher values could be associated to Histosols, Andosols, Czernozems and Kastanozems, and Histosols. If not data is available, can be estimated from a C:N = 10.	Batjes (1996); Brady and Weil (1999).
pH in water(SLHW)/pH in buffer(SLHB)	pH measured in water (SLHW) should range between 3.5 and 9.0. pH measured in buffer (SLHB) should have a value lower than SLHW.	Brady and Weil (1999); Wild (1993); USDA-NRCS (2011).
Soil cation exchange capacity	Range from 0 to 45 $cmol^{(+)}$ kg^{-1} for most mineral soils. Histosols and Vertisols are the exception, reaching around 150 $cmol^{(+)}$ kg^{-1}.	Holmgren et al. (1993); Hemni (1980).
Soil $CaCO_3$ concentration	<50% for agricultural soils.	Brady and Weil (1999).
Soil aluminum concentration	Range from 0 to 12.4 $cmol^{(+)}$ kg^{-1}. At pH 5.5 and above exchangeable Al^{3+} is no longer present. Al chemistry is dominated by a complex mixture of hydroxyl-Al ions, many of them highly polymerized and virtually non-exchangeable.	Kamprath (1980); Singh and Talibudeen (1969); Aitken (1992); Dong et al. (1999); Hsu and Rich (1960); Marion et al. (1976); Lathwell and Peech (1964).
Soil base saturation	Range from 0 to 100%. Equal to 100% when no Al^{3+} + H^{+} is available in soil solution.	Brady and Weil (1999).
Exchangeable cations	Ca: <35 $cmol^{(+)}$ kg^{-1*} Mg: <20 $cmol^{(+)}$ kg^{-1*} K: <30 $cmol^{(+)}$ kg^{-1} Na: <20 $cmol^{(+)}$ kg^{-1} *Most Ca^{2+} and Mg^{2+} values on Vertisols were not modified since they exceeded greatly the established limits.	Hendershot and Duquette (1985); Tucker (1954).
Soil electrical conductivity	<16 dS m^{-1} for agricultural soils (except very saline soils).	Brady and Weil (1999).

2.7 Web API Framework

Web API leads to reusability, simplicity, extensibility, and clear separation of component responsibilities (Sun, 2009). Representational State Transfer (REST) is a web service architecture that focuses on a system's resources, including how resource states are addressed and transferred over HTTP (Rodriguez, 2015). A resource is an item of interest which can be application objects, database records, algorithms, and so on. Every resource is uniquely addressable using a Universal Resource Identifier (URI). Resources are manipulated using HTTP methods, GET, POST, PUT, and DELETE. HTTP methods are mapped to create, read, update and delete (CRUD) actions for a resource (Huang, et al., 2009). Basic usage of HTTP methods is illustrated as follow.

1. POST method is used for creating resource on a server.
2. GET method is used for retrieving a resource.
3. PUT method is used for changing a state of a resource or to update it.
4. DELETE method is used for removing or deleting a resource.

Resources have representations. A representation is a content in a HTTP message which can be represented in multiple formats. XML, JavaScript Object Notation (JSON), Atom, binary formats such as PNG, JPEG, GIF, plain text, and proprietary formats are used to represent resources. REST provides the flexibility to represent a single resource in multiple formats (Sun, 2009).

An interaction of requests between the client and server is stateless. The meaning is each request from the client to the server must contain all the information necessary to understand the request including credentials for the Resource Request Handler to process the request. A service should not rely on the previous request. REST has emerged as a promising alternative to SOAP-based services due to their simplicity, lightweight nature, and the ability to transmit data directly over HTTP

(Sun, 2009). Mainstream web 2.0 service providers such as Google, Facebook and Yahoo adopted REST for providing their services. Clients can be implemented using a wide variety of languages such as Java programs, Perl, Ruby, Python, PHP, and JavaScript.

RESTful Web services are generally accessed by an automated client or an application acting on behalf of the user. However, their simplicity makes it possible for humans to interact with them directly, constructing a GET URL with their Web browser and reading the content that is returned.

Exposing a system's resources through a RESTful web service is a flexible way to provide different kinds of services with data formatted in a standard way (Rodriguez, 2015). It helps to meet integration requirements system that needs to combine several data sources and services. Each data and service can be refered as a unique resource which means that a resource can be directly requested by one request. Due to the statelessness principle, REST makes the system really scalable since the servers do not keep any information from any of the clients. This means that every single machine that is part of the system is able to receive any kind of request from any of the clients and generate a proper response which is good for managing distributed resources.

2.8 Agriculture 4.0

Agriculture 4.0 is a modern agriculture form supported by Internet of Thing (IoT), sensors, network and cloud computing. The term Agriculture 4.0 adopted from the term industry 4.0. The main concepts are Cyber-Physical System (CPS) and data exchange (Hermann, et al., 2015). The fundamental principles are as follows.

1. Interoperability: It is key principle for Industry 4.0. The ability of machines, devices, sensors, and people to connect and communicate with each other via the Internet of Things (IoT) or the Internet of People (IoP) (Hermann, et al., 2015).
2. Virtualization: It is an ability to monitor physical processes. Data are linked to simulation models (Hermann, et al., 2015). The ability of information systems to create a virtual copy of the physical world by simulation models with sensor data. This requires the aggregation of raw sensor data to higher-value context information.
3. Decentralization: The ability of CPS to make decision by their own based on monitoring data. Tasks can be performed automatics as much as possible (Hermann, et al., 2015).
4. Real-Time Capability: The ability for collecting and analyzing data in real-time (Hermann, et al., 2015). First, the ability of assistance systems to

support humans for making informed decisions and solving urgent problems on short notice.

5. Service Orientation: Industry 4.0 is based on service oriented architecture. CPS has their functionalities as an encapsulated web service (Hermann, et al., 2015).
6. Modularity: Modules should have flexibly adaptation and expansibility.

Agriculture 4.0 comes with the concept of precision agriculture. It is an agricultural practice that utilizes data and technologies to make better decisions and management for minimizing resource usage while maximizing yield. Data are playing an important role in this era.

Agriculture 4.0 will be extremely efficient to agriculture. Characteristics of farm land and progress of crop growth vary even in a single farm. Adaptation of agriculture 4.0 will make it possible to obtain state of the farm and crop dynamically, then optimize treatments in a local context with very small granule, which is an ultimate precision agriculture.

CHAPTER 3: Development lesson

3.1 Developing web-based yield prediction application

We developed a web-based crop simulation system for wheat and rice named as Tomorrow's Crop (TMC). The application is available on http://tesla.isc.chubu.ac.jp. DSSAT is used as a simulation engine that calculates expected yield for different weather, climate and agronomic scenarios. The system links various data sources such as Daily agro-weather data generated by NIAES (Kuwagata, et al., 2011) from AMeDAS data (Japan Meteorological Agency, 2016) in Japan, IRI data sources (IRI, 1997), NASA data source and World Bank's climate change dataset. Japanese data sources are collected and stored in a cloud sensor infrastructure named cloudSense (Honda, et al., 2014). The cloudSense provides historical weather data, which are minimum temperature, maximum temperature, precipitation and solar radiation, via Sensor Observation Service (SOS) interface defined as OGC's standard API. The long-term weather data are seamlessly connected to a weather generator for generating 100 weather realization in order to reflect the uncertainty in weather scenario into expected yield. The application was developed in monolithic style which includes several components as visualized in Figure 5.

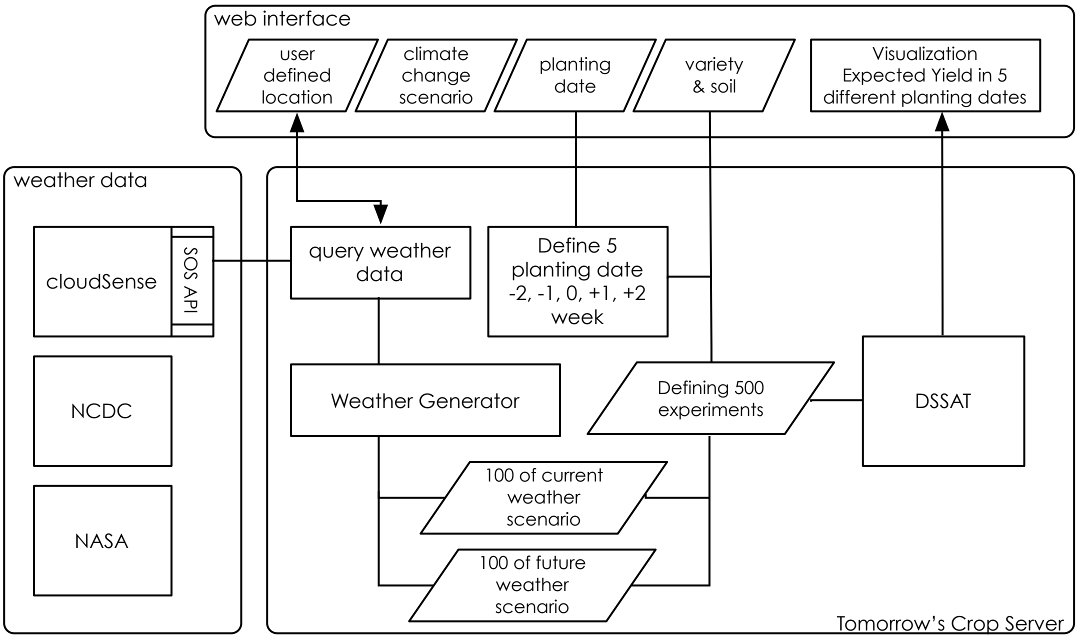


Figure 5. Tomorrow's Crop component diagram

Figure 6 visualizes a user interface when the application acquires the historical data and visualizes them on graphs. Users can set other factor of scenarios by giving a planting date, a variety of wheat and soil characteristics from an interactive web user interface as show in Figure 7. The result of the crop simulation, as shown in Figure 8, is expected wheat yield distributions shown as box-plots for five different planting dates. It supports decision making of farmers to set the best planting time window to optimize the yield considering the stability and risk.

Figure 6. User Interface of Tomorrow's Crop on acquiring long-term historical data

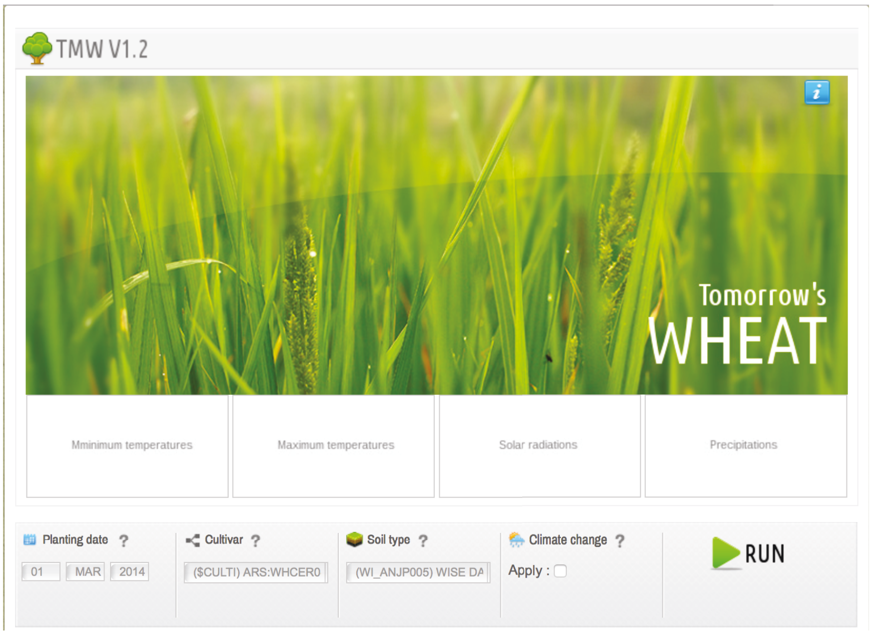


Figure 7. User Interface for setting simulation's parameters

Figure 8. Expecting yield with its distribution regarding planting date in boxplot graphs

Developing this kind of decision support application requires multidisciplinary knowledge on agronomy, climatology and computer science. Figure 9 visualizes development flow diagram of TMC. Various data sources are connected as input for running models. Several models, i.e. DSSAT and weather generator, and helper programs, i.e. SWGPRMADJ.f (a proprietary program written in Fortran language for climate projection analysis), Estimateprm.exe (a proprietary program for climatological analysis), exportPHNT.exe (a proprietary program written in C language for trimming weather realization), were compiled as executable components and run in the back-end.

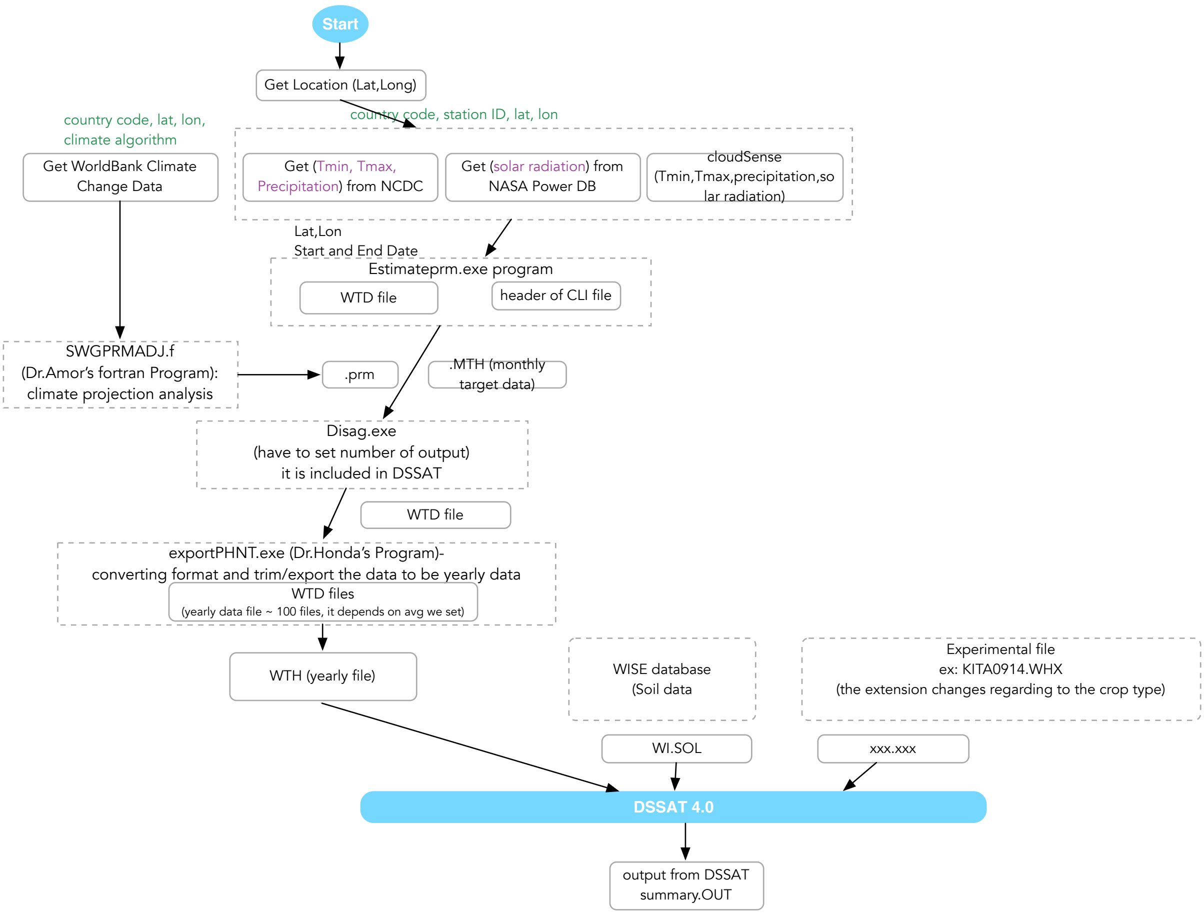


Figure 9. Development diagram of TMC

3.1.1 Procedure for preparing climatological data

we describe the general procedures in preparing climatological data for running legacy weather generators in this section. These steps must be performed often manually, as illustrated in Figure 10.

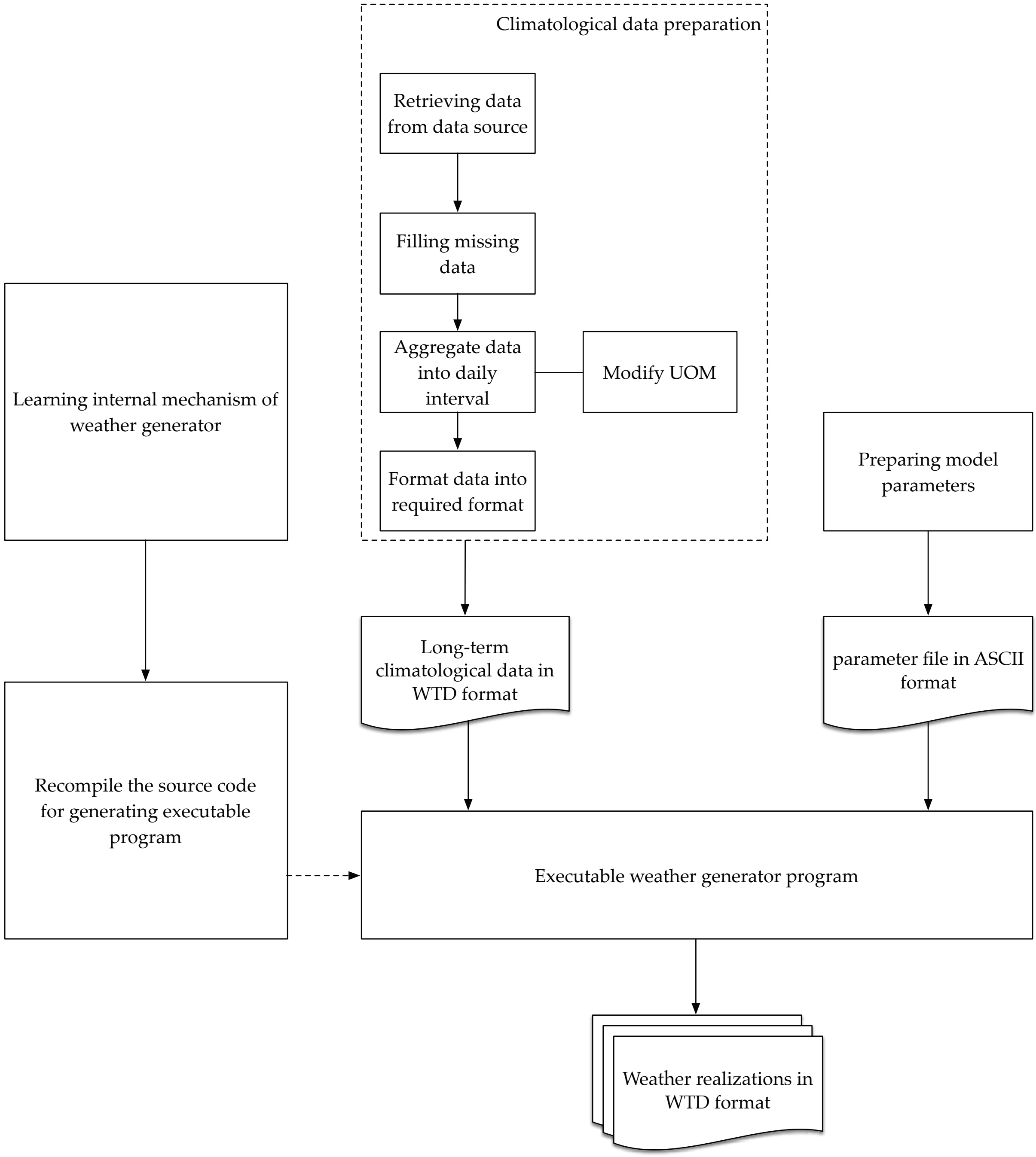


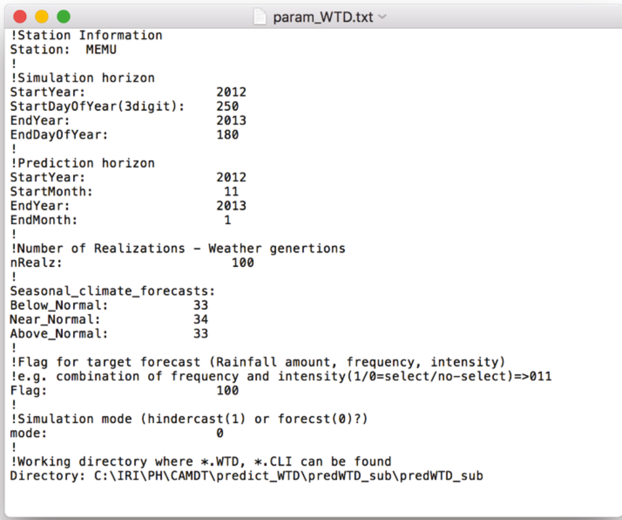
Figure 10. Climatological data preparation for weather generators

From the retrieval of data (from a data source), one needs to learn the data accessing protocol and format. After retrieving the data, all missing data need to be filled, as weather generators cannot process a dataset with missing data above a threshold volume. A weather station captures weather data at various time-intervals. In case of public weather station networks, they usually publish their daily data, but some stations capture data at every 5 or 10-minute intervals and publish those data as it is. In such a case, the data must be aggregated at daily-interval and its unit of measurement (UOM) must be changed, accordingly. Matching the UOM of each climate variable is very important; otherwise, there could be some hidden errors in the final dataset. Lastly, the aggregated daily data must be transformed in formats compatible with weather generators.

3.1.2 Procedure for running weather generator

This section describes the operation of weather generators. DisAg was implemented in Delphi, and predictWTD was written in Fortran. These programming languages are strong for scientific computations; however, they lack of good support for cross-platform and cross-languages portability, or compilers are expensive. Before running the weather generators, the programs must be recompiled to create executable programs so that they can be compatible with users' operating system. If the operating system changes, the programs will need to be recompiled. It is not always an easy task for users to find compilers suitable for their OS (e.g., mac OS, Linux and windows) to re-compile and re-create executable programs for their needs.

To operate the weather generators, two main input files are required. One is a long-term historical weather data file in WTD format, which is used for calibrating the stochastic models. The other one is a parameter file (ASCII) that provides the conditions for running the weather generator. The parameter file includes a weather station name that becomes the name of the generated weather file (WTD format), simulation period, number of realizations to be generated and seasonal climate scenario, as shown in Figure 11.




```
!Station Information
Station:  MEMU
!
!Simulation horizon
StartYear:                 2012
StartDayOfYear(3digit):    250
EndYear:                   2013
EndDayOfYear:              180
!
!Prediction horizon
StartYear:                 2012
StartMonth:                 11
EndYear:                   2013
EndMonth:                   1
!
!Number of Realizations - Weather genertions
nRealz:                      100
!
Seasonal_climate_forecasts:
Below_Normal:          33
Near_Normal:           34
Above_Normal:          33
!
!Flag for target forecast (Rainfall amount, frequency, intensity)
!e.g. combination of frequency and intensity(1/0=select/no-select)=>011
Flag:                      100
!
!Simulation mode (hindercast(1) or forecst(0)?)
mode:                      0
!
!Working directory where *.WTD, *.CLI can be found
Directory: C:\IRI\PH\CAMDT\predict_WTD\predWTD_sub\predWTD_sub
```

Figure 11. An example of a parameter file (e.g., param_WTD.txt) for running the weather generators

The following shows the contents of the parameter file (Figure 11) that set the conditions for running weather generator.

- Station: MEMU indicated that historical weather data as input to weather generator is in MEMU.WTD file
- Simulation horizon: is used for defining time span of crop calendar. In this example, it is for informing the crop model that the starting date of the cropping is on the day of year (DOY) 250 of year 2012 and harvesting date is on DOY 180 of year 2013.
- Prediction horizon: is used for defining time span of weather realization to be generated. In this example, weather realization will be generated from the first day of November, 2012 until 31st January 2013.
- nRealz: is for identifying number of realizations to be generated.
- Seasonal_climate_forecasts: is a seasonal climate forecast scenario parameter. In this case, the probability of rainfall being Below_Normal category is 33%, Near_Normal category is 34% and Above_Normal category is 33%. These tercile probabilities indicate a climatology scenario of seasonal rainfall for the season of interest.
- Flag: is to define the method of rainfall function which is described in the previous section. The digit "1" in the flag indicate that the corresponding method is selected. Matching method is referred in the

parameter file as "rainfall amount". Conditioning rainfall intensity method is referred as "intensity". Conditioning rainfall frequency method is referred as "frequency". The model developer suggests to set the Flag to 100 which means that the model will regenerate rainfall amount based on rainfall amount method.

- Mode: is to identify if the generator runs on a hindcast or actual forecast.
- Directory: is to define a working directory.

The format of param_WTD.txt file is strict. If the file is not in the right structure, the process will not work. Users need to understand every parameter for operating the model. This is a barrier for new comers who wanted to utilize weather generators as a product. The file needs to be in the same directory with the executable program. Operating conditions for running a weather generator using a plain text format is not convenient and can easily lead to errors e.g., file formatting mismatch. If we can bundle the weather generator operation including data input preparations as a web service, generating weather scenarios will be more convenient.

Most of the weather generators have been developed as stand-alone programs using programming languages suitable for scientific computations, such as, Fortran and Pascal. However, stand-alone programs developed in these programming languages cannot readily serve clients over the Internet as a service, and cannot be linked with other web services, e.g., web-based simulations. In this regard, the usability of weather generators is limited. Similarly, weather generators' output formats can vary and metadata are not included resulting in the lack of portability and interoperability. Nowadays, some climatological data are available online, e.g. IRI/LDEO Climate Data Library (IRI, 2016), daily Agro-meteorological Grid Square Data System (Chinnachodteeranun & Honda, 2016), among others, that can be retrieved and transferred over the Internet, however, their data formats and APIs are oftentimes different. Weather generators have been developed as standalone applications, while decision support systems (DSS) for farm management are now being developed as web applications. Processing and providing climate and weather data for DSS web applications, however, is still a bottleneck.

3.1.3 Procedure for running crop simulation

To run DSSAT as back end of the application, we need to recompile the Fortran source code to be able to call using command line in suitable environment with operation system (OS) environment. DSSAT is written in Fortran programing language under hundreds of files from several developers. To compile these files, it is not a straightforward procedure. We need to find a suitable compiler and libraries as well as cracking into the source code according to the difference of directory definition between Window OS and Linux OS. In spite of the good support from DSSAT community (http://dssat.net/210), huge effort and time are required for recompiling the code.

3.2 Problem statement from the development lesson

We encountered problems from developing TMC (Teeravech, et al., 2013). We needed to work on developing iterative data processing, starting from querying data from climatological data sources, followed by data quality control since weather generator could not process incomplete data series (contained missing data), and lastly formatting the data to make them compatible as input to the weather generator model. Moreover, legacy weather generators and crop model, written in different programming languages such as Delphi and Pascal, are not portable, hence a significant effort must be spent on compiling them and keeping them up-to-date for different Operating Systems (OS). Intergovernmental Panel on Climate Change (IPCC) reported that environment and climate scientists had to spend huge time and effort in data handling (Intergovernmental Panel on Climate Change, 2001). These processes are time-consuming and require meteorological knowledge and programming skills. For pre-processing data, such as filling missing data and data formatting, most of agronomists who run crop simulations do it manually or develop individual programs as they need. These bottlenecks must be addressed to increase efficiency of research and development in agriculture.

CHAPTER 4: Designing of the API integration platform

This section describes overview of the framework and relationship of each component and service as visualized in Figure 12. Three main data sources; climatological data source, soil data source and cultivar and management information, are connected into the framework. The framework is conceptually designed as multi-layer web service as some component need data from previous component, however, some component can work independently. Each component has its own service endpoint for clients who request the service directly. Meanwhile, they are internally wired for supplying input to the next procedure. Adding a data source into the framework (lowest layer of Figure 12) does not affect upper level layers. For example, climatological data source can be added and wired to Weather generator API without effect to any component or layer. A detail design and implementation of each service is described separately in Chapter 5 to 8.

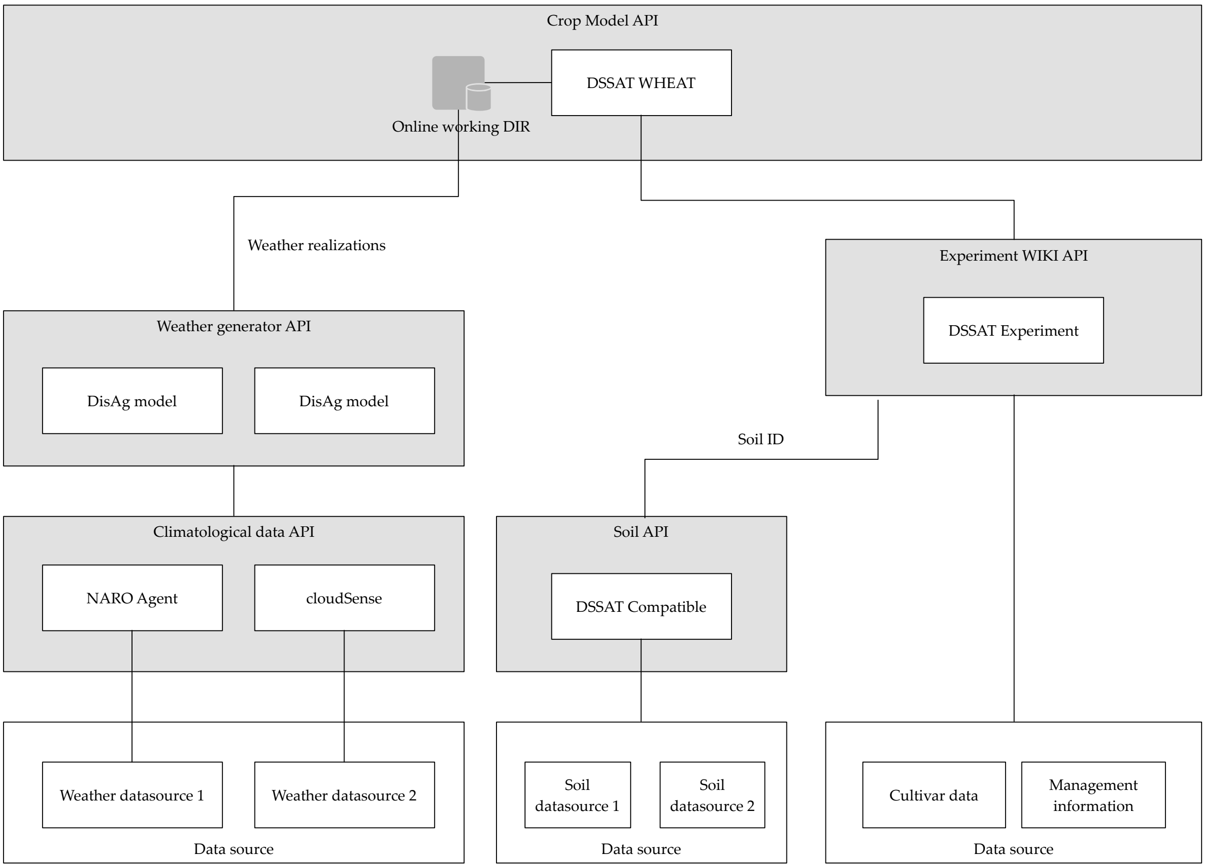


Figure 12. Overview diagram of API integration platform

The design constraints of the API integration platform are global accessibility through resource identification, interface standardization, service interoperability, and classification of user role. A detail of each constraint is explained as follows.

Global accessibility through resource identification; all resources, historical climatological data, weather realization, agronomic models, data transformation and soil profile in the system are designed to be defined in unique form in order to enable interaction over the network.

Interface Standardization; climatic service interoperability is an essential requirement of this development. Interfaces of retrieving historical climatological data and weather realizations are in a standard format so that users need not spend time on learning various interfaces. Historical weather data are provided in compliance with SOS interface. Weather realizations are provided in compliance with both ICASA standards and O&M format.

Interoperability of data and model; seamless data flow from data source to API is a key for the success in this development. Moreover, different models which are written in different programing language can be connected via the API integration platform.

Classification of user role; currently, the role of users are blended together. For example, users who want to aquire weather realization need to manage the weather generator program too. Classification of user role will separate users from service providers, modelers and end users such as researchers and software developers.

CHAPTER 5: Climatological data API

The primary objective of this chapter is to develop a single data access mechanism on a web API for obtaining climatological data in standard procedure. The climatological data API developed in this research can be wired to weather generator for generating weather realization. Agricultural applications can obtain climate data regardless of its data type, i.e., point-based or gridded data, and difference in data structure. Application developers need not to spend precious time on learning data structure of each data source one by one. Developers can obtain data using a same procedure regardless of the data organization whether gridded or point-based. A gridded climate data source from National Agriculture and Food Research Organization (NARO) is utilized for this development. With this data source, agricultural applications can seamlessly obtain weather data and long-term historical climate data to run simulations and scenario analysis.

5.1 Data Source

NARO developed a daily agro-meteorological grid square data system (AmGSD) (Ohno, 2014). AmGSD acquires based data from the Automated Meteorological Data Acquisition System (AMeDAS) operated by Japan Meteorological Agency (JMA). AmGSD provides gridded climate dataset at approximately 1-km resolution from 1980 to present and new weather data is updated every day. The data is available for the whole Japan in 6 areas as shown in Figure 13.

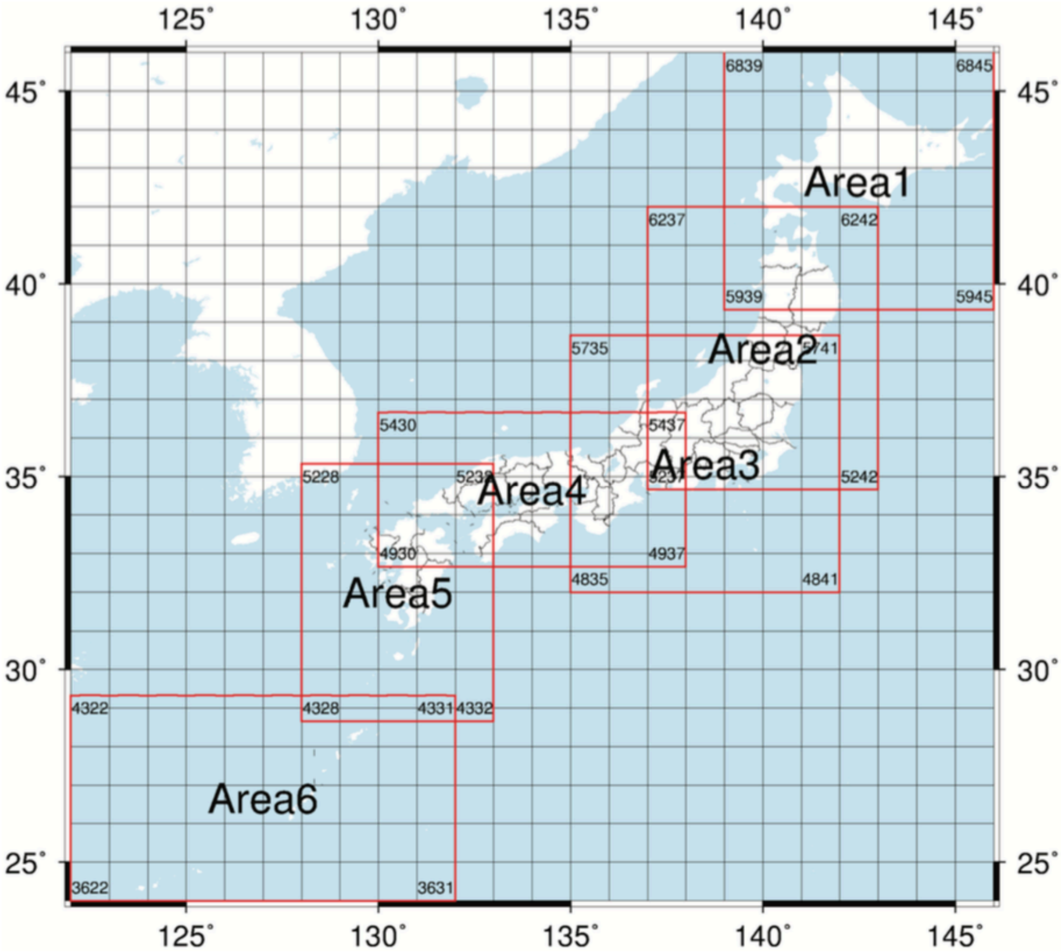


Figure 13. Coverage of 6 areas in AmGSD from (Ohno, 2014).

Each grid contains nine daily meteorological variables, which are average air temperature, maximum air temperature, minimum air temperature, average relative humidity, sunshine duration, global solar radiation, downward long-wave radiation, precipitation and average wind speed. AmGSD uses Japanese 3rd order mesh code that has 8-digit integer defined by Japanese Industrial Standard JIS X0410. Land use information for each grid is provided by National Land Information Division, MILT of Japan (National Land Information Division, MILT of Japan, 2016).

The data are encoded in Network Common Data Form (NetCDF). Each file contains one year of daily data of a specific meteorological variable. The file is separately stored in "Area/Year/Item" directory structure on their server. For example, maximum air temperature of Area1 in the year 1980 is kept in /Area1/1980/AMD_Area1_TMP_max.nc. Metadata is recorded separately from data, for example, land use information is kept in GEODATA NetCDF file in a separate directory.

Due to the complicated hierarchical structure and internally defined structure, it is not convenient for ordinary scientists and programmers themselves to access to AmGSD and convert the data into point-based climate data.

NARO provides AMD_Tools for connecting and retrieving data from AmGSD. AMD_Tools is written in Python programming language and available on https://ml-wiki.sys.affrc.go.jp/MeshUser/. Even if AMD_Tools is provided, a user still needs an extra program for calling the AMD_Tools as well as understanding the structure of data and coverage of each area.

We developed a system named NARO Agent that works as an intermediate layer between AmGSD and client. It translates gridded climate data into point-based data and then responds to clients' requests through SOS API. NARO Agent absorbs the complicated structure of AmGSD with SOS API and internal components. NARO Agent consists of several components as shown in Figure 14. Each component supports each other to provide sufficient SOS API for agricultural applications. To efficiently serve data for agricultural applications, several internal components such as caching, mesh system conversion, controlling multi-process and threading access modes are implemented.

NARO Agent is written in Python programing language on Flask framework (Armin, 2010). It has been deployed on Conoha (GMO Internet, 2016), an OpenStack cloud platform, for easy maintenance, scalability, and future migration purpose.

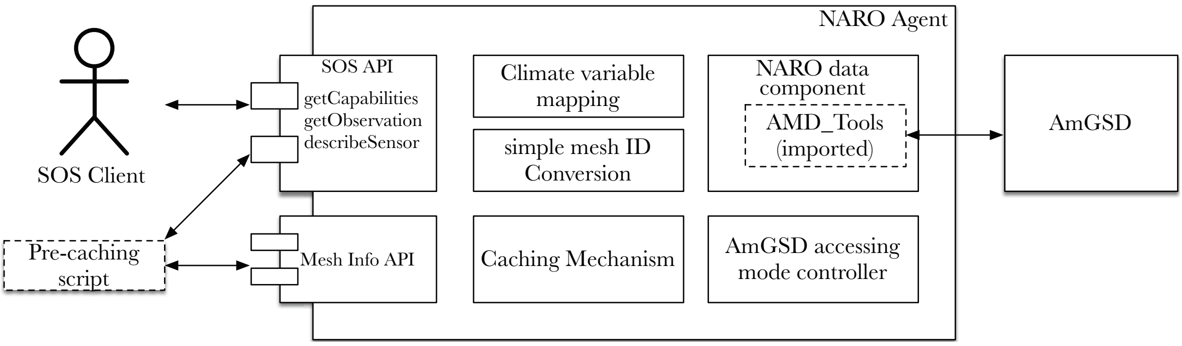


Figure 14. NARO Agent diagram

The detail of each component is described consequently.

1. NARO data component
2. AmGSD accessing mode controller
3. Simple mesh ID conversion
4. Climate variable mapping
5. SOS API
6. Mesh Info API
7. Caching mechanism

5.2 NARO data component

The NARO data component is for managing internal data of NARO Agent system. It also works on retrieving data from AmGSD. NARO data component imports AMD_Tools for retrieving data from AmGSD. Figure 15 demonstrates the data flow from AmGSD to a client, and how NARO data component processes the data. When there is a request from a client, NARO data component calls getData function of AMD_Tools for retrieving data and then the data is returned to AMD_Tools in an array format. NARO data component converts the data array into CSV file that is internal data model used for caching. Later, the data is converted into O&M format that is mandatory format of SOS, and returned to the client.

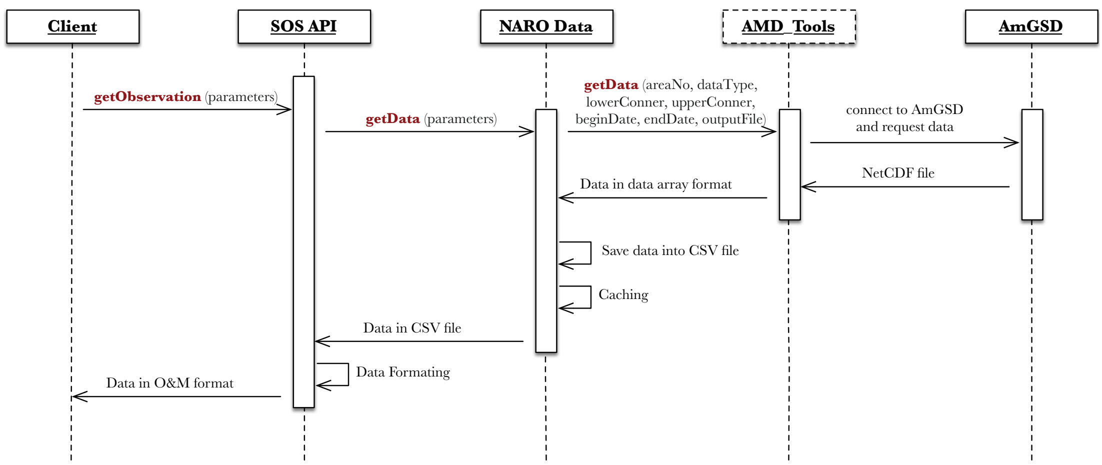


Figure 15. Sequence diagram of retrieving data

A client only issues a getObservation request to SOS API of NARO Agent; then it can receive data in a standard O&M format. The client does not need to know the data structure of AmGSD and functions of AMD_Tools. Request and response are independent of programming language. A client only needs to know the specification of SOS API and service endpoint.

5.2.1 AmGSD accessing mode

AmGSD structures netCDF by each meteorological variables and by each year. If a user requests four variables for 30 years, NARO Data component calls AMD_Tools; then AMD_Tools accesses 120 NetCDF files of AmGSD in a sequence. AMD_Tools combines data into one piece and then returns to NARO Data component. This process takes a long time. To improve response time, we develop three accessing mode controllers for issuing a request to AmGSD as follows.

I. *The Single-process accessing mode*

This mode is to call one getData request of AMD_Tools for a whole period, for example, 30 years from AmGSD. This mode as explained above consumes least memory and CPU but the response is slow.

II. *The multi-process mode*

This mode is to run a requesting process for each element of each year for one request. For example, when a client requests four climate variables for ten years of data then NARO Data component will generate 40 processes. Each process calls getData function of AMD_Tools for retrieving data of one climate variable for one year. After AMD_Tools returns data of each climate variable for each year, NARO Data component combines each year data into one piece, which contains four climate variables of ten years of data and then returns to the client. This mode consumes high memory and CPU but gives the best response time. The maximum number of processing is set to 120 processes to allow users for accessing 30-year time span of four climate items. This number was decided by the memory available on our server which is 1GB. For instance, a user requests 30 years of 5 climate variables, number of process is 150 which exceed the maximum processing. Due to the request exceeds the maximum number of process, the exception message, “The number of properties and year [30 x 5] exceeds the maximum [120]. Reduce either”, is responded. Admin can optimize the number of process regarding to the hardware availability and user requirement.

III. The multithread mode

This mode is similar to Multi-process mode, but generates threads instead of complete processes. If a user request four climate variables for ten years, 40 threads are generated for retrieving the data. The CPU and memory load of a thread is lighter than that of a process, thus we allow users to request all nine variables for whole observation period.

5.2.2 Simple mesh ID conversion

As for each grid, AmGSD internally uses a mesh ID system that follows Japanese public 3rd order mesh ID (National Land Information Division, MILT of Japan, 2016). This Japanese ID system requires rather complicated calculation to encode latitude and longitude or decode mesh ID to latitude and longitude. To simplify encoding and decoding on latitude and longitude to mesh ID, we implement a simple mesh ID equation.

Equation 2

$$simple\ mesh\ ID = X \times 10000 + Y$$

Where:

$simple\ mesh\ ID\ is\ formated\ in\ XXXXYYYY$

$X = \frac{(latitude - lat0)\times 3600}{\Delta lat}$

$Y = \frac{(longitude - lon0)\times 3600}{\Delta lon}$

$latitude\ is\ the\ latitude\ of\ the\ target\ grid\ in\ degree$

$longitude\ is\ the\ longitude\ of\ the\ target\ grid\ in\ degree$

$lat0 = 24;\ lowest\ latitude\ of\ AmGSD\ coverage\ in\ degree$

$lon0 = 122;\ lowest\ longitude\ of\ AmGSD\ coverage\ in\ degree$

$\Delta lat = 30;\ grid\ spacing\ of\ AmGSD\ in\ latitude\ in\ arcsecond$

$\Delta lon = 45;\ grid\ spacing\ of\ AmGSD\ in\ longitude\ in\ arcsecond$

Clients can filter data within the grid of interest by specifying simple mesh ID to retrieve data instead of using a bounding box. We use Uniform Resource Name (URN) concept to refer to AmGSD and each grid in SOS. Concatenation of procedure name and simple mesh ID indicates specific grid for querying data as described in Table 1. For instance, a user wants to retrieve climatological data of farms which are located on around latitude 42.8 and longitude 143.2. By referring to the area in mesh, the location in latitude and longitude is converted to simple mesh ID. In this case, simple mesh ID is 22561696. To filter data within this mesh, the URN is urn:NARO1km:Agro:22561696. And this URN is a filter parameter of offering tag in getObservation request which is described earlier. If clients want to use a bounding box for filtering grids, clients should use a bounding box that has two identical points which is equivalent to a single coordinate.

Table 6. Mapping coverage and grid of AmGSD to SOS

AmGSD	Procedure of SOS	Description
AmGSD; 6 areas	urn:NARO1km:Agro Weather	Referring to whole AmGSD
Each grid is referred by Japanese public 3rd mesh ID	urn:NARO1km:Agro Weather:simple mesh ID	Clients can refer to each grid by concatenating simple mesh ID after procedure value

5.2.3 Meteorological variable mapping

SOS API publishes available observedProperty in SOS response to inform clients of data availability. The names of climate variable of AmGSD are mapped to observedProperty of SOS. These observed Properties are published for securing interoperability of the climate data. observedProperty is also used as filtering parameter in querying data through SOS API.

Climate variables of AmGSD are named internally. To understand the meaning of each name, clients must go through the AmGSD manual. From the viewpoints of clients, the names published need to be comprehensible by human and machines so that data flow can be done automatically and smoothly.

We proposed a naming concept of agro-environmental elements, and Japanese government declared it as a standard guideline which is available on their website

(Prime Minister of Japan and His Cabinet, 2015). The meteorological variables of AmGSD are mapped to observedProperty to be self-descriptive according to the guideline as shown in Table 7. Clients can clearly understand the property of the data without going through specification of AmGSD. These names are used in SOS API for publishing and filtering services. This mapping is important to secure the interoperability at the level of semantics.

Table 7. Mapping Meteorological variable of AmGSD to observedProperty

Meteorological variable of AmGSD	Refer to observedProperty in SOS
TMP_mea	daily_average_air_temperature
TMP_max	daily_maximum_air_Temperature
TMP_min	daily_minimum_air_Temperature
RH	daily_average_relative_humidity
SSD	daily_sunshine_duration
GSR	daily_global_solar_radiant_exposure
DLR	daily_downward_longwave_radiation
APCP	daily_precipitation
WIND	daily_average_wind_speed

5.3 SOS API

To ensure interoperability, NARO Agent provides sensor data and metadata through SOS API conforming to OGC SOS 2.0. Three core operations of SOS, getCapabilities, getObservation and describeSensor, are implemented. The list of APIs and its documents is available on http://133.130.90.193/static/index.html. We applied IP restriction to the API according to NARO requirement for user registration to access data.

As mentioned above, the structure of AmGSD is keeping the data based on area, year and a meteorological variable while SOS is based on observed property in time series structure. To simultaneously provide point time series data in SOS API while maintaining grid information, climate variable of AmGSD is mapped to observedProperty of SOS as mentioned in Table 7. AmGSD is considered as a sensor platform, which performs the observations. AmGSD is named as urn:NARO:AgroWeatherMesh which refers to the whole system and covers all grids. A specific grid can be referred by simple mesh ID as described in Table 6 or by a bounding box that is equivalent to a single coordinate.

We implement SOS API, which supports HTTP GET and POST method. Through the GET method, a user can request data by specifying service endpoint of each service. Through the POST method, a user should send a SOS query in XML for processing each operation. Each operation is described separately as follows:

5.3.1 getCapabilities

Clients can ask for metadata urn:NARO:AgroWeatherMesh by issuing getCapabilities request to service endpoint. Service endpoint of getCapabilities is http://133.130.90.193/api/sos/getcapabilities. An example of XML request is available on http://133.130.90.193/static/sosxmls/GetCapabilities.xml.

Table 8 represents the parameter of getCapabilities. The response is information on available observedProperty, data availability in temporal and spatial dimensions.

Table 8. Request parameter of getCapabilities operation

Parameter		Description
HTTP/GET	HTTP/POST	
sos	N/A	Defining response encoding. true: response is encoded in XML false: response is encoded in JSON

Some part of XML encoded response is presented as follows.

```
<swes:offering>
    <sos:ObservationOffering>
        <swes:description>Agro-Meteorological Grid Square Data System</swes:description>
    <swes:identifier>urn:NARO:AgroWeatherMesh</swes:identifier>
        <swes:name>Chubu University</swes:name>
    <swes:procedure>urn:NARO:AgroWeatherMesh</swes:procedure>
        <swes:observableProperty>daily_average_air_temperature</swes:observableProperty>
        <swes:observableProperty>daily_maximum_air_temperature</swes:observableProperty>
        <swes:observableProperty>daily_minimum_air_temperature</swes:observableProperty>
        <swes:observableProperty>daily_average_relative_humidity</swes:observableProperty>
        <swes:observableProperty>daily_sunshine_duration</swes:observableProperty>
        <swes:observableProperty>daily_global_solar_radiant_exposure</swes:observableProperty>
        <swes:observableProperty>daily_downward_longwave_radiant_exposure</swes:observableProperty>
        <swes:observableProperty>daily_precipitation</swes:observableProperty>
        <swes:observableProperty>daily_average_wind_speed</swes:observableProperty>
        <sos:observedArea>
            <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
                <gml:lowerCorner>24.0 122.0</gml:lowerCorner>
                <gml:upperCorner>46.0 146.0</gml:upperCorner>
            </gml:Envelope>
        </sos:observedArea>
    <sos:phenomenonTime>
            <gml:TimePeriod gml:id="phenomenonTime0">
                <gml:beginPosition>1980-01-01T00:00:00+0900</gml:beginPosition>
                <gml:endPosition>2016-04-24T00:00:00+0900</gml:endPosition>
            </gml:TimePeriod>
        </sos:phenomenonTime>
    </sos:ObservationOffering>
```


The above response means nine observedProperties are available for requesting. Coverage area of the service is within a bounding box 24.0 122.0 and 46.0 146.0 and the data is available from 1980-01-01 until 2016-04-24. Referring to getCapabilities response, clients can filter data on getObservation operation.

5.3.2 getObservation

For retrieving data, clients issue getObservation request to the service endpoint which is published on http://133.130.90.193/api/sos/getobservation. An example of XML request is available on http://133.130.90.193/static/sosxmls/GetObservation.xml.

Clients can filter data based on observedProperty response by temporal and spatial dimensions shown in Table 9. Clients define grid of interest by defining simple mesh ID or a bounding box specified by the lower and upper in the offering parameter.

Table 9. Request parameter of getObservation operation

Parameter		Description
HTTP/GET	HTTP/POST	
offering	offering	Value is mentioned in Table 1 urn:NARO:AgroWeatherMesh[: simple mesh ID]
props	observedProperty	Value is mentioned in Table 2
begin	beginPosition	Specify begining of the observation time period. Date format is "yyyy-MM-ddTHH:mm:ssZ" (T is fixed)
end	endPosition	Specify end of the observation time period.
lower	lowerCorner	Area of interest which can be specific point or within boundary Latitude and Longitude EX: 36.3333 140.4444
upper	upperCorner	Area of interest which can be specific point or within boundary. Lower and upper should be same as we allow only 1 grid to be filtered.
simpleID	Defined with offering	Refer to simple mesh ID equation
cache	N/A	true: accepting cache data. false: request data from AmGSD directly HTTP/POST accepts cache data by default
sos	N/A	true: response is encoded in XML false: response is encoded in JSON
external	N/A	true: request external link to CSV file
complementary	N/A	true: missing data filling is applied false: original data

When clients request long-term data such as 35-year period for the first time, it will take a long time to respond and transfer those data to clients. Once a grid is accessed, data is cashed in NARO agent. The "external" parameter is provided for requesting data as a link to CSV file which contains the data. The CSV file are not directly transferred to clients. Clients receive a response that contains a temporal link that points the CSV data.

The response of HTTP POST method is O&M document by default. The value of the observed property represents in value tag with time sequence.

5.3.3 DescribeSensor

Clients issue a describeSensor request to obtain metadata of the platform. The service endpoint is published on http://133.130.90.193/api/sos/describesensor. The SOS parameter is for defining response encoding either SensorML or JSON. Parameters for requesting metadata from DescribeSensor is available in Table 10.

Table 10. Request parameter of describeSensor operation

Parameter		Description
HTTP/GET	HTTP/POST	
proc	procedure	urn:NARO:AgroWeatherMesh
sos	N/A	Defining response encoding. true: response is encoded in SensorML false: response is encoded in JSON

An example of XML request is available on

http://133.130.90.193/static/sosxmls/DescribeSensor.xml

The response of HTTP POST method is SensorML document by default.

5.4 Mesh Info API

AmGSD provides proportion of land use class of each grid in percentage. There are eleven land use classes as shown in Table 11.

Table 11. Landuse ID in AmGSD system (Ohno, 2014)

Landuse ID	Description
landuse_H210100	Paddy field
landuse_H210200	Other type of farmland

landuse_H210500	Forest
landuse_H210600	Unused land
landuse_H210700	Urban area
landuse_H210901	Road
landuse_H211000	Athletics facilities
landuse_H211100	Rivers and lakes
landuse_H211400	Sea shore
landuse_H211500	Tideland
landuse_H211600	Golf course

This information can be used as filter parameter for agriculture purpose. In order to obtain the land use information, we developed Utilities API. Clients can query data based on parameters mentioned in

Table 12. Mesh Info API supports only HTTP GET method. The Mesh Info API is shown on [Utility] section on

http://133.130.90.193/static/index.html#!/Utilities/get_api_util_mesh.

Table 12. Request parameters of Mesh Info API

Parameter	Example Value	Description
lower	35.3542 139.456	Lower corner of targeting area
upper	35.3542 140.456	Upper corner of targeting area
Land use	landuse_H210100	Land use code or prefecture code can be used for filtering. AmGSD provides those codes on (Ohno, 2014)
ratio	50	Percentage for filtering of the land use. Default is 50 percent
sort	timestamp	Sort key is for giving priority of searching mesh. The list of value is timestamp, frequency and cache
reverse	True or False	This parameter is related to sort value. It is to give a condition of sorting which is descending or ascending. True is descending and false is ascending. False is set as default.

Sort is for accessing frequency and timestamp. The Mesh Info API will filter grid ID within given parameters. Threshold is applied to limit the storage usage so that it will not exceed the plan.

5.5 Caching mechanism

The total Japanese territory is 377,972 sq.km, while NetCDF mesh count is 2,457,600 meshes. The total area of farmland in Japan is 4.5 million hectares or 45,000 sq.km (Ministry of Agriculture, Forestry and Fisheries, 2016). We cannot simply store all data on our cloud platform, as it requires a huge disk space and leads to expensive system maintenance. In order to maximize user satisfaction in response time while minimize maintenance cost, a caching mechanism was developed.

Referring to cloud platform pricing list on https://www.conoha.jp/en/pricing, we focus on a basic plan, which is 1 GB of RAM, 2 cores of CPU and 50 GB of storage. One fifth of storage is used for operation and 40 GB is for caching system.

Each time when NARO Agent responds to SOS getObservation request, the simple mesh ID is recorded, and the data is cached in the system. The grids in the cache system are updated every day because we expect that user may frequently access the same grids. The access timestamp and frequency are recorded whenever clients access them so that we can dynamically optimize the caching system. When the amount of cached data exceeds the threshold, the grids that have seldom been accessed will be removed from the cache. In this way, the agent can respond quickly to ever-accessed grids while limiting the required disk space and the cost.

According to AmGSD land use information of each grid, we utilize this information for initializing most efficient cache by identifying grids to be pre-cached in the system with criteria based on land use status and prefecture.

5.5.1 Pre-caching mechanism

The caching system explained in the previous section improves the response time from the second access. To maximize the response from the first access, we developed a pre-caching mechanism. Primarily, the purpose of this system is to provide climatological data to agriculture applications. Therefore, we expected that most request data will be in farmland. Before starting SOS API service, we developed a pre-caching script. The script issues getObservation request to grids that are expected to receive high frequency of accesses, so that the NARO agent register them into the caching system. The grids to be pre-cached were selected based on land use and prefecture using Mesh Info API. We identified and counted grids that had agricultural land use ratio more than a certain threshold.

As this research is a part of a project that has its study area in Aichi prefecture, , we give priority to the prefecture. Grids within Aichi prefecture which contain paddy field more than 20 percent are pre-cached into the system before clients' requests. We utilize Mesh Info API for retrieving grids which correspond to the given parameters as shown in Table 13.

Table 13. List of Mesh Info API parameters for pre-caching script

Parameter	Value	Description
lower	34.573540 136.671033	Lower corner of Aichi prefecture
upper	35.424788 137.838116	Upper corner of Aichi prefecture
Land use	landuse_H210100	The parameter is landuse_H210100 which is indicated paddy field. And pref_2300 is for Aichi prefecture
ratio	20	The grids which have 20% of farmland in Aichi prefecture are expected.

The curl command for requesting simple mesh ID is

```
curl -X GET --header 'Accept: application/json'
'http://133.130.90.193/api/util/mesh?lower=34.573540%20136.671033&upper=35.424788%20137.838116&landuse=landuse_H210100&ratio=20'
```

An example response of a grid which corresponds with the conditions is presented as follows:


```
{
  "cached": 0,
  "center": "34.5833333333 137.025",
  "date": "",
  "frequency": 0,
  "id": "51377002",
  "landuse": "landuse_H210100",
  "percent": "27.0",
  "simpleId": "12701202",
  "timestamp": 0
}
```


The information shows that the grid, simple mesh ID 12701202, contains 27 percent of paddy field.

The number of the grids that contain more than 20 percent of paddy field in Aichi prefecture is 1471. After receiving the list of grids, getObservation is issued for requesting data of those grids. The sequence of data and reaction of each component is described in Figure 16.

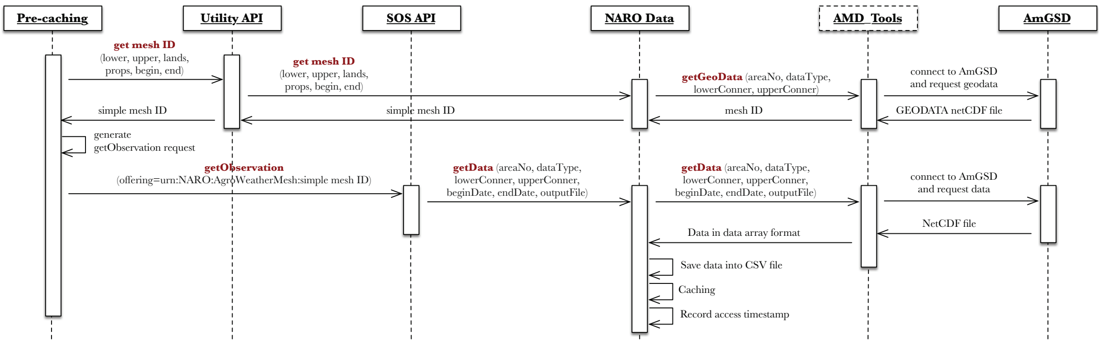


Figure 16. Sequence diagram represents pre-caching process

The pre-caching mechanism ensures the performance response even for the first request in the study area. This mechanism can be expanded to other areas based on disk space and admin's decisions.

5.6 Evaluation of SOS API performance

We evaluated the performance of NARO Agent by setting up realistic accessing scenarios which considered as requirements of agricultural applications. The scenarios were tested to measure the response time, as well as to evaluate the contribution of accelerating mechanism implemented internally.

The characteristics of accesses from actual applications vary. Some applications may require only single climate variable for one year or shorter time period. Other applications require several items for a long term. Some applications need good response time for further processing. For example, generating weather scenario requires long-term climate data of four climate variables (Hanse, et al., 2015). A development of weather generators as web service (Chinnachodteeranun, et al., 2015) used four climate variables of long-term climate data in the range of several ten years for generating weather scenarios in a targeted area. Crop models such as DSSAT (Hoogenboom, et al., 2015) and ORYZA (Bouman, et al., 2009) require daily data of four climate variables, namely maximum temperature, minimum temperature, solar irradiance, and precipitation for simulating crop growth.

Considering the actual requirement of agricultural applications, we define conditions for testing the performance of our mechanism as follows.

Targeted area

The point for retrieving data is longitude = 137.0854 and latitude = 35.1113. It is the location of an experimental farm in the study area.

Meteorological variable

Number of Meteorological variable requested in the scenarios is one and four. For one element, we use daily_average_air_temperature. For four elements, we use daily_minimum_air_temperature, daily_maximum_air_temperature, daily_precipitation and daily_sunshine_duration.

Data range

There are two types of data range, 1 year and 30 years. For 1-year range, the request is from 2015-01-01 to 2015-12-31. For 30-year range, the request is from 1986-01-01 to 2015-12-31.

Accessing mode control

There are three accessing modes for testing, Single-process, Multi-process and Multithread.

Caching status

Two conditions of caching request, yes and no.

Table 14. Result of NARO agent performance test

Unit in second	Single-Process with Caching	Single-Process without Caching	Multi-Process with Caching	Multi-Process without Caching	Multithread with Caching	Multithread without Caching
1-year with 1-element	0.608	1.103	0.795	1.185	1.078	1.264
1-year with 4-elements	0.708	23.413	0.8	1.37	0.813	13.913
30-year with 1-element	1.018	35.513	0.863	3.575	0.744	21.171
30-year with 4-elements	1.534	676.195	1.168	91.882	1.162	639.457

Note: response time was average of 10 times of accessing

The responses of a 1-year range of 1-element in Table 14 show a similarly good result in any accessing mode and caching status condition. As the challenge of this research is to provide long-term data with multi-meteorological variables. we then look at the result of requesting 30-year range with four climate variables. The responses of requesting data without caching on single and multithread mode are very slow. The slow responses were caused by the structure of AmGSD, which stores each climate variable of each year separately. It means that 120 NetCDF files of AmGSD are read and then return to NARO Agent. Multithread mode does not improve the response performance much according to Python runtime architecture. Under Multi-processing mode, the response time reduces to 91.882 seconds, an improvement of 86 percent. The Multi-processing mode is therefore a practical option for this structure. In the viewpoint of a service provider, optimization of response performance and CPU load can be done by setting a maximum number of a process depending on the number of available CPU. In our case, we set maximum processing number to 120 processes as mentioned in AmGSD accessing mode. The Multi-processing is set as default for providing the service.

The response with caching of all scenarios shows similar time, between 0.6 to 1.6 seconds, which is in the acceptable range for users based on response time guideline from the information visualizer (Card, et al., 1991). Response performance is significantly improved by the combination of caching and multi-process implementation.

5.7 Result and Discussion

In this research, a grid climatological data source, AmGSD, was wrapped and exposed through standard web service, SOS API. Connecting AmGSD with SOS API can expand the usability of AmGSD while agricultural applications have more options to receive point-based climatological data throughout Japan from 1980 to present. A Japanese national agricultural application which is developed for demonstrating an impact of climate conditions on cropping yield at farm scale is retrieving AmGSD data through our SOS API. It retrieves 30 years of historical data to visualize the climate trend of a farm. Moreover, the long-term historical data is used to generate weather realizations which then are used as input to a crop model for simulating cropping yield of the farm (detail of generating weather realization from this data source through SOS API are described in Chapter 6). SOS API demonstrates ability to simplify data retrieval by hiding complicated heterogeneous data source and data format behind the SOS API. As a result, agriculture applications can easily query data through SOS API using getObservation request and obtain point-based time series data. Developers do not need to spend time on learning a proprietary structure of data source and data format one by one. Benefit of retrieving data through SOS API is the same mechanism no matter whether it is gridded climate data source or point-based data source. Thus, data fusion of gridded climate and point-based weather data source can be easily implemented in agricultural applications. Data availability in terms of coverage and temporal scale will be broader. It means that agricultural applications will have more data source to utilize. SOS API compliance with the international standard ensures interoperability between gridded climate data and point-based one. Our mechanism contributes to agricultural society by providing standard API for obtaining climate data. The mechanism can be applied to meteorological sources to improve data services and make the development of agricultural applications much more convenient.

The coverage of the data source in this research is only Japan; however, similar mechanisms can be applied to global-scale gridded climate data sources. Accordingly, SOS API is the international standard and globally used in various sectors not only in agriculture but also in environmental monitoring such as terrestrial environmental observatories (Devaraju, et al., 2015) and disaster warning system such as Sensor based Landslide Early Warning System (SLEWS) (Walter & Nash, 2009).

CHAPTER 6: Weather generator API

Climate and weather realizations are essential inputs for simulating crop growth and yields to analyze the risks associated with future conditions. To simplify the procedure of generating weather realizations and make them available over the internet, we implemented novel mechanisms for providing weather generators as web API, as well as a mechanism for sharing identical weather realizations given a climatological information. The weather generator APIs, which are the core components of this chapter, analyze climatological data, and can take seasonal climate forecasts as inputs for generating weather realizations. The generated weather realizations are encoded in ICASA standard format, which are ready for use to crop modeling. All outputs are also generated in SOS standard, which broadens the extent of data sharing and interoperability with other sectoral applications, e.g. water resources management. These services facilitate the development of agricultural application and other applications requiring input weather realizations, as these can be obtained easily by just calling the service. The workload of analysts related to data preparation can be reduced and handling of legacy weather generator programs is eliminated. The architectural design and implementation presented here can be used as a prototype for constructing further services on top of interoperable sensor network systems.

6.1 Design and Implementation

We developed a web service framework for providing weather generator models based on RESTful principle (IBM Knowledge Center, 2016) by utilizing Jersey libraries toolkit (Oracle Corporation, 2016). The architecture consists of several components, as shown in Figure *17*. Each component is wired to one another to provide sufficient and interoperable services.

The main development in this research is Web Service Server part. Details of each part and component are described as follows. The system has been deployed on Amazon Web Service (AWS), a cloud computing service (Amazon, 2016) for scalability and future migration purposes.

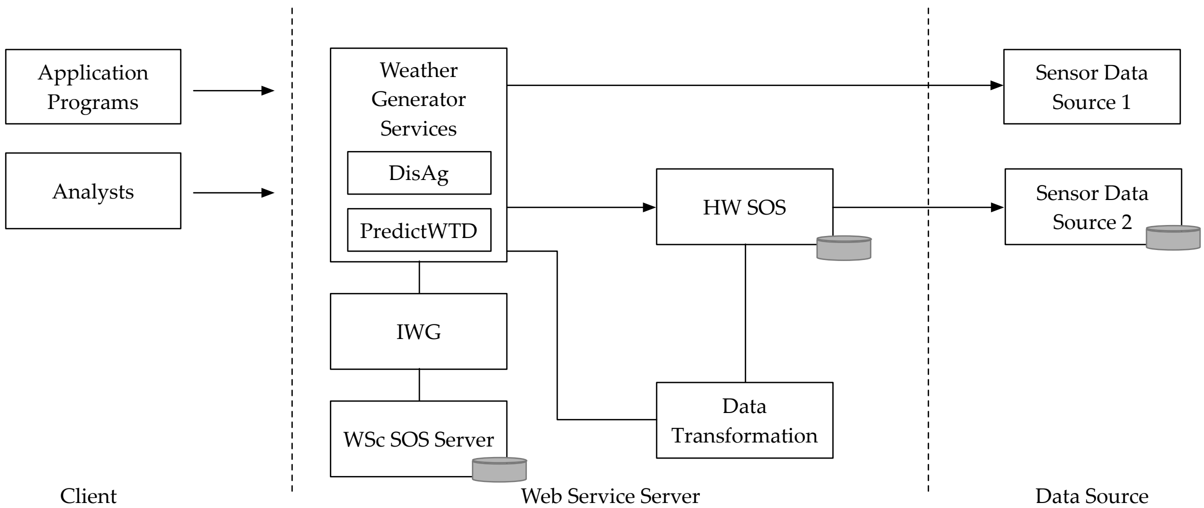


Figure 17*. Overview architecture diagram of weather generator web service*

6.1.1. Client

In this system, clients can be either application programs, such as agricultural web applications and decision support system application, or human analysts. Clients can call directly all service end points provided in Web Service Server. Clients only need to know interfaces for acquiring data. Based on this design and implementation, ordinary clients, developers of weather generating model, weather generation service providers, and source data providers are separated. Clients do not need to manage the weather generation models by themselves.

6.1.2. Data Source

Data source that can be connected to this system must provide climatological data in a web-accessible way. In this research, we focus on data sources that comply with SOS. There are two main data sources in this research which are SOS platform named cloudSense (Listenfield Co.,Ltd., 2016) and AmGSD data source through SOS API (Described in Chapter 5). cloudSense is a sensor cloud service platform that is acquiring sensor data from various sources and providing them through SOS API. The data sources are such as MeteoCrop DB (National Agriculture and Food Research Organization, 2013) that is generating agro-weather data using national weather station network of Japan Meteorological Agency (JMA) (Kuwagata, et al., 2011). Various weather variables are available on cloudSense though we focus on only four variables that are required as input to weather generators.

6.1.3. Web Service Server

This part is a main development in this research. It is a service provider and internal wiring management. There are four inter-connected components in this part. These service components can also be viewed as resources upon which HTTP methods can be invoked.

Historical weather service

Based on 52 North's SOS package (52North, 2016), we developed historical weather service, hereinafter referred to as HW SOS, to provide weather sensor observations via standard SOS operations and data format. HW SOS acquires possibly incomplete climatological data from external sources through cloudSense (Listenfield Co.,Ltd., 2016), then fills missing data so that weather generators can obtain complete datasets. This service provides data for both internal use and external use. Internal use means it supplies long-term climatological data to weather generator for generating weather realization. External use means, external users can directly call this service for receiving historical weather data or climatological data through SOS API. To connect a new climatological data source to the framework, a new HW SOS Server component will be generated.

The mechanism of HW SOS is described in Figure 18. HW SOS has data feeder internally for connecting to and acquiring climatological data from a SOS based data source, cloudSense. HW SOS initiates connection with cloudSense and retrieves long-term climatological data using GetObservation operation. After receiving the data, HW SOS fills missing data and then stores quality controlled climatological data (QC climatological data) into SOS database (SOS DB). HW SOS has internal component for filling missing data (QC climatological data). The service endpoint of HW SOS is publicly available (Chinnachodteeranun, 2016).

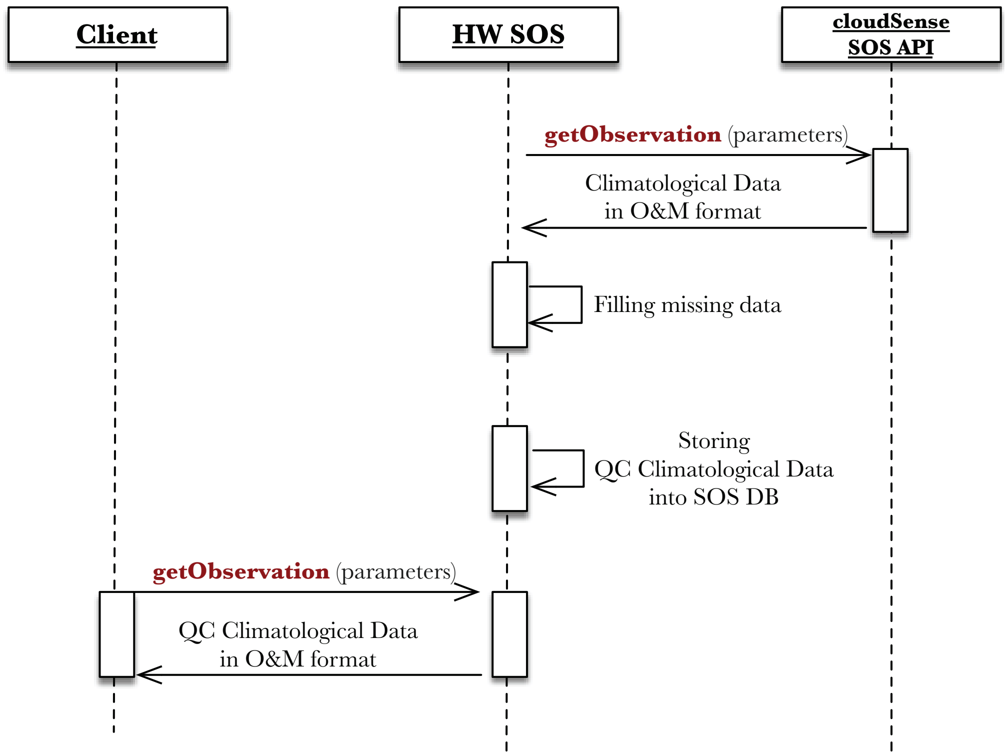


Figure 18. Sequence diagram presents climatological data acquisition and provision of HW SOS

Two main operations of SOS API, GetCapabilities and GetObservation, are described here.

GetCapabilities

Users can ask for available resources of HW SOS by issuing GetCapabilities request to service endpoint. Service endpoint and parameters of GetCapabilities are listed in Table 15.

Table 15. HW SOS and GetCapabilities request

HTTP GET of GetCapabilities	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&request=GetCapabilities&AcceptVersions=2.0.0	
HTTP GET Parameter	**Example Value**	**Description**
service	SOS	To identify the service
request	GetCapabilities	To identify SOS operation
AcceptVersions	2.0.0	To identify the version of SOS API

The response is encoded in XML. The “Value” tag lists available locations for which this service provides climatological data. The important part of the response is visualized as follows.


```xml
<ows:Parameter name="procedure">
<ows:AllowedValues>
<ows:Value>http://www.hondalab.net/weatherhistories/weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281</ows:Value>
<ows:Value>http://www.hondalab.net/weatherhistories/weatherhistory_urn:IBUNYA:AMeDAS-NIAES:KATSUNUMA-49151</ows:Value>
<ows:Value>http://www.hondalab.net/weatherhistories/weatherhistory_urn:IBUNYA:AMeDAS-NIAES:KOFU-49142</ows:Value>
<ows:Value>http://www.hondalab.net/weatherhistories/weatherhistory_urn:IBUNYA:AMeDAS-NIAES:KOSHINO-57051</ows:Value>
<ows:Value>http://www.hondalab.net/weatherhistories/weatherhistory_urn:IBUNYA:AMeDAS-NIAES:MIKUNI-57001</ows:Value>
<ows:Value>http://www.hondalab.net/weatherhistories/weatherhistory_urn:IBUNYA:AMeDAS-NIAES:MINAMICHITA-51311</ows:Value>
<ows:Value>http://www.hondalab.net/weatherhistories/weatherhistory_urn:IBUNYA:AMeDAS-NIAES:NAGOYA-51106</ows:Value>
…
</ows:AllowedValues>
</ows:Parameter>
```


This response also describes available observed property and their time ranges on this HW SOS. The value in each “Value” tag can be used in GetObservation for acquiring climatological data.

GetObservation

For retrieving climatological data, clients issue GetObservation request to the service endpoint and identify location using offering parameter as described in Table 16.

Table 16. HW SOS and GetObservation request

Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?	
HTTP GET Parameter	**Example Value**	**Description**
service	SOS	To identify the service
version	2.0.0	To identify the version of SOS API
request	GetObservation	To identify the SOS operation
MergeObservationIntoDataArray	true	Optional parameter to request result in data array
offering	weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281	Identify location to acquire climatological data. This value can be obtained from GetCapabilities response

The response is O&M document which contains climatological data with four climate variables.

Core Weather Generator Service

This component is to provide weather generator as services to clients. Legacy weather generator programs, DisAg and predictWTD, are wrapped into the web service framework. This component initially wires to HW SOS Servers for retrieving long-term climatological data as an input to the weather generator programs. Each weather generator web service has its own service endpoint for users to invoke. The complicated structure of legacy weather generator models is hidden behind the web service framework. The essential parameters for running the model described in Figure 11 (e.g. param_WTD.txt) are translated into service parameters. Weather realizations as output from these weather generator services are in WTD or WTDE format that are one of the ICASA standard format widely used in agricultural applications and crop modeling.

We exposed two services for DisAg service and predictWTD. Parameters of each services are listed in Table 17 and Table 18.

DisAg weather generator web service (DISAGWS)

This is a web service, which wraps DisAg weather generator program into the framework. The complicated structure of legacy DisAg program is hidden behind. We simplified the service by allowing users to invokes the program using HTTP GET. The service parameters are minimized and described in Table 17.

Table 17. Request parameter of the DisAg service

Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DISAGWS/rest/generate?	
HTTP GET Parameter	**Description**	**Example**
num	Number of weather realization scenario that client wants to generate	100 then 100 scenarios of weather realization are generated.
from	Starting year and month for generating weather realization	2016,1 then weather realization will start from the 1st January 2016
to	Ending year and month for generating weather realization	2017,1 then weather realization will end at 31st January 2017
weatherhistory	Long-term climatological data in WTD format giving in the URI form	Link to the data

An example request for requesting 10 scenarios of weather realization from weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281 is as follows.

HTTP GET Request

http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DISAGWS/rest/generate?num=10&from=2016,1&to=2017,1&weatherhistory=%20http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DataTransformation/rest/transform/http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

Response

A XML document containing links to weather realization in WTD format is returned Information of the model and climatological data that was used to run the model are also returned.

predictWTD weather generator service (PDISAGWS)

This is a web service which wraps predictWTD weather generator model into itself. Wrapping mechanism is as same as DisAg service. predictWTD accepts climate forecast represented by probability of BN (below normal), NN (near normal) and AN (above normal). Users can invoke the service using HTTP GET and parsing parameters as described in Table 18.

Table 18. Request parameter of the predictWTD

Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/PDISAGWS/rest/generate?	
HTTP GET Parameter	**Description**	**Example**
num	Number of weather realization scenario that client wants to generate	If 100 then 100 scenarios of weather realization are generated.
climateforecast	Climate forecast scenario in the format of BNxx, NNyy, ANzz. xx, yy, zz are the probability in percent.	BN33,NN34,AN33
from	Starting year and month for generating weather realization	2016,1 then weather realization will start from the 1st January 2016
to	Ending year and month for generating weather realization	2017,1 then weather realization will end at 31st January 2017
weatherhistory	Long-term climatological data in WTD format giving in the URI form	Link to the data

Example request for requesting 10 scenarios of weather realization from weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281 is as follows.

HTTP GET Request

http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/PDISAGWS/rest/generate?num=10&climateforecast=BN15,NN32,AN53&from=2016,2&to=2017,2&weatherhistory=http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DataTransformation/rest/transform/http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

Response

A XML document which contains the list of URLs to weather realization in WTD format is returned as shown below. The information of the model and climatological data that was used to run the model are also returned.


```
<weatherScenarios>
    <wgmodel>
        <name>DISAGWS</name>
    <post-processing>
        <MonthlyTrendAdjustment>True</MonthlyTrendAdjustment>
    </post-processing>
    </wgmodel>
    <from>2016,30</from>
    <to>2017,30</to>
    <historicalWeather href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/ofubqbkjfumhrsrthkrp/COMM.WTD"/>
    <climateForecast>BN15,NN32,AN53</climateForecast>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0001.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0002.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0003.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0004.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0005.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0006.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0007.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0008.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0009.WTDE"/>
    <scenario href="http://weatherscenarios.s3-website-ap-northeast-1.amazonaws.com/hdotbfgyhsnfehvstmnh/COMM0010.WTDE"/>
</weatherScenarios>
```


Weather realization sharing service

The core weather generation service described in the previous section responds to user's individual requests. The weather realizations generated differ each time even the requesting parameters are the same as it uses a stochastic method. When several applications are serving to multiple users in common regions, they need to use same weather realizations. Thus we implemented a mechanism which multiple users can share the same set of weather realizations. The mechanism provides weather realizations under normal forecast condition as the seasonal forecast scenarios varies according to individual user's needs. In order to give maximum interoperability, the weather realizations are encoded to O&M documents. Weather realization scenarios by SOS (WSc SOS) and Intermediate Weather Generator (IWG) are stacked to serve sharing of weather realizations. WSc SOS is for caching weather realizations and providing them through SOS. Based on specific location of user request, IWG searches for weather realizations that has been archived in WSc SOS Server. If none of weather realization in WSc SOS matches the request, weather generation service will be called for generating the weather realizations. The generated weather realizations will be stored into WSc SOS and returned to the user in O&M documents.

We intend to provide this service to applications not only in agriculture but other application domains. WTD file is one of the standards (ICASA) in the agricultural community, but it is not popular in other application fields. In WTD file, as UOM is not shown, there can easily be errors in the final result by using mismatched UOM. UOMs of these climate variables are known internally that UOM of SRAD must be MJ per m2-day. UOM of TMAX and TMIN is degree Celsius and UOM of RAIN must be mm per day. These rules are not known in application fields except agriculture. With this service, weather realizations are provided through SOS API and the data is in O&M format with UOM, which is convenient to convert to other format used in other type of applications.

The detail of two services are described as following.

I. WSc SOS server for caching

WSc SOS Server archives weather realizations and publish them as SOS. The weather realizations generated by IWG are encoded in O&M document are stored in WSc SOS server. WSc SOS itself is an independent service, thus it is possible for clients to call WSc SOS service as well.

The challenge of sharing weather realization on SOS is a unique characteristic of weather realization data. Weather realizations have multiple values at a same location, same phenomena and same time, i.e. maximum temperature on 2016, DOY=1 may have different values in a different realization. Due to this uniqueness, we cannot simply transform them to comply with SOS standard. In order to standardize weather realizations under SOS and make them available online, we introduced a naming rule for describing each scenario of weather realization in order that we can refer to it.

- Uniform Resource Name (URN) rule for a weather realization scenario

We utilize URN concept for referring to each weather realization. And we use this URN as a filter parameter for offering tag in the getObservation request. The format of the URN of weather realizations is as follows.

scenario_n_generator_G_climateforecast_C_foi_XXX

Where:

scenario is a constant word

n is a scenario number of weather realization

generator is a constant word

G is name of weather generator which is used for generating weather realization.

In this research, DISAG refers to DisAg weather generator and PDISAG refers to PredictWTD weather generator.

climateforecast is constant

C is to refer to seasonal climate forecast scenario.

In this research, we only provide normal forecast which is BN = 33, NN = 34, AN = 33. The keyword to refer is "NORMAL".

foi is the abbreviation for feature of interest. It is used to refer to a specific location, which is a physical weather station name in source SOS.

The URN of the first weather realization generated from historical weather data; foi= urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

is generated by DISAG weather generator as

scenario_1_generator_DISAG_climateforecast_NORMAL_foi_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

IWG for sharing weather realizations

IWG is developed for improving efficiency of weather generator web service and promoting caching and sharing for weather realizations. IWG allows client to request weather realizations by specifying location in latitude and longitude with acceptable distance to a weather station. Service endpoint of IWG and parameters are shown in Table 19.

Table 19. Request parameter of the IWG service

Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/IWG_2/rest/scenarios?	
HTTP GET Parameter	**Description**	**Example**
generator	Name of weather generator service	DISAG
foi	Location in latitude and longitude with distance in unit of degree	lat:35.12lon:136.1within:2.0
refresh	This parameter is for telling the service if user allows caching data to be returned. true: caching data is not accepted. The service must generate new weather realizations false: caching data is accepted	true or false

Example of HTTP GET

http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/IWG_1/rest/scenarios?generator=DISAG&foi=lat:35.12lon:136.1within:2.0&refresh=false

Figure 19 demonstrates the workflow for sharing weather realizations using IWG service and WSc SOS. When a user invokes service of IWG, IWG will first search for cached weather realizations on WSc SOS by issuing GetCapabilities request. If the list of available weather realizations matches user's request, the weather realizations are returned to user. If not, IWG invokes weather generator (WG) service for generating weather realizations and returns them to the user. At the same time, the generated weather realizations are cached to WSc SOS using InsertObservation operation. An operator of this framework may use IWG to prepare weather realizations for all available historical weather data in order to give fast response to clients for a better service in a region.

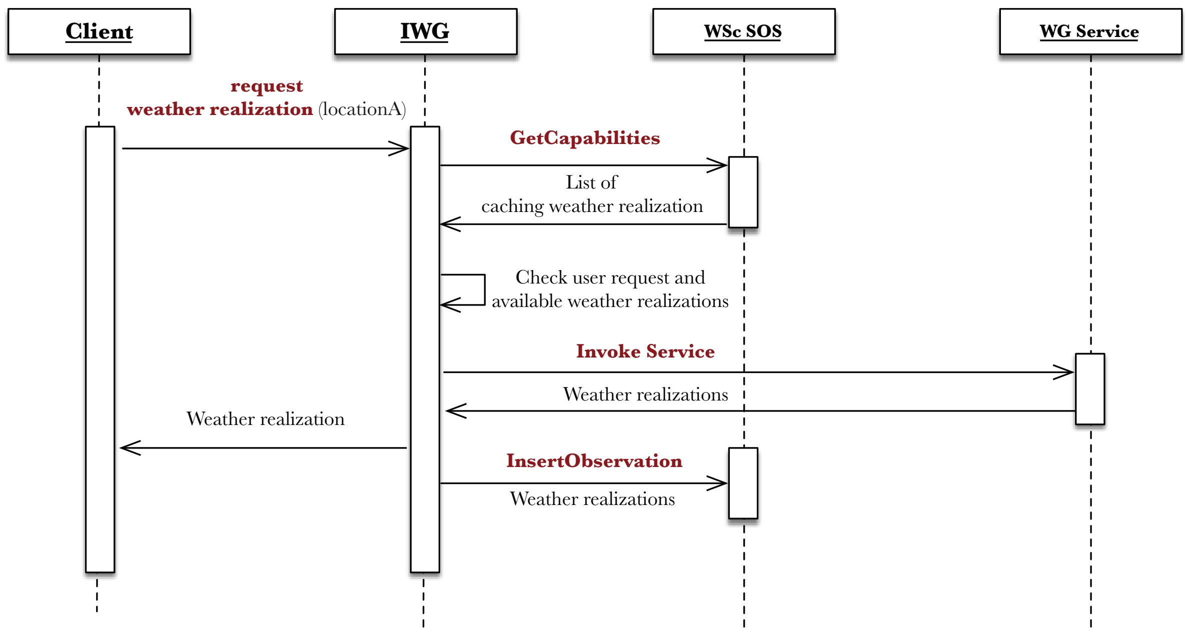


Figure 19. Sequence diagram presents mechanism of IWG and data flow

Weather realizations, which are generated, are stored in WCs SOS. User can invoke the service through SOS API. Service endpoint and parameters are explained in Table 20.

Table 20. Request parameter of the WSc_SOS service

Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/WSc_SOS_2/service?
Operation	**Description**
GetCapabilities	This operation is to request a list of available weather realization which is available on this WCs SOS Server http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/WSc_SOS_2/service?service=SOS&request=GetCapabilities&AcceptVersions=2.0.0
GetObservation	This operation is to request data which is weather realization. The following HTTP GET is for acquiring two scenarios of weather realization.

	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/WSc_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=scenario_1_generator_DISAG_climateforecast_NORMAL_foi_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281,scenario_2_generator_DISAG_climateforecast_NORMAL_foi_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

Data Transformation Service

This service is for data formatting purpose. It is designed for switching data format to match user requirement and model requirement. In this system, transformation of O&M format to WTD is implemented. If an output of one component cannot be directly used as the input of the next component, data transformation service will be called. We designed this service by allowing parsing the link of data to service endpoint using concatenating method. Table 21 provides information and parameters of this service.

Table 21. Data transformation service and parameter

Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DataTransformation/rest/transform/
HTTP GET Parameter	**Description**
URL of historical data	URL of historical data in O&M format

Example request for transforming O&M to WTD is as follows.

http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DataTransformation/rest/transform/http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

6.2 Case study

Several researches utilize DisAg and PredictWTD model for generating weather realizations in order to predict yield with its probability by crop modeling for example (Ines & Hansen, 2006). This section demonstrates two case studies in which weather generator web services are used for preparing weather realizations. The generated weather realizations are in WTD format following ICASA as for the input to DSSAT crop model. The objective of each case study, data flow of receiving data, and service interface are described as follows:

Case study 1: Generating a hundred of weather realizations using DISAG based on thirty-year span of climatological data from urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281. The data source is referred by location name. This station is located at 34.845 degrees' latitude and 137.217 degrees' longitude in Japan. The sample of HTTP GET request and parameters are described in

Table 22.

Table 22. DISAG Web service request

HTTP GET Request	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DISAGWS/rest/generate?num=100&from=2016,3&to=2016,12&weatherhistory=http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DataTransformation/rest/transform/http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281
Parameter	**Value**
Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DISAGWS/rest/generate?
num	100
from	2016,3
to	2016,12
weatherhistory	Data transformation service is called to translate historical data in O&M format to be WTD format. The URL of historical climatological data is http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

One hundred weather realizations are generated. The return is a XML document that contains a list of URL to the weather. The graph of hundred weather realizations is visualized in Figure 20. Each weather realization contains daily maximum temperature, daily minimum temperature, daily solar radiation and daily precipitation from the 1st February to the end of November, which covers one rice season. The graph shows mean, minimum, maximum, 33rd percentile and 67th

percentile of each climate variable. These weather realizations are prepared to be input to simulate expected rice yield of 2016 cropping calendar. In central Japan, rice season is from around April to October.

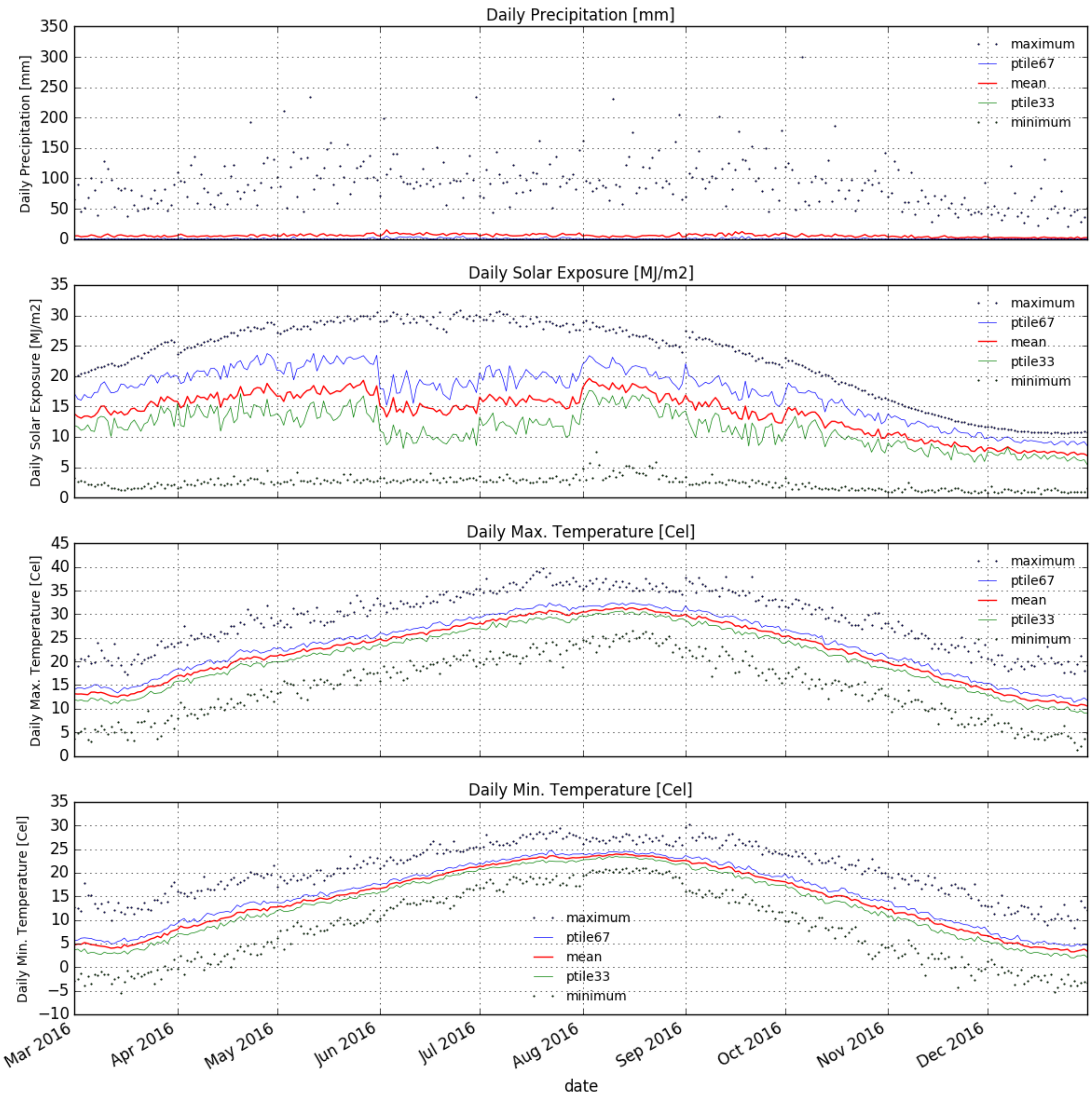


Figure 20. Graph of each climate variable of case study 1 (DISAGWS, normal forecast)

Case study 2: Generating a hundred weather realizations using PDISAG web service by considering seasonal forecast scenarios. HTTP GET request and parameters are in Table 23. The climatological input to the service is same as the case study 1.

Table 23. PDISAG Web service request

HTTP GET Request	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/PDISAGWS/rest/generate?num=100&climateforecast=BN15,NN32,AN53&from=2016,3&to=2016,12&weatherhistory=http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DataTransformation/rest/transform/http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281
Parameter	**Value**
Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/PDISAGWS/rest/generate?
num	100
climateforecast	BN=15,NN=32,AN=53
from	2016,3
to	2016,12
weatherhistory	Data transformation service is called to translate historical data in O&M format to be WTD format. The URL of historical climatological data is http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

We assumed that climate forecast scenarios are BN=15, NN=32 and AN=53. 15%, 32% and 53% probability of precipitation are considered below average, near average and above average respectively. Figure 20 shows a graph of mean, minimum, maximum, 33rd percentile and 67th percentile of each climate variable. The graph shows that mean of rainfall amount is higher than the normal climate as visualized in Figure 21 of case study 1.

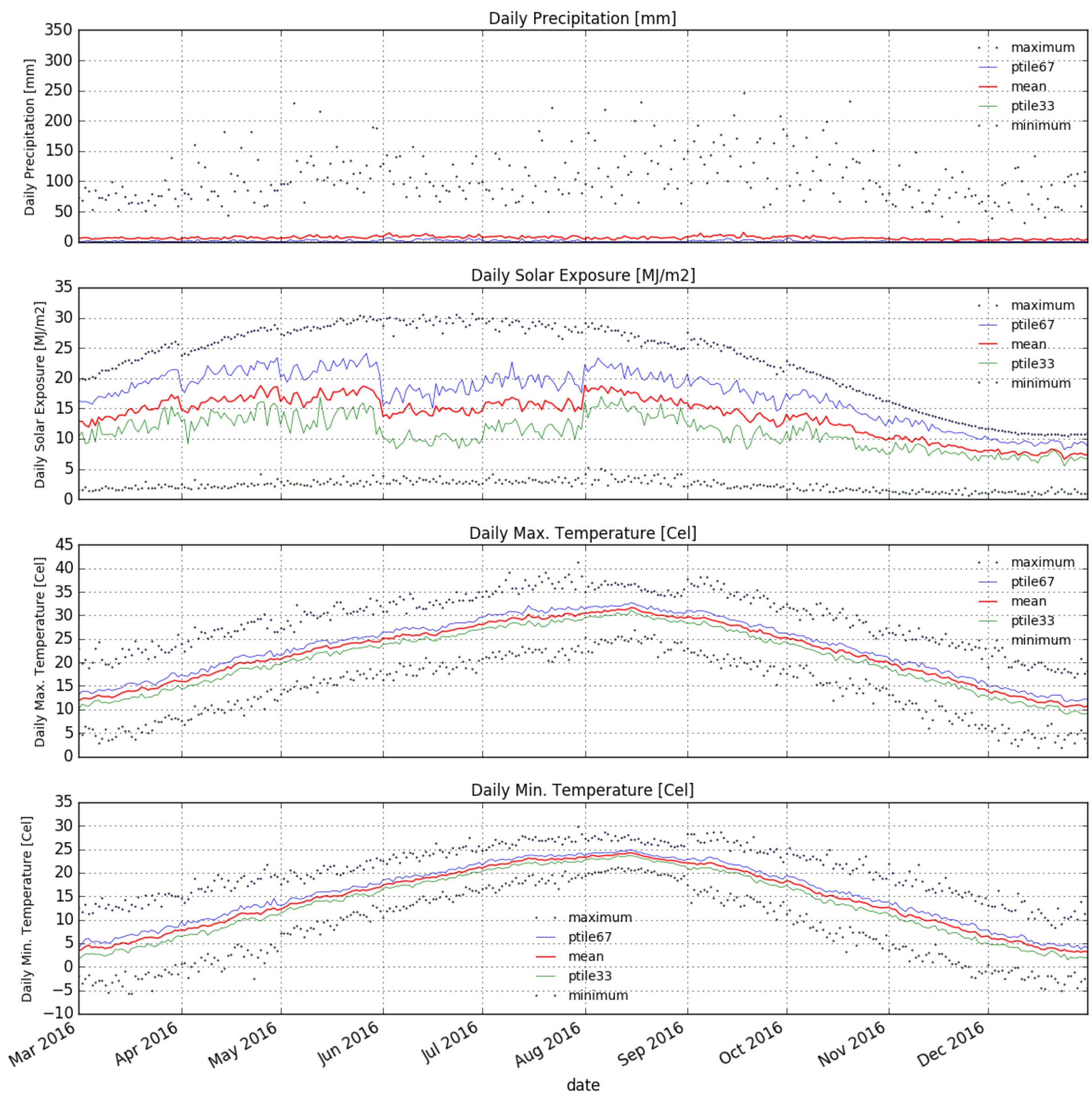


Figure 21. Graph of each climate variable of case study 2 (PDISAGWS, BN=15, NN=32, AN=53)

The average of response time to obtain the URL list of the realizations in both case studies take 46.2 seconds, which is efficient for practical applications. The response time depends on data preparation processes. A larger file of long-term climatological data needs more time to parse to the weather generator services than the smaller one.

6.3 Summary and Discussion

The weather generator web service framework proposed in this research is a novel development. It extends usability of standalone weather generators to be online services. The responses of the services also provide certain level of traceability on data source and mechanism to weather realization. Researchers from a small team without capacity to operate their own weather generator model can utilize these services and focus on their business logic for solving specific problems.

The framework ensures interoperability of the service and data using SOS and O&M documents as well as proposing sharing mechanism for identical weather realizations. Legacy weather generator programs are wrapped into RESTful web service. The complicated structure of weather generator model is hidden behind the framework. Users do not need to handle complicated structure of input file format. Users only need to know the service endpoint and parameters to generate weather realizations. Users' role is clearly separated between model developers, service providers, and general users.

Climatological data source, HW SOS, data transformation service and weather generator model are seamlessly wired. Long-term climatological data, the essential input of weather generators, is passed to the models using concatenation mechanism. It is convenient for users to pass data to the model without downloading and uploading data back and forth across a server and a local machine. The time consumed in manual data preparation can be dramatically reduced by utilizing the data preparation service. Connecting data source and running weather generator can be automated.

Weather generator is important not only in agricultural fields, but many other fields. It can be used to evaluate plans in such as disaster managements or logistics. We designed the framework to secure interoperability of the service and data using SOS and O&M documents. The web service framework proposed in this chapter will expose the function of weather generator models which facilitate the application developments that require weather realization.

CHAPTER 7: Soil API

Soil profile is one of the main required data for running crop models. Since the most of crop models have their own format for soil input and require different soil characteristics information, additional efforts are required to convert the publically available soil dataset to model-specific soil input in a specific format. The DSSAT, for example, requires numbers of soil characteristics in a specific format as shown in Table 4. Global soil databases are often written in ICASA standard, and DSSAT accepts ICASA. Finding a soil profiles for a given location, however, is not a simple work. Furthermore, a national to local scale soil databases that have much more detail than global databases are in their proprietary formats. It is barrier for new comers who want to run crop simulation to understand and be able to prepare a soil profile in required format.

This chapter describes a development of web API framework to facilitate soil profile preparation in ready-to-use format to run DSSAT model. Moreover, to overcome the limitation of soil data in Japan that gives only the data of top soil layer, a mechanism to generate profile data by fusing it with a global database is designed. With this service, users can request soil profile by defining a location in latitude and longitude. Users do not need to spend precious time on learning data structure of soil profile, searching soil profile ID and formatting data.

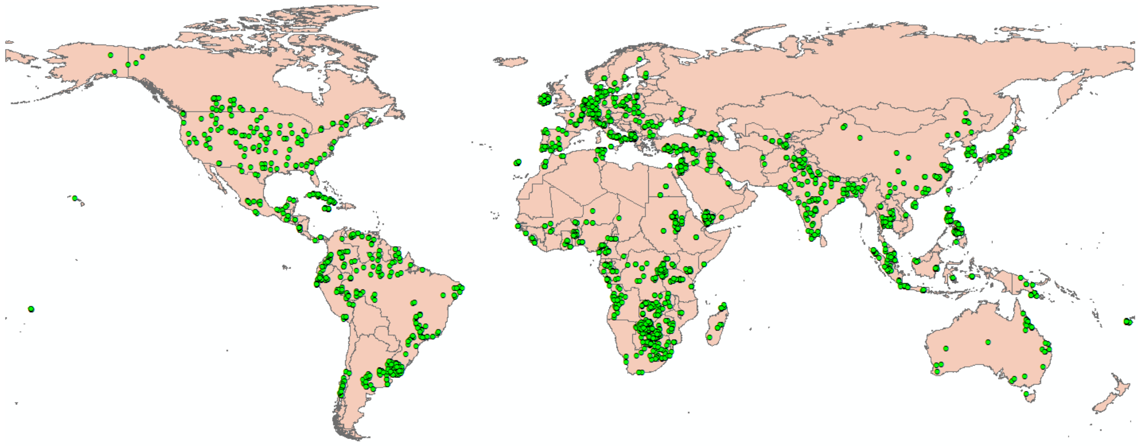
7.1 Data Source

There are three soil data sources in this research. One is a global data source provided by WISE which is described in the section 7.1.1. Another are HC27 generic soil profiles by HarvestChoice (described in the section 7.1.2) and Japanese physicochemical database (described in the section 7.1.3).

7.1.1 WISE dataset

The most readily available soil data throughout the globe is the World Inventory of Soil Emission potentials (WISE) dataset, which has 3404 soil profiles for the world (Arjan, et al., 2007) and (Consuelo & al., 2012) as shown in the Figure 22 (a). In Japan, there are only 29 soil profiles are available from the WISE dataset shown in the Figure 22 (b). The list of available soil profiles with location is shown in Table 24.

(a)



(b)

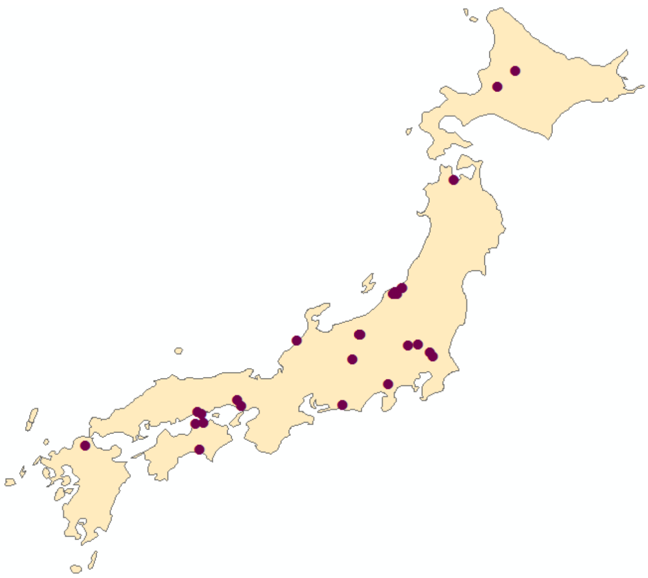


Figure 22. (a) World map showing the location of 3404 soil profiles available from WISE and (b) Japan map showing the location of 29 soil profiles from WISE (note: some locations have more than two different soil profiles)

Table 24. WISE soil profile in Japan

Soil profile	Location
Eutric Fluvisol (FLe)	36.083 140.167
Eutric Gleysol (GLe)	37.800 139.150
Eutric Gleysol (GLe)	36.650 138.167
Dystric Gleysol (GLd)	43.767 142.383
Gleyic Andosol (ANg)	36.000 138.000
Gleyic Cambisol (CMg)	34.767 137.733
Ferric Acrisol (ACf)	34.883 134.895
Eutric Fluvisol (FLe)	34.534 133.920
Eutric Fluvisol (FLe)	34.581 133.831
Eutric Fluvisol (FLe)	34.260 133.776
Dystric Cambisol (CMd)	36.183 140.083
Chromic Cambisol (CMx)	33.550 133.867
Eutric Gleysol (GLe)	37.750 139.183
Dystric Fluvisol (FLd)	43.350 141.883
Gleyic Acrisol (ACg)	34.722 134.990
Eutric Gleysol (GLe)	34.270 133.973
Haplic Podzol (PZh)	36.650 138.200
Cambic Podzol (PZb)	40.833 140.717
Gleyic Podzol (PZg)	36.650 138.167
Humic Cambisol (CMu)	36.667 138.167
Vitric Andosol (ANz)	36.383 139.767

Eutric Gleysol (GLe)	37.917	139.333
Eutric Gleysol (GLe)	37.750	139.083
Eutric Cambisol (CMe)	37.750	139.083
Eutric Gleysol (GLe)	37.917	139.333
Mollic Andosol (ANm)	36.367	139.483
Haplic Andosol (ANh)	35.333	138.967
Eutric Fluvisol (FLe)	33.667	130.800
Eutric Fluvisol (FLe)	36.500	136.500

The structure of soil profile in WISE dataset is in ICASA format which is suitable for DSSAT crop simulation. Figure 23 shows structure of soil profile from WISE dataset. The meaning of each variable in Figure 23 can be found in section 2.6.

```
*WI_ANJP031  WISE        SL      100 WISE DATABASE, SOIL JP031
@SITE        COUNTRY          LAT     LONG SCS Family
 -99         Japan          35.333  138.967 Haplic Andosol (ANh)
@ SCOM  SALB  SLU1  SLDR  SLRO  SLNF  SLPF  SMHB  SMPX  SMKE
    BN  0.13  8.96  0.60 80.00  1.00  1.00 SA001 SA001 SA001
@  SLB  SLMH  SLLL  SDUL  SSAT  SRGF  SSKS  SBDM  SLOC  SLCL  SLSI  SLCF  SLNI  SLHW  SLHB  SCEC  SADC
     5   A11 0.148 0.337 0.621  1.00  3.00  0.76 10.40 12.00 19.00 -99.0  0.70  5.70  5.30 17.40 -99.0
    25   A12 0.128 0.305 0.513  0.74  2.04  1.11  4.60 10.00  7.00 -99.0  0.30  6.00  5.30  9.70 -99.0
    41    AB 0.149 0.337 0.544  0.52  1.51  1.06  4.60 21.00 25.00 -99.0  0.30  6.00  5.40 13.90 -99.0
    66    BC 0.149 0.337 0.556  0.34  1.68  1.02  4.10 21.00 26.00 -99.0  0.30  5.90  5.30 16.20 -99.0
   100   IIC 0.154 0.323 0.498  0.19  1.35  1.17  3.40 19.00 20.00 -99.0  0.20  5.80  5.20 16.30 -99.0
```

Figure 23. Soil profile selected for DSSAT crop simulation in Tokachi region

7.1.2 HarvestChoice dataset

HarvestChoice (HarvestChoice , 2015) by John Dimes (ICRISAT) and Jawoo Koo (HarvestChoice/IFPRI) developed generic soil profile database as called HC27 Generic Soil Profiles by HarvestChoice (HC27) (Koo, 2010) to overcome the limitation of location-specific soil profile data for crop modeling applications. HC27 is a set of generic soil profiles based on three criteria that crop models are most responsive to; texture, fertility (or organic carbon content) and depth (or rooting depth) in formats compatible with DSSAT (Koo, 2010). Soil profile in HC27 is classified as shown in Figure 24.

```
TEXTURE -- FERTILITY - DEPTH --- SOILPROILE
===========================================
+- CLAY -+- HIGH ---+- DEEP ---- HC_GEN0001
|        |          +- MEDIUM -- HC_GEN0002
|        |          +- SHALLOW - HC_GEN0003
|        +- MEDIUM -+- DEEP ---- HC_GEN0004
|        |          +- MEDIUM -- HC_GEN0005
|        |          +- SHALLOW - HC_GEN0006
|        +- LOW ----+- DEEP ---- HC_GEN0007
|                   +- MEDIUM -- HC_GEN0008
|                   +- SHALLOW - HC_GEN0009
+- LOAM -+- HIGH ---+- DEEP ---- HC_GEN0010
|        |          +- MEDIUM -- HC_GEN0011
|        |          +- SHALLOW - HC_GEN0012
|        +- MEDIUM -+- DEEP ---- HC_GEN0013
|        |          +- MEDIUM -- HC_GEN0014
|        |          +- SHALLOW - HC_GEN0015
|        +- LOW ----+- DEEP ---- HC_GEN0016
|                   +- MEDIUM -- HC_GEN0017
|                   +- SHALLOW - HC_GEN0018
+- SAND -+- HIGH ---+- DEEP ---- HC_GEN0019
         |          +- MEDIUM -- HC_GEN0020
         |          +- SHALLOW - HC_GEN0021
         +- MEDIUM -+- DEEP ---- HC_GEN0022
         |          +- MEDIUM -- HC_GEN0023
         |          +- SHALLOW - HC_GEN0024
         +- LOW ----+- DEEP ---- HC_GEN0025
                    +- MEDIUM -- HC_GEN0026
                    +- SHALLOW - HC_GEN0027
```

Figure 24. Soil profile classification in HC27

7.1.3 Japanese topsoil database(J-topsoilDB)

National Agriculture and Food Research Organization (NARO) provides soil physicochemical data of topsoil in Japan (called as J-topsoilDB). Soil samples of topsoil from thousands of farms regarding to farm types were analyzed for chemical and physical characteristics. The physicochemical characteristics of topsoil layer have been published on NIASE webpage (National Institute for Agro-Environmental Sciences, Japan, 2009) and (Soil Map, National Institute for Agro-Environmental Sciences, Japan, 2009). Sixteen major classifications of soil are described (Agriculture, Forestry and Fisheries Research Council Secretariat, 2016). Each major classification has subclasses, i.e. gleysoil (http://agrimesh.dc.affrc.go.jp/soil_db/explain_14.phtml) has 7 subclasses which are fine strong gleysoil (14A), medium-strong gleysoil (14B), gravelly strong gleysoil (14C), fine gleysoil (14D), medium coarse gleysoil (14E), lower-black gleysoil (14F), lower organic matter gleysoil (14G). There are 60 soil classifications of topsoil layer in total. Each soil classification provides textural information as shown in Figure 25 and Figure 26.

Fine strong gley soil (14A)

Gley layer (reduction layer) emerged from within the surface layer under 30cm, the soil of the layer from the surface to 35cm~60cm (below) is sticky to strong sticky (HC, SiC, LiC, SC, SiCL, CL, SCL) it is a gley soil. Consists of alluvial sediments, groundwater level, such as fluvial alluvial plains and the old river bed is high, are distributed to where drainage bad. Hokoeryoku is high, natural fertility is likely to cause a high, but high root system failure is reducing. The land improvement, such as underdrain, occurrence position of gley layer decreases, it is reported that in some cases are changed to fine gley soil or fine gray lowland soil.

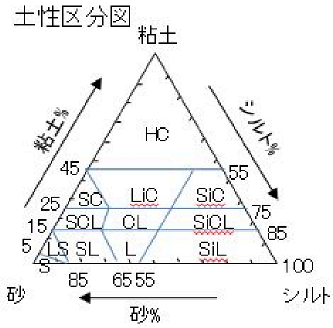


Figure 25. Physical and chemical properties of 14A soil classification (from agrimesh.dc.affrc.go.jp)

分析値一覧(第1層)　14A　細粒強グライ土　土壌機能モニタリング調査　第1巡目:1999〜2003 年

土地利用	項目	01 作土の厚さ (cm)	02 ち密度 (mm)	03 仮比重 (g/m3)	04 固相率 (%)	05 液相率 (%)	06 気相率 (%)	07 保水性 pF 0.0 (%)	08 保水性 pF 1.5 (%)	09 保水性 pF 2.7 (%)	10 有効水分 (%)
水田	平均値	14.6	7.9	0.88	36.6	53.0	10.4		51.7	45.8	9.9
	中央値	14.0	7.0	0.86	35.8	56.2	7.2		55.2	47.9	7.8
	標準偏差	2.8	4.3	0.19	9.0	14.0	9.9		15.3	9.3	7.0
	有効数	245	207	201	181	181	181		177	119	119
普通畑	平均値	.	.	.	.	.	.		.	.	.
	中央値	.	.	.	.	.	.		.	.	.
	標準偏差	.	.	.	.	.	.		.	.	.
	有効数	0	0	0	0	0	0		0	0	0
樹園地	平均値	14.5	14.5	1.00	39.3	52.0	8.9		25.6	7.0	36.0
	中央値	14.5	14.5	1.00	39.3	52.0	8.9		25.6	7.0	36.0
	標準偏差	2.1	0.7	0.18	7.9	9.0	1.1		24.7	.	.
	有効数	2	2	2	2	2	2		2	1	1
牧草畑	平均値	.	.	.	.	.	.		.	.	.
	中央値	.	.	.	.	.	.		.	.	.
	標準偏差	.	.	.	.	.	.		.	.	.
	有効数	0	0	0	0	0	0		0	0	0
全体	平均値	14.9	8.0	0.88	36.6	52.5	10.9		51.1	44.9	10.0
	中央値	14.0	7.0	0.87	35.8	56.2	7.5		54.9	47.0	7.9
	標準偏差	3.5	4.3	0.19	9.0	14.2	10.4		15.6	10.2	7.3
	有効数	253	214	207	187	187	187		183	124	124

土地利用	項目	11 pH (H2O)	12 pH (KCl)	13 置換酸度(y1)	14 電気伝導度 (mS/cm)	15 全窒素 (%)	16 全炭素 (%)	17 CEC (me/100g)	18 交換性 CaO (mg/100g)	19 交換性 MgO (mg/100g)	20 交換性 K2O (mg/100g)	21 塩基飽和度 (%)	22 可給態リン酸 (mg/100g)	23 可給態ケイ酸 (mg/100g)	24 可給態窒素 (mg/100g)
水田	平均値	5.6	4.5	3.85	0.1	0.2	2.6	21.6	280.6	71.7	31.1	67.1	13.7	21.5	17.6
	中央値	5.6	4.3	1.45	0.1	0.2	2.5	20.9	257.7	58.9	23.6	62.9	10.1	12.3	17.2
	標準偏差	0.5	0.6	5.26	0.2	0.1	0.9	7.3	127.5	51.5	29.2	22	12	33.8	9.2
	有効数	267	176	116	181	258	257	259	259	259	259	259	263	248	233
普通畑	平均値	.	.	.	.	.	.	.	.	.	. .	.	.	.	
	中央値	.	.	.	.	.	.	.	.	.	. .	.	.	.	
	標準偏差	.	.	.	.	.	.	.	.	.	. .	.	.	.	
	有効数	0	0	0	0	0	0	0	0	0	0	0	0	0	0
樹園地	平均値	5.3	.	.	0.1	0.2	1.7	24.5	382.9	153.1	84.0	76.5	6.9	13	12.9
	中央値	5.3	.	.	0.1	0.2	1.7	24.5	382.9	153.1	84.0	76.5	6.9	13	12.9
	標準偏差	0.8	.	.	0.0	0.0	0.5	20.4	438.6	169.6	89.1	42.4	3.9	11.7 .	
	有効数	2	0	0	2	2	2	2	2	2	2	2	2	2	1
牧草畑	平均値	.	.	.	.	.	.	.	.	.	. .	.	.	.	
	中央値	.	.	.	.	.	.	.	.	.	. .	.	.	.	
	標準偏差	.	.	.	.	.	.	.	.	.	. .	.	.	.	
	有効数	0	0	0	0	0	0	0	0	0	0	0	0	0	0
全体	平均値	5.6	4.5	3.71	0.1	0.2	2.6	21.5	282.7	72.4	32.8	67.7	16.7	21.5	17.3
	中央値	5.6	4.4	1.30	0.1	0.2	2.5	20.9	261.6	58.8	23.9	64	10.5	12.3	16.5
	標準偏差	0.6	0.6	5.16	0.2	0.1	0.9	7.4	129.2	52.6	31.5	22.2	25	33.5	9.3
	有効数	275	182	122	189	266	265	267	267	267	267	267	271	252	240

農林水産省生産局　土壌保全調査事業成績書　（平成20年８月）を土壌統群別にまとめた。

(from http://agrimesh.dc.affrc.go.jp/soil_db/pac/14A/14A_5.pdf)

Figure 26. Texture information of 14A

7.2 Fusion method to merge J-topsoilDB with HC27

As there are only 29 points of soil profile in WISE database all over Japan, it is not possible to interpolate soil profile from this data set. J-topsoilDB provides many soil classifications in Japan from several thousand sampling points yet, they are only on topsoil layer. HC27 generic soil profiles improved the applicability of WISE by defining 27 soil classes regarding texture, fertility and depth criteria. For running crop simulations, soil profile in topsoil and subsoil layers are required for simulating water movement. To overcome the limitation of location-specific and layer-specific of soil profile data for crop modeling applications, in this research, a fusion method for combining J-topsoilDB and HC27 generic soil profiles are introduced. The fusion method consists of three main steps which are described in 7.2.1, 7.2.2 and 7.2.3 respectively. The fusion soil datasets are generated and kept as newly created JP.SOL. JP.SOL is wrapped into Soil API for providing soil profile in Japan.

7.2.1 Analysis of correlation between J-topsoilDB and HC27

Analysis of correlation between soil classification of J-topsoilDB and soil profile of HC27 on topsoil layer are considered by texture, fertility, and depth criteria. These three criteria are the criteria that HC27 used to define soil profile as shown in Figure 24.

Texture analysis

Firstly, b value (pore pressure coefficient) of J-topsoilDB is used to determine the texture type. b value is a slope of the line obtained from matrix potential between pF 1.5 and pF 2.7. There are three texture classifications, clay, loam and sand as shown in Table 25. The method to decide the threshold is as follow.

- b values were generated for all combinations of clay, loam and sand contents from 0.05 to 0.95 by step 0.05 following (Campbell, 1985).
- Each combination is classified regard to class of major texture.
- All combinations are sorted by b value.
- Identify the b value that appears at boundary of texture class
  - Clay and loam is clearly separated within 5 combinations and the 3rd b value is identified as the threshold.
  - Loam and clay class have rather wider overlaps, 48 classes. The threshold is identified at the center of consecutive 7 combinations where the probability of both class is 50-50%.

Table 25. Determination of Texture

Type of texture	b value threshold
Clay	Equal or greater than 12.63
Loam	Between 5.16 and 12.63
Sand	Less than 5.16

Fertility analysis

To classify fertility, which has high, medium and low class, the soil organic carbon (SOC) is used to determine the fertility classification as shown in Table 26. The defined SOC value is referred to (Jawoo Koo, 2010).

Table 26. Determination of Fertility

Type of fertility	SOC value threshold
High	Equal or greater than 1.2
MED	Between 0.7 and 1.2
LOW	Between 0.0 and 0.7

Depth analysis

The final step is to classify depth criteria in order to define which soil profile of HC27 can be a representative of each soil classification of J-topsoilDB. There are three levels of depth which are deep (D), medium (M) and shallow (S). Depth is classified by available water storage capacity (AWC) as shown in Table 27. Classification formula for defining depth is from (Eunjin Han, 2015).

Table 27. Determination of Depth

Class	AWC (mm/m)	Clay	Loam	Sand
1	Greater than 150	D	D	D
2	Equal or greater than 125	M	M	D
3	Equal or greater than 100	S	M	M
4	Equal or greater than 75	S	S	M
5	Equal or greater than 50	S	S	S
6	Equal or greater than 15	S	S	S
7	Less than 15	S	S	S

After determining three criteria, texture, fertility and depth, we can specify the best match between soil classification of J-topsoilDB and soil profile of HC27 as summarized in Table 28.

Table 28. Matching summary between J-topsoilDB and HC27

Soil classification in J-topsoilDB	Classification			Soil profile in HC27
	Texture	Fertility	Depth	
01A	LOAM	HIGH	SHALLOW	HC_GEN0012
02A	LOAM	MEDIUM	SHALLOW	HC_GEN0015
03A	LOAM	HIGH	DEEP	HC_GEN0010
03B	LOAM	HIGH	MEDIUM	HC_GEN0011
03C	LOAM	HIGH	MEDIUM	HC_GEN0011
03D	LOAM	HIGH	MEDIUM	HC_GEN0011
03E	LOAM	HIGH	SHALLOW	HC_GEN0012
04A	LOAM	HIGH	MEDIUM	HC_GEN0011
04B	LOAM	HIGH	MEDIUM	HC_GEN0011
04C	CLAY	HIGH	SHALLOW	HC_GEN0003
04D	CLAY	HIGH	SHALLOW	HC_GEN0003
04E	LOAM	HIGH	SHALLOW	HC_GEN0012
05A	SAND	HIGH	SHALLOW	HC_GEN0021
05B	LOAM	HIGH	DEEP	HC_GEN0010
05C	LOAM	HIGH	MEDIUM	HC_GEN0011
06A	LOAM	HIGH	SHALLOW	HC_GEN0012
06B	LOAM	HIGH	SHALLOW	HC_GEN0012
06C	LOAM	HIGH	SHALLOW	HC_GEN0012
07A	CLAY	HIGH	SHALLOW	HC_GEN0003
07B	LOAM	HIGH	SHALLOW	HC_GEN0012
07C	CLAY	HIGH	SHALLOW	HC_GEN0003
07D	CLAY	MEDIUM	SHALLOW	HC_GEN0006
08A	SAND	HIGH	SHALLOW	HC_GEN0021
08B	LOAM	HIGH	MEDIUM	HC_GEN0011
08C	CLAY	HIGH	SHALLOW	HC_GEN0003
09A	CLAY	HIGH	SHALLOW	HC_GEN0003
09B	LOAM	HIGH	MEDIUM	HC_GEN0011
09C	LOAM	HIGH	SHALLOW	HC_GEN0012
10A	LOAM	HIGH	SHALLOW	HC_GEN0012
10B	LOAM	HIGH	SHALLOW	HC_GEN0012
10C	LOAM	HIGH	SHALLOW	HC_GEN0012
10D	LOAM	HIGH	MEDIUM	HC_GEN0011
10E	LOAM	HIGH	SHALLOW	HC_GEN0012
10F	CLAY	HIGH	SHALLOW	HC_GEN0003
11A	CLAY	HIGH	SHALLOW	HC_GEN0003
11B	CLAY	HIGH	SHALLOW	HC_GEN0003
12A	CLAY	HIGH	SHALLOW	HC_GEN0003
12B	LOAM	HIGH	MEDIUM	HC_GEN0011
12C	LOAM	HIGH	SHALLOW	HC_GEN0012
12D	CLAY	HIGH	SHALLOW	HC_GEN0003

12E	CLAY	HIGH	SHALLOW	HC_GEN0003
12F	LOAM	HIGH	SHALLOW	HC_GEN0012
13A	CLAY	HIGH	SHALLOW	HC_GEN0003
13B	LOAM	HIGH	MEDIUM	HC_GEN0011
13C	LOAM	HIGH	SHALLOW	HC_GEN0012
13D	LOAM	HIGH	SHALLOW	HC_GEN0012
13E	LOAM	HIGH	SHALLOW	HC_GEN0012
13F	CLAY	HIGH	SHALLOW	HC_GEN0003
13G	LOAM	HIGH	SHALLOW	HC_GEN0012
13H	LOAM	HIGH	MEDIUM	HC_GEN0011
13I	LOAM	HIGH	SHALLOW	HC_GEN0012
14A	LOAM	HIGH	SHALLOW	HC_GEN0012
14B	LOAM	HIGH	MEDIUM	HC_GEN0011
14C	LOAM	HIGH	DEEP	HC_GEN0010
14D	CLAY	HIGH	SHALLOW	HC_GEN0003
14E	LOAM	HIGH	MEDIUM	HC_GEN0011
14F	LOAM	HIGH	MEDIUM	HC_GEN0011
14G	CLAY	HIGH	SHALLOW	HC_GEN0003
15A	CLAY	HIGH	SHALLOW	HC_GEN0003
16A	LOAM	HIGH	DEEP	HC_GEN0010

7.2.2 Updating physicochemical properties required for crop modeling

Dominant variables that govern water movement, soil moisture and potential in simulating soil water content, are listed in Table 29. These variables are derived from J-topsoilDB. The value and methodology for mapping these parameters are described in Table 29. Other variables out of the list in Table 29 are derived directly from HC 27.

Table 29. Overview of how soil characteristic variances are derived from soil datasets

Variable	Definition	Estimation method
SLB	Depth until base of layer (cm)	Thickness of topsoil layer (J-topsoilDB)
SLLL	Lower limit of plant extractable soil water (cm3 cm-3)	θ1500; calculated from line cleated when b value was obtained (J-topsoilDB)
SDUL	Drained upper limit (cm3 cm-3)	θ33; calculated from line cleated when b value was obtained (J-topsoilDB)
SSAT	Saturated upper limit (cm3 cm-3)	θS; ("liquid phase"+ "solid phase")/100
SSKS	Saturated hydraulic conductivity (cm h-1)	1930(θS-θ33) (3-λ) λ: 1/b (J-topsoilDB)
SLOC	Soil organic carbon concentration (%)	Total carbon (J-topsoilDB)
SLNI	Total nitrogen concentration (%)	Total nitrogen
SLHW	pH in buffer (unitless)	pH (J-topsoilDB)

7.2.3 Generating soil profile based on farm type

J-topsoilDB was generated from soil investigation on 4 types of farm which are paddy, general farm, orchard area and pasture. The average of all types is also shown. Paddy field in Japan has special characteristic as it has hardpan horizon layer. Hardpan horizon is a condensed soil layer for keeping water in paddy field. This layer is about 20-25 cm depth beneath the topsoil layer. Landuse and land management have influence on soil characteristics. In order to maintain the special characters of paddy field, four types of soil files, which are paddy (PD), paddy with hardpan horizon (PH), non-paddy (NP) and all (AL), are generated. Then, soil profile ID, as described in Table 30, is defined regarding to landuse.

Table 30. Naming rule for defining soil profile ID

Code	JP_NR	[XX]	[XXX]
Definition	Japan and NARO source	PD: paddy PH: paddy with hardpan horizon NP: non-paddy AL: all	Soil classification code from J-topsoilDB
Example	JP_NRPH07A		

For instance, JP_NRPH07A in JP.SOL which combines a soil profile from HC27 and soil properties at topsoil layer of J-topsoilDB in paddy field with hardpan horizon is shown in Figure 27.

For instance, 07A classification of J-topsoilDB is matched to HC_GEN003 classification of HC27 which is clay texture, high fertility and shallow depth. The value in red underlined of Figure 27 are updated by using the method mentioned in Table 29.

```
*JP_NRPH07A  NARO        C       060 Revised based on HC27 with NARO data, Clay HF060
@SITE        COUNTRY          LAT     LONG SCS Family
 -99         Japan         -00.000  -00.000 Clay Shal (HF)
@ SCOM  SALB  SLU1  SLDR  SLRO  SLNF  SLPF  SMHB  SMPX  SMKE
    BK  0.05  8.00  0.20 85.00  1.00  1.00 SA001 SA001 SA001
@  SLB  SLMH  SLLL  SDUL  SSAT  SRGF  SSKS  SBDM  SLOC  SLCL  SLSI  SLCF  SLNI  SLHW  SLHB  SCEC  SADC
    15     A 0.370 0.447 0.587  1.00  0.59  1.20  2.10 60.00 25.00 -99.0  0.20  5.90 -99.0 -99.0 -99.0
    20    AB 0.230 0.410 0.467  0.65  0.80  1.20  0.87 60.00 25.00 -99.0  0.07  6.50 -99.0 -99.0 -99.0
    25     A 0.370 0.447 0.587  1.00  0.06  1.20  2.10 60.00 25.00 -99.0  0.20  5.90 -99.0 -99.0 -99.0
    30    AB 0.230 0.410 0.467  0.65  0.80  1.20  0.87 60.00 25.00 -99.0  0.07  6.50 -99.0 -99.0 -99.0
    60    BA 0.260 0.415 0.467  0.38  0.60  1.20  0.69 60.00 25.00 -99.0  0.06  6.50 -99.0 -99.0 -99.0
@  SLB  SLPX  SLPT  SLPO CACO3  SLAL  SLFE  SLMN  SLBS  SLPA  SLPB  SLKE  SLMG  SLNA  SLSU  SLEC  SLCA
    15 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
    20 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
    25 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
    30 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
    60 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
```

Figure 27. Soil profile of JP_NRPH07A

7.3 Designing and implementing soil API

Soil API is implemented in PHP programing language version 5.4.16. The API is published on Swagger RESTful API framework (Swagger, 2016). Soil data sources in plaint text files are wrapped in the web service. The algorithm for processing and providing soil profile is visualized in Figure 28.

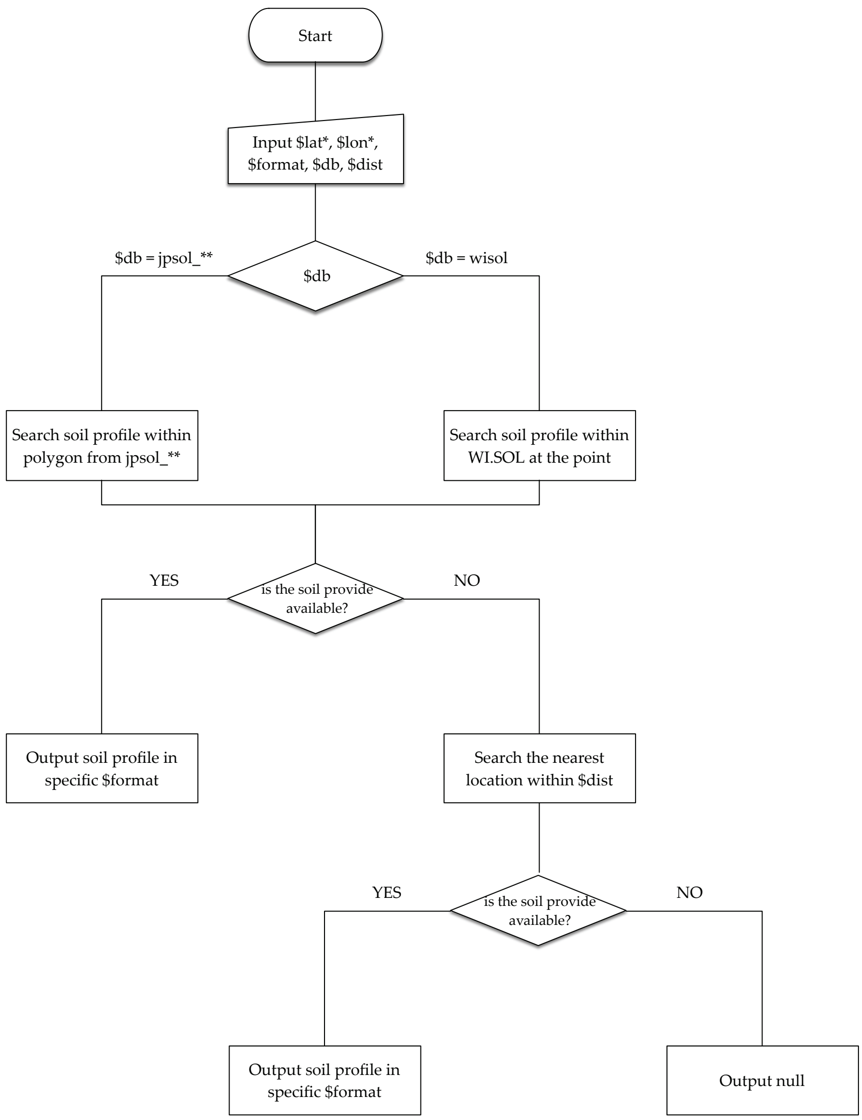


Figure 28. Protocol to obtain soil profile

Table 31. Service endpoint and parameters of SoilAPI

Service endpoint	http://de11.digitalasia.chubu.ac.jp/swagger/#!/project/get_soil_searchSoil_php
HTTP GET Parameter	**Description**
lat	Latitude of interest area
lon	Longitude of interest area
format	Three formats are available which are DSSAT (ICASA format), DSSAT_JSON (ICASA in JSON format) and URL_JSON (the URL to the file). The default is DSSAT
db	Specify the database for retrieving soil profile. Two databases are available; JP.SOL and WI.SOL. Possible parameters are jpsol_al (all), jpsol_pd (paddy field), jpsol_ph (paddy field with hardpan horizon), jpsol_np (non-paddy field), and wisol (WISE database).
dist	Distance in kilometer in case of searching for possible soil profile in the given distance. The default is 0 km.
Example of request URL	http://de11.digitalasia.chubu.ac.jp/soil/searchSoil.php?lat=35.279&lon=137.021&format=dssat&db=niaes2hc

Soil profiles can be obtained by identifying latitude and longitude. Users can choose to receive one of the 3 formats as described in "format" parameter in

Table 31. The response to an example request defined in

Table 31 is shown in Figure 29. Definition of each variable is described in the section 2.6.

Curl

```
curl -X GET --header 'Accept: text/html' 'http://de11.digitalasia.chubu.ac.jp/soil/searchSoil.php?lat=35.279&lon=137.02
```

Request URL

```
http://de11.digitalasia.chubu.ac.jp/soil/searchSoil.php?lat=35.279&lon=137.021&format=dssat&db=niaes2hc
```

Response Body

```
*HC_GEN0011  IFPRI       L       120 HarvestChoice HC27, Loam HF120
@SITE        COUNTRY          LAT     LONG SCS Family
 -99         Generic      -00.000  -00.000 Loam Medm (HF)
@ SCOM  SALB  SLU1  SLDR  SLRO  SLNF  SLPF  SMHB  SMPX  SMKE
    BK  0.10  6.00  0.50 75.00  1.00  1.00 SA001 SA001 SA001
@  SLB  SLMH  SLLL  SDUL  SSAT  SRGF  SSKS  SBDM  SLOC  SLCL  SLSI  SLCF  SLNI  SLHW  SLHB  SCEC  SADC
    10     A 0.170 0.301 0.400  1.00  4.00  1.40  1.40 30.00 45.00 -99.0  0.12  6.50 -99.0 -99.0 -99.0
    30    AB 0.180 0.310 0.410  0.70  1.80  1.40  0.87 30.00 45.00 -99.0  0.07  6.50 -99.0 -99.0 -99.0
    60    BA 0.190 0.310 0.420  0.50  1.60  1.40  0.69 30.00 45.00 -99.0  0.06  6.50 -99.0 -99.0 -99.0
    90     B 0.215 0.315 0.430  0.40  1.50  1.40  0.63 30.00 45.00 -99.0  0.05  6.50 -99.0 -99.0 -99.0
   120   BC1 0.250 0.317 0.440  0.30  1.40  1.40  0.60 30.00 45.00 -99.0  0.05  6.50 -99.0 -99.0 -99.0
@  SLB  SLPX  SLPT  SLPO CACO3  SLAL  SLFE  SLMN  SLBS  SLPA  SLPB  SLKE  SLMG  SLNA  SLSU  SLEC  SLCA
    10 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
    30 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
    60 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
    90 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
   120 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0 -99.0
```

Response Code

```
200
```

Response Headers


```
{
  "server": "Apache/2.4.6 (CentOS) PHP/5.4.16",
  "content-type": "text/html; charset=UTF-8",
  "date": "Sun, 20 Nov 2016 08:35:31 GMT",
  "connection": "Keep-Alive",
  "x-powered-by": "PHP/5.4.16",
  "content-length": "1557",
  "keep-alive": "timeout=5, max=100"
}
```


Figure 29. Example response of Soil API

7.4 Summary and Discussion

Soil information is one of the key inputs to crop simulation, however, preparing them is difficult because the detail and complex parameters are required to be filled in a strict format. Soil profile influences simulation result greatly. The combination of topsoil from J-topsoilDB and subsoil from HC27 are generated for enhancing soil profiles in Japan. This research introduced a fusion mechanism for combining them. The global soil profile (WI.SOL) and enhanced Japanese soil profile (JP.SOL) are provided via Soil API. Soil API simplifies the procedure of soil profile preparation. Users can acquire soil information in the ready-to-use format for inputting to DSSAT crop model online by only knowing latitude and longitude of the target area.

In order to expand the usability of the Soil API to models that requires format other than ICASA, we need to implement data transformation options to other soil profile formats. Data transformation for translating soil information into other format need more investigation.

CHAPTER 8: Crop Model API

This chapter explains crop model API development. The approach of crop model API is to facilitate operation of crop models so that development of agricultural applications can easily utilize crop models to solve problems in a farm, and broader scales. Crop simulation models are useful tools for a simulating impact of an environment to crop production and yield. It is the reason that the demand on integrating crop models into decision support systems has been dramatically increasing. Complexities of manipulating crop models, however, are a barrier to development. Crop model API is developed to make crop models accessible online. Users do not need to operate the model by themselves. Experiment WIKI API is also developed for users to manage conditions of running the model online. With this combination, users can run crop model by just accessing service endpoint while having full control in running conditions.

8.1 Designing

The crop model API was conceptually designed as the top layer of the platform (shown in Figure 12). This part is to control and run crop models using web service principle. The service endpoint is constructed as a combination of a crop model and a crop type. For instance, DSSAT-WHEAT means DSSAT is used to simulate wheat for this service. This design is for supporting further development to add a new model and a new crop into the API. Underneath this service, several services, which are mentioned in previous chapters, are involved for preparing data to the model. When a service endpoint was called, the API invokes the corresponding crop model then return a result to the caller.

Most of crop models require a control file or parameter file for initializing the condition of a simulation. In case of DSSAT crop model, the control file is called an experiment file. It contains essential information and simulation condition for controlling a simulation run (Jones, et al., 2003). Essential but complex information on cultivar, fertilizer, irrigation, crop residue, organic material, soil profile and weather information are structured in the experiment file. Some part, i.e. soil profile and weather information, is referred by ID. When the experimental file is loaded to the main program, several files are referred during the runtime.

To manage the experiment file, we provide a service named "Experiment WIKI API" for generating control file for running DSSAT crop model.

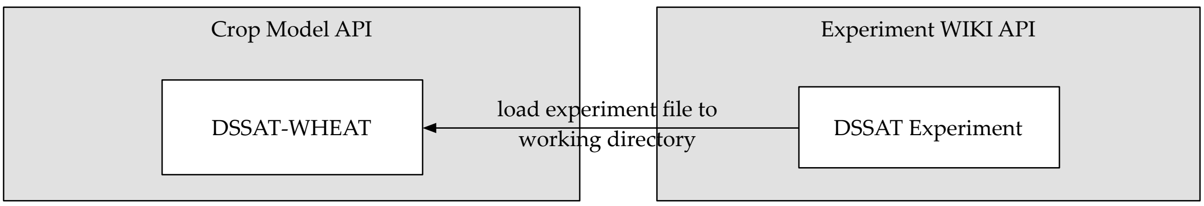


Figure 30. Relationship of Crop Model API and Experiment WIKI API

When a user calls the DSSAT-WHEAT service endpoint, the experiment file, as the required parameter, is loaded to the working directory as visualized in Figure 30.

Model calibration and model validation are not in the scope of this research. Several researches on these topics have been done by crop modelers that prove reliability and performance of DSSAT crop models.

8.2 Implementing

There are two part on this development. One is Crop Model API and another is Experiment WIKI API. They are described as follows.

8.2.1 Crop Model API

The decision support system for agrotechnology transfer (DSSAT) cropping system model (CSM), DSSAT-CSM version 4.6 (Hoogenboom, et al., 2015) is a package that has a function to simulate crop growth and yield for 16 different crops (Jones, et al., 2003). DSSAT-CSM package was wrapped into the web service framework. A service instance in this layer is based on crop model and crop type. For example, DSSAT-WHEAT is an instance which has a service endpoint to simulate wheat production using DSSAT-CSM model.

For this research as a prototype, DSSAT-WHEAT, is deployed on Amazon Web Service (AWS), a cloud computing service (Amazon, 2016).

Table 32 described the service endpoint and parameters required for operating the service.

Table 32. DSSAT-WHEAT service endpoint and parameters

Service endpoint	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DSSATWS/rest/simulate?
HTTP GET Parameter	**Description**
lat	Latitude of interest area
lon	Longitude of interest area
within	Distance in kilometer in case of searching for possible soil profile in the given distance.
experiment	URL of the experimental file
Example of request URL	http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DSSATWS/rest/simulate?lat=10&lon=20&climateforecast=NORMAL&within=1&experiment=https://experiments-dssat.herokuapp.com/experiments/1?format=text

Result of Crop Model API is in XML format as shown in Figure 31. It contains information of crop development and yield. Detail of parameters can be found in files with CDE extension.


```
▼<dssat-run>
  ▼<experimentFile>
     https://experiments-dssat.herokuapp.com/experiments/1?format=text
   </experimentFile>
   <weatherScenarios>links to weather scenarios will be listed here</weatherScenarios>
  ▼<Summary.OUT>
     *SUMMARY : JAPS8102WH RI Hokkaido Wheat Modeling Experiment DSSAT Cropping System Model Ver.
     4.6.0.041 -develop NOV 23, 2016; 06:05:18 !IDENTIFIERS.......................
     TREATMENT............... SITE INFORMATION...........
     DATES...................................... DRY WEIGHT, YIELD AND YIELD
     COMPONENTS..................................
     WATER.........................................
     NITROGEN.................................. PHOSPHORUS........... POTASSIUM............
     ORGANIC MATTER............................. WATER
     PRODUCTIVITY.......................................... NITROGEN
     PRODUCTIVITY.......... SEASONAL ENVIRONMENTAL DATA (Planting to harvest)............. @
     RUNNO TRNO R# O# C# CR MODEL... TNAM..................... FNAM.... WSTA.... SOIL_ID... SDAT
     PDAT EDAT ADAT MDAT HDAT DWAP CWAM HWAM HWAH BWAH PWAM HWUM H#AM H#UM HIAM LAIX IR#M IRCM
     PRCM ETCM EPCM ESCM ROCM DRCM SWXM NI#M NICM NFXM NUCM NLCM NIAM CNAM GNAM PI#M PICM PUPC
     SPAM KI#M KICM KUPC SKAM RECM ONTAM ONAM OPTAM OPAM OCTAM OCAM DMPPM DMPEM DMPTM DMPIM YPPM
     YPEM YPTM YPIM DPNAM DPNUM YPNAM YPNUM NDCH TMAXA TMINA SRADA DAYLA CO2A PRCP ETCP ESCP EPCP
     1 1 1 0 0 WH CSCER046 DRYLAND - 180 KG N/HA SP OBHR0001 MEMU0801 WI_ANJP031 2008100 2008268
     2008274 2009176 2009213 2009213 72 10708 3838 3838 0 4165 0.0440 8722 26.9 0.358 1.9 0 0
     1221 709 167 541 99 416 176 2 120 -99 106 272 110 91 78 -99 -99 -99 -99 -99 -99 -99 -99 0
     29724 29723 0 0 456288 456256 13.7 26.9 64.0 -99.0 4.91 9.63 22.95 -99.00 89.2 100.7 32.0
     36.1 305 10.1 -1.9 11.8 13.00 386.6 782.0 398.3 -99.9 -99.9
   </Summary.OUT>
 </dssat-run>
```


Figure 31. The response of Crop Model API in XML file

8.2.2 Experiment WIKI API

Experiment WIKI API is a service for constructing experiment file or control file of each model. When a new model is connected to the framework, a service instance for generating experiment file for conducting the model will be generated.

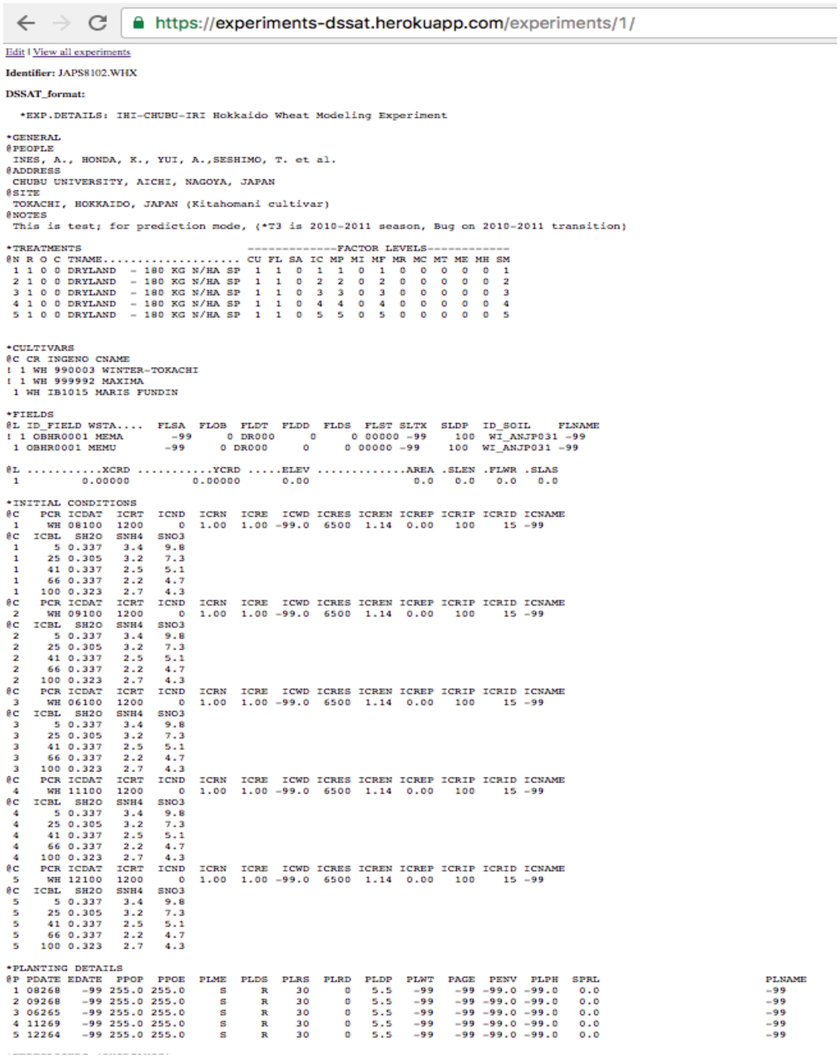
In this research, the experiment WIKI API is developed based on a package named "Translator Development" (AgMIP Software Development, 2016). The package is open source package provided by the agricultural model intercomparison and improvement project (AgMIP). It can be downloaded from https://github.com/agmip/translator-dssat. Translator Development was developed with a concept to describe a process of creating a new data translation component for a crop model (AgMIP Software Development Project, 2016).

The DSSAT experiment service is deployed on HEROKU, a cloud platform based on a managed container system (HEROKU, 2016). This service allows a user to modify the experiment file online. The service endpoint and parameter are described in Table 33.

Table 33. Service endpoint and parameter of DSSAT experiment API

Service endpoint	https://experiments-dssat.herokuapp.com/experiments
HTTP GET Parameter	**Description**
ID	ID of the experiment file
Example of request URL	https://experiments-dssat.herokuapp.com/experiments/1

A user can modify an experiment file by referring it using service endpoint and experiment ID. A sample of experiment file is shown in Figure 32. The URL which is the concatenation of service endpoint and experiment ID can be used as an input to run the crop model.



```
https://experiments-dssat.herokuapp.com/experiments/1/
Edit | View all experiments

Identifier: JAPS8102.WHX

DSSAT_format:

  *EXP.DETAILS: IHI-CHUBU-IRI Hokkaido Wheat Modeling Experiment

*GENERAL
@PEOPLE
 INES, A., HONDA, K., YUI, A.,SESHIMO, T. et al.
@ADDRESS
 CHUBU UNIVERSITY, AICHI, NAGOYA, JAPAN
@SITE
 TOKACHI, HOKKAIDO, JAPAN (Kitahomani cultivar)
@NOTES
 This is test; for prediction mode, (*T3 is 2010-2011 season, Bug on 2010-2011 transition)

*TREATMENTS                        -------------FACTOR LEVELS------------
@N R O C TNAME.................... CU FL SA IC MP MI MF MR MC MT ME MH SM
 1 1 0 0 DRYLAND  - 180 KG N/HA SP  1  1  0  1  1  0  1  0  0  0  0  0  1
 2 1 0 0 DRYLAND  - 180 KG N/HA SP  1  1  0  2  2  0  2  0  0  0  0  0  2
 3 1 0 0 DRYLAND  - 180 KG N/HA SP  1  1  0  3  3  0  3  0  0  0  0  0  3
 4 1 0 0 DRYLAND  - 180 KG N/HA SP  1  1  0  4  4  0  4  0  0  0  0  0  4
 5 1 0 0 DRYLAND  - 180 KG N/HA SP  1  1  0  5  5  0  5  0  0  0  0  0  5


*CULTIVARS
@C CR INGENO CNAME
! 1 WH 990003 WINTER-TOKACHI
! 1 WH 999992 MAXIMA
 1 WH IB1015 MARIS FUNDIN

*FIELDS
@L ID_FIELD WSTA....  FLSA  FLOB  FLDT  FLDD  FLDS  FLST SLTX  SLDP  ID_SOIL    FLNAME
! 1 OBHR0001 MEMA       -99     0 DR000     0     0 00000 -99    100  WI_ANJP031 -99
 1 OBHR0001 MEMU       -99     0 DR000     0     0 00000 -99    100  WI_ANJP031 -99

@L ...........XCRD ...........YCRD .....ELEV .............AREA .SLEN .FLWR .SLAS
 1         0.00000         0.00000      0.00               0.0   0.0   0.0   0.0

*INITIAL CONDITIONS
@C   PCR ICDAT  ICRT  ICND  ICRN  ICRE  ICWD ICRES ICREN ICREP ICRIP ICRID ICNAME
 1    WH 08100  1200     0  1.00  1.00 -99.0  6500  1.14  0.00   100    15 -99
@C  ICBL  SH2O  SNH4  SNO3
 1     5 0.337   3.4   9.8
 1    25 0.305   3.2   7.3
 1    41 0.337   2.5   5.1
 1    66 0.337   2.2   4.7
 1   100 0.323   2.7   4.3
@C   PCR ICDAT  ICRT  ICND  ICRN  ICRE  ICWD ICRES ICREN ICREP ICRIP ICRID ICNAME
 2    WH 09100  1200     0  1.00  1.00 -99.0  6500  1.14  0.00   100    15 -99
@C  ICBL  SH2O  SNH4  SNO3
 2     5 0.337   3.4   9.8
 2    25 0.305   3.2   7.3
 2    41 0.337   2.5   5.1
 2    66 0.337   2.2   4.7
 2   100 0.323   2.7   4.3
@C   PCR ICDAT  ICRT  ICND  ICRN  ICRE  ICWD ICRES ICREN ICREP ICRIP ICRID ICNAME
 3    WH 06100  1200     0  1.00  1.00 -99.0  6500  1.14  0.00   100    15 -99
@C  ICBL  SH2O  SNH4  SNO3
 3     5 0.337   3.4   9.8
 3    25 0.305   3.2   7.3
 3    41 0.337   2.5   5.1
 3    66 0.337   2.2   4.7
 3   100 0.323   2.7   4.3
@C   PCR ICDAT  ICRT  ICND  ICRN  ICRE  ICWD ICRES ICREN ICREP ICRIP ICRID ICNAME
 4    WH 11100  1200     0  1.00  1.00 -99.0  6500  1.14  0.00   100    15 -99
@C  ICBL  SH2O  SNH4  SNO3
 4     5 0.337   3.4   9.8
 4    25 0.305   3.2   7.3
 4    41 0.337   2.5   5.1
 4    66 0.337   2.2   4.7
 4   100 0.323   2.7   4.3
@C   PCR ICDAT  ICRT  ICND  ICRN  ICRE  ICWD ICRES ICREN ICREP ICRIP ICRID ICNAME
 5    WH 12100  1200     0  1.00  1.00 -99.0  6500  1.14  0.00   100    15 -99
@C  ICBL  SH2O  SNH4  SNO3
 5     5 0.337   3.4   9.8
 5    25 0.305   3.2   7.3
 5    41 0.337   2.5   5.1
 5    66 0.337   2.2   4.7
 5   100 0.323   2.7   4.3

*PLANTING DETAILS
@P PDATE EDATE  PPOP  PPOE  PLME  PLDS  PLRS  PLRD  PLDP  PLWT  PAGE  PENV  PLPH  SPRL                        PLNAME
 1 08268   -99 255.0 255.0     S     R    30     0   5.5   -99   -99 -99.0 -99.0   0.0                        -99
 2 09268   -99 255.0 255.0     S     R    30     0   5.5   -99   -99 -99.0 -99.0   0.0                        -99
 3 06265   -99 255.0 255.0     S     R    30     0   5.5   -99   -99 -99.0 -99.0   0.0                        -99
 4 11269   -99 255.0 255.0     S     R    30     0   5.5   -99   -99 -99.0 -99.0   0.0                        -99
 5 12264   -99 255.0 255.0     S     R    30     0   5.5   -99   -99 -99.0 -99.0   0.0                        -99
```

Figure 32. Some part of experiment file from the service

8.3 Summary and Discussion

The Crop Model API development in this chapter exposes a standalone crop model as a web service. A novel mechanism to manipulate crop model and manage experiment file was implemented. With this crop model API, a development of decision support systems or applications that require a crop model as their engine are easier than ever.

Admittedly, in the current development, it is seen as a proof of concept the best. Further development of consistent connection from this service to the lower level of services for obtaining required input data need to be investigated. The output from the crop model API needs more improvement including better provision of other result such as information about daily output such as LAI. A further development can provide result in an XML file with a list of output files from crop model.

Experiment WIKI API provides freedom to edit the experiment file which is convenient for agronomic experts. Further development is required to provide a template for editing only essential parameters.

Output of this API is sanctified for developing decision support system to present expected yield. Comparing running crop model API and standalone DSSAT crop model by users in their local environment, the advantage of the API is evident in terms of data preparation, learning time for operation of the crop model.

CHAPTER 9: Summary and Discussion

As the core of this novel developed platform, APIs for data preparation and running agronomic models were developed for providing data and predictive analytics on climate and crop production as web services. The platform integrated various data sources and agronomic models together in order to facilitate the development of decision support system in agriculture. DSSAT-CSM, a standalone crop model program, version 4.6 was embedded into the crop model API. DSSAT crop models have been recognized as useful tools to simulate impact of environmental factors and management factors to crop productions. They are often used for yield prediction before harvesting. Crop model API enables users to obtain simulation results by issuing HTTP request. Generating an experiment file for controlling conditions of crop models' operation is be performed online by the experiment WIKI API. Two stochastic weather generator programs, DisAg and PredictWTD, were embedded into weather generator APIs for generating weather realizations. These weather generator services are for providing weather realizations which represent possible weather data in a coming cropping season. The PredictWTD API extended the functionality of DisAg API by considering seasonal forecast scenarios in the process of generating weather realizations. Climatological data are among the key inputs for agronomic models. Two main data sources of climatological data, which have a long-term time span over whole Japan, were made available through SOS API as a basic service of the platform, as well as wired to the agronomic models. One is AmGSD, gridded climate data source with 1-km resolution. The other is cloudSense, sensor cloud service platform; that is acquiring sensor data from various sources i.e. weather stations and a national database; MeteoCrop DB. These data sources provided long-term climatological data to weather generator APIs. Another primary input for running crop models is soil profile. Multiple soil data sources fused into Soil API in order to provide soil profiles service in global scale as well as a high-resolution soil profile service in Japan. Three soil data sources, 1) WISE dataset (global soil profile) 2) HC27 (global soil profile by HarvestChoice) and 3) J-topsoilDB (Japan high-resolution soil data at topsoil layer) were integrated into this platform. These fused soil data are provided in standard and ready-to-use format for crop models via Soil API. A fusion mechanism of J-topsoilDB and HC27 was developed in order to extend the national top soil data to profile data. Users can obtain soil profile with hard pan layer, which play an important role especially in paddy fields to retain water.

This agronomic API integration platform can work cooperatively or independently. Each API works independently in the platform. Users can access directly to each API for acquiring data. The APIs also serves internally for preparing input data to other services in the platform.

This research can be considered as a remarkable guideline on taking advantage of standards for providing interoperability on data required for agronomic models. Standardized interfaces absorb complexity for obtaining data of each data source. Users need not to learn proprietary interface of each data source one by one. Standardized encoding formats can reduce workloads on data preparation. SOS standard were complied in the platform to ensure interoperability among climatological data sources as well as observed weather and synthetic weather. Soil API provides soil profile in compliance with ICASA standard.

The platform can be a good foundation for connecting existing standalone models to provide the functionalities in an integrated manner as web services globally. The service oriented model can reduce work loads of ordinary users. Users, as service consumers, need not to prepare servers, install programs, manage format of input, learn operational procedures. Only service providers need to work on constructing standalone models to be service oriented models. Model developers, service providers and service consumers can be clearly separated, thus they can concentrate on their scope of work which will increase their productivity.

The platform was designed for scalability; however, only DSSAT crop model was embedded in this research. Further research can be performed for connecting more agronomic models such as Simulation Model for Rice-Weather relations (SimRiw) (National Agriculture Research Organization, 2016) and Soil Water Atmosphere Plant (SWAP) model (Wageningen University, 2016) to this platform. A mechanism for connecting new modules and models can be developed as an adapter API. The adapter API should work as a registry mechanism so that wiring new models can be done through API without modifying to programming code of the platform. Sooner or later, providing agronomic models as web services will be a standard way. The adapter API will play an important role to provide diverse range of agronomic services. Once a new model has been registered, data preparation service can be called by the models registered in this platform.

As each model requires proprietary input format, data transformation will be big challenge for connecting new models into the platform. A data model for storing data and metadata will be the key for this development. The data model must be able to store essential metadata in order to convert one format to another without losing essential contents. rmAgro data model developed by Prof. Ir. D. Goense with support of FIspace project (FIspace, 2017) can be one of the best solution. The main purpose of rmCrop is a domain reference model for agricultural production (Goense, 2016). It is standardized data for exchanging between applications and systems.

In this research, the purpose of APIs' implementation was to simplify the operation of agronomic models. Standard interface for providing crop model as a web service can be a good further research. Agricultural production system simulator (APSIM) is a foundation that plays an important role on standardized simulation programs on input and output (Agricultural Production Systems Simulator, 2016). The guidelines defined by APSIM can be a good foundation for defining standard interfaces of crop models as web services that can be further implemented on this API integration platform.

Characteristics of farm land and progress of crop growth vary even in a single farm. Adaptation of agriculture 4.0 is taking local context in account for precision treatment which is extremely efficient to agriculture. This platform integrated essential localized data, and analyzing and predictive services which enable decision makers to obtain state of the farm and crop dynamically, then optimize treatments in a local context with very small granule, which is an ultimate precision agriculture.

CHAPTER 10: Conclusion

The research combines software engineering and agronomy to contribute to agricultural field. This agronomic API integration platform connects various data sources and web services together to allow seamless data flow from one service to another for facilitating agronomic models' operation. Observed data and predictive analytics are available in this platform via APIs. Development of useful agricultural applications for better decisions can be easily developed by issuing requests and then receiving data set or model output as response data. The complex internal structure of each data source is hidden behind the APIs. This platform will play a huge role in transforming standalone agronomic models into web services. The web services were built so that standalone agronomic programs can be operated online. The complex structure with hundreds of internal modules of agronomic models is also hidden behind the APIs. The platform can reduce bottlenecks in developing decision support systems as well as in data flow from data sources to models. The API integration platform revolutionizes methodologies to operate the agronomic models as well as data preparation processes. The agronomic API Integration platform can be a good foundation for providing data and models to software development. It provides equal opportunity for small application development teams or small research teams, who do not have the capacity to operate models by themselves. Recently, development of decision support system (DSS) and farm management information systems (FMIS), are accelerating. These systems try to utilize crop models for providing useful information for making a better decision on agronomic management. With this agronomic API integration platform, the development will be simpler than ever. The platform can be a noticeable foundation for expanding applications and services in the agronomic arena. This platform is a novel initiation to support transformation of current IT agriculture into agriculture 4.0 by providing online accessibility to localize data and predictive analytics. Various localized data are integrated and analyzed through agronomic models. Decentralized decision making can be made based on specific localized context for optimizing management and maximizing the productivity.

References

52North, 2016. *SOS bundle source code package.* [Online]

Available at: http://52north.org/downloads/category/3-sos [Accessed 18 Jul 2016].

AgMIP Software Development Project, 2016. *AgMIP software products.* [Online]

Available at: http://research.agmip.org/display/dev/Getting+Started

[Accessed 22 Nov 2016].

AgMIP Software Development, 2016. *Translator Development.* [Online]

Available at: http://research.agmip.org/display/dev/Translator+Development

[Accessed 22 Nov 2016].

Agricultural Production Systems Simulator, 2016. *Agricultural Production Systems Simulator.* [Online]

Available at: https://www.apsim.info/ [Accessed 19 02 2017].

Agriculture, Forestry and Fisheries Research Council Secretariat, 2016. *Soil classes in Japan.* [Online]

Available at: http://agrimesh.dc.affrc.go.jp/soil_db/explain_outline.phtml

[Accessed 5 Dec 2016].

Amazon, 2016. *Amazon web services.* [Online]

Available at: https://aws.amazon.com [Accessed 8 Sep 2016].

APSIM, 2016. *APSIM Initiative.* [Online]

Available at: https://www.apsim.info [Accessed 25 Oct 2016].

Arjan, G., Thornton, P. K. & Hoogenboom, G., 2007. Using the WISE database to parameterize soil inputs for crop simulation models. *Computers and Electronics in Agriculture,* 52(2), pp. 85-100.

Armin, R., 2010. *Flask Framework.* [Online]

Available at: http://flask.pocoo.org [Accessed 2 May 2016].

Barnston, A. G. et al., 2010. Verification of the First 11 Years of IRI's Seasonal Climate Forecasts. 1 March, 2010(49), pp. 493-520.

Batjes, N., 2012. *ISRIC-WISE derived soil properties on a 5 by 5 arc-minutes global grid (ver. 1.2),* Wageningen: World Soil Information.

Bouman, B. et al., 2009. *ORYZA2000: modeling lowland rice.* Version 2.13 ed. s.l.:s.n.

Campbell, G. S., 1985. *Soil Physics with Basic, Transport Models for Soil - Plant System.* First edition ed. Pullman(WA): Elsevier Science.

Card, S. K., Robertson, G. G. & Mackinlay, J. D., 1991. *The information visualizer: An information workspace.* New Orleans, LA, s.n., pp. 181-188.

Chinnachodteeranun, R., 2016. *HW SOS Service Endpoint.* [Online] Available at: http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service [Accessed 8 Sep 2016].

Chinnachodteeranun, R. & Honda, K., 2016. Sensor Observation Service API for Providing Gridded Climate Data to Agricultural Applications. *Future Internet,* 9 Augest, 8(3), p. 40.

Chinnachodteeranun, R. et al., 2015. *Web-Based Wheat Simulation by DSSAT on Sensor Observation Service Standard API.* Hangzhou.

Chinnachodteeranun, R., Hung, N. D. & Honda, K., 2015. *Designing dynamic binding framework for weather generator web service.* Tokyo.

Consuelo, R. & al., e., 2012. Reanalysis of a global soil database for crop and environmental modeling. *Environmental Modeling & Software,* 35(2012), pp. 163-170.

Devaraju, A., Jirka, S., Kunkel, R. & Sorg, J., 2015. Q-SOS—A Sensor Observation Service for Accessing Quality Descriptions of Environmental Data. *ISPRS International Journal of Geo-Information,* 10 August, 2015(4), pp. 1346-1365.

Eunjin Han, A. I. J. K., 2015. *Global High-Resolution Soil Profile Database for Crop Modeling Applications,* Harvest Choice.

FIspace, 2017. *FIspace.* [Online]

Available at: https://www.fispace.eu [Accessed 16 02 2017].

GMO Internet, 2016. *Conoha.* [Online]

Available at: https://www.conoha.jp/en/ [Accessed 6 May 2016].

Goense, I. D., 2016. *rmAgro Guideline: The domain reference model for agricultural production,* s.l.: Report.

Hanse, J. W. & Ines, A. V., 2005. Stochastic disaggregation of monthly rainfall data for crop simulation studies. *Agriculture and Forest Meteorology,* 131(3-4), p. 233–246.

HarvestChoice , 2015. *HarvestChoice.* [Online]

Available at: https://harvestchoice.org/ [Accessed 19 Nov 2016].

Hermann, M., Pentek, T. & Otto, B., 2015. *Technic University Dortmund.* [Online]

Available at: http://www.thiagobranquinho.com/wp-content/uploads/2016/11/Design-Principles-for-Industrie-4_0-Scenarios.pdf [Accessed 27 12 2016].

HEROKU, 2016. *HEROKU platform.* [Online]

Available at: https://www.heroku.com/ [Accessed 22 Nov 2016].

Hiroyuki, O., SASAKI, K., Genj, O. & Kou, N., 2016. Development of grid square air temperature and precipitation data compiled from observed, forecasd, and climatic normal data. *Climate in Bioshere,* Volume 16, pp. 71-79.

Honda, K. & et.al., 2016. *Crop Growth Simulation System for Predicting Yield and Maturity Dates based on Data Assimilation,* Kasugai: Hondalab.

Honda, K. et al., 2014. *FieldTouch: An Innovative Agriculture Decision Support Service Based on Multi-scale Sensor Platform.* San Francisco.

Hoogenboom, G. et al., 2015. *Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.6*

Huang, Y. M., Wu, D. F. & Guo, Q., 2009. *Build a RESTful Web service using Jersey and Apache Tomcat.* [Online]

Available at: http://www.ibm.com/developerworks/library/wa-aj-tomcat/ [Accessed 15 June 2016].

IBM Knowledge Center, 2016. *Defining the resources in RESTful applications.* [Online] Available at: https://www.ibm.com/support/knowledgecenter/was_beta/com.ibm.websphere.base.doc/ae/twbs_jaxrs_defresource.html [Accessed 19 June 2016].

Ines, A. V. & Han, E., 2015. *A Novel Approach for Downscaling Probabilistic Seasonal Climate Forecasts: Parametric or Non-Parametric?.* Minneapolis, American Society of Agronomy, p. 8.

Ines, A. V. & Hansen, J. W., 2006. Bias correction of daily GCM rainfall for crop simulation studies. *Agricultural and Forest Meteorology,* 24 March, 138(2006), pp. 44-53.

Intergovernmental Panel on Climate Change, 2001. *Climate Change 2001.* Geneva, Switzerland, s.n.

IRI, 1997. *International Research Institute for Climate and Society.* [Online] Available at: http://iri.columbia.edu/our-expertise/climate/forecasts/iri-real-time-seasonal-climate-forecasts-and-models/ [Accessed 13 08 2016].

IRI, 2016. *IRI/LDEO Climate Data Library.* [Online] Available at: http://iridl.ldeo.columbia.edu/index.html?Set-Language=en [Accessed 8 Sep 2016].

Japan Meteorological Agency, 2016. *AMeDAS.* [Online] Available at: http://www.jma.go.jp/en/amedas/ [Accessed 11 Nov 2016].

Jawoo Koo, J. D., 2010. HC27: Generic/Prototypical Soil Profiles for grid-based regional-scale crop modeling applications. *Crop Modeling Note 4*, 4 July.

Jones, J. et al., 2003. The DSSAT cropping system model. *European Journal of Agronomy,* 18(2003), pp. 235-265.

Koo, J., 2010. *HC27: Generic/Prototypical Soil Profile.* [Online]
Available at: http://harvestchoice.org/labs/hc27-genericprototypical-soil-profiles
[Accessed 20 Nov 2016].

Kuwagata, T. et al., 2011. MeteoCrop DB: an agro-meteorological database coupled with crop models for studying climate change impacts on rice in Japan. *J. Agric. Meteorol.,* 64(4), pp. 297-306.

Laurenson, M., Kiura, T. & Ninomiya, S., 2001. A Tool for Estimating the Risk of Extreme Climatic Events. *Agricultural Information Research,* 10(2), pp. 79-90.

Laurenson, M. & Ninomiya, S., 2001. Delivering a Simple Maize Maturity and Harvest Data Prediction Model. *Agricultural Information Research,* 10(1), pp. 1-12.

Listenfield Co.,Ltd., 2016. *cloudSense.* [Online]
Available at: http://54.65.217.127/csibunya/ [Accessed 8 Sep 2016].

Ministry of Agriculture, Forestry and Fisheries, 2016. *Statistics On Agricultural Land In Japan.* [Online]
Available at: http://www.maff.go.jp/j/tokei/sihyo/data/10.html [Accessed 21 April 2016].

Min, Y.-M., Kryjov, V. N. & Park, C.-K., 2016. A Probabilistic Multimodel Ensemble Approach to Seasonal Prediction. *American Meteorological Society,* 1 June, 36(4), pp. 1657-1675.

NARO Agric.Rec.Cent., 2016. *Agro-Meteorological Grid Square Data Portal, NARO.* [Online]
Available at: http://mesh.dc.affrc.go.jp/opendap/ [Accessed 23 April 2016].

National Agriculture and Food Research Organization, 2013. *MeteoCrop DB.* [Online]
Available at: http://meteocrop.dc.affrc.go.jp/real/top.php [Accessed 30 Auguest 2016].

National Agriculture Research Organization, 2016. *SIMRIW Model.* [Online]
Available at: http://agrid.diasjp.net/model/simriw/ [Accessed 16 2 2017].

National Institute for Agro-Environmental Sciences, Japan, 2009. *Soil Physicochemical data.* [Online]
Available at: http://agrimesh.dc.affrc.go.jp/soil_db/explain_outline.phtml
[Accessed 25 Nov 2016].

National Land Information Division, MILT of Japan, 2016. *National Land Numerical Information Download Service.* [Online] Available at: http://nlftp.mlit.go.jp/ksj/old/old_data_mesh.html [Accessed 8 May 2016].

National Oceanic and Atmospheric Administration, 2016. *IOOS Sensor Observation Service (SOS).* [Online] Available at: http://sdf.ndbc.noaa.gov/sos/ [Accessed 25 April 2016].

Ninomiya, S. et al., 2007. *Seamless Integration of Sensor Network and Legacy Weather Databases by MetBroker.* IEEE.

Ohno, H., 2014. *User's manual of the Agro-Meteorological Grid Square Data, NARO/ARC,* s.l.: s.n.

Open Geospatial Consortium, 2012. *OGC Sensor Observation Serice Interface Standard.* [Online] Available at: http://www.opengis.net/doc/IS/SOS/2.0 [Accessed 10 June 2016].

Open Geospatial Consortium, 2016. *Observations and Measurements (O&M).* [Online] Available at: http://www.ogcnetwork.net/om [Accessed 27 April 2016].

Open Geospatial Consortium, 2016. *Sensor Model Language (SensorML).* [Online] Available at: http://www.opengeospatial.org/standards/sensorml

Oracle Corporation, 2016. *Jersey - RESTful Web Services in Java.* [Online] Available at: https://jersey.java.net [Accessed 18 June 2016].

Prime Minister of Japan and His Cabinet, 2015. *Guideline for the data element in agricultural IT System.* [Online] Available at: http://www.kantei.go.jp/jp/singi/it2/senmon_bunka/nougyou.html

Prime Minister of Japan and His Cabinet, 2015. *Standard APIs for managing agro-environmental sensor data in agriculture.* [Online] Available at: http://www.kantei.go.jp/jp/singi/it2/senmon_bunka/nougyou.html

Rodriguez, A., 2015. *RESTful Web services: The basics.* [Online] Available at: http://www.ibm.com/developerworks/library/ws-restful/ws-restful-pdf.pdf [Accessed 14 June 2016].

Soil Map, National Institute for Agro-Environmental Sciences, Japan, 2009. *Soil physicochemical data and soil polygon data map.* [Online] Available at: http://agrimesh.dc.affrc.go.jp/soil_db/map_select_figure.phtml [Accessed 25 Nov 2016].

Sun, B., 2009. *A multi-tier architecture for building RESTful Web services.* [Online] Available at: http://www.ibm.com/developerworks/library/wa-aj-multitier/wa-aj-multitier-pdf.pdf [Accessed 11 June 2016].

Swagger, 2016. *Swagger : The World's Most Popular Framework for APIs.* [Online] Available at: http://swagger.io [Accessed 24 Nov 2016].

Teeravech, K., Honda, K., V.M.Ines, A. & Chinnachodteeranun, R., 2013. *Tomorrow's Rice Ver 1: Rice Yield Simulation and Prediction by DSSAT on Web.* Chonburi, s.n.

Timsina, J. & Connor, D., 2000. Productivity and management of rice-wheat cropping system; issues and challenges. *Field Crops Research,* 2 November, 69(2001), pp. 93-132.

Tsuji, G. Y., Hoogenboom, G. & Thornton, P. K., 1998. *Understanding Options for Agricultural Production.* s.l.:SPRINGER-SCIENCE+BUSINESS MEDIA, B.V..

Uehara, G. & Tsuji, G. Y., 1993. The IBSNAT project. In: *Systems Approaches for Agricultural Develpment.* Hawaii: Springer Netherlands, pp. 505-513.

Wageningen University, 2016. *SWAP: Soil Water Atmosphere Plant.* [Online] Available at: http://www.swap.alterra. [Accessed 16 2 2017].

Walter, K. & Nash, E., 2009. *Coupling Wireless Sensor Networks and the Sensor Observation Service – Bridging the Interoperability Gap.* Germany, pp. 1-9.

White, J. W. et al., 2013. Integrated description of agricultural field experiments and production: The ICASA Version 2.0 data standards. *Computers and Electronics in Agriculture,* 1 April, 2013(96), pp. 1-12.

Wilks, D. & Wilby, R., 1999. The weather generation game: a review of stochastic weather models. *Progress in Physical Geography 23,* Volume 3, pp. 329-357.

WMO, 2012. *S2S Prediction Project.* [Online] Available at: https://www.wmo.int/pages/prog/arep/wwrp/new/documents/Andrew_Robertson.pdf [Accessed 13 Sep 2016].