Redevelopment of the DSSAT Model using C++ for Facilitation of Large Scale Data Processing

Kwang Soo Kim
Department of Plant Science and Interdisciplinary Program in Agricultural and Forest Meteorology, Seoul National University, Seoul, Korea
E-mail: luxkwang@snu.ac.kr

Summary: Crop growth models have been designed to predict crop yield at a specific site. Crop models are usually written in Fortran, which could cause low performance in large scale simulation because of inefficient file process and data management. To facilitate large scale simulation, modern computer programming language that supports stream based file input and output processes as well as dynamic data management could be used to implement a crop model. The objective of this study was to rewrite the source code of the CERES-Rice model, which is a rice module of the DSSAT model, using C++. The difference between Fortran and C++ versions of the CERES-Rice was relatively small for five major variables that represented phonological stage, vegetative growth, and energy exchange between crop and atmosphere. Still, there were cases such that simulated flowering and maturity dates differed by two versions of the CERES-Rice model, which would require further evaluation of the C++ version model. Incorporation of structure data types for memory allocation and passing arguments between modules in the C++ version of the CERES-Rice model allowed simpler implementation for data assimilation of remote sensing data. When MODIS LAI product data were forced to the C++ version of the CERES-Rice model, predicted rice yield had greater values of determination coefficients. These results suggested that addition of stream based file process and data management using composite data types would facilitate large scale simulation for crop yield prediction, e.g., spatial assessment of climate change impact on crop yield, which merits further studies to improve current source codes of crop models in terms of high performance computing.
Keywords: DSSAT, compiler compatibility, spatial simulation, high performance computing, spatial prediction

1. INTRODUCTION

Crop growth simulation models have been used to predict the possible impacts of future climate on food security. The use of crop growth simulation could also help identification of potential risk associated with crop production under a given climate conditions, which allow development of adaptation measures under future climate conditions. For example, the Decision Support System for Agrotechnological Transfer (DSSAT) model has been used to assess impact of climate change on productivity of major stable crops including rice, maize, and soybean (Rosenzweig et al. 2014).

Spatial prediction of crop production using crop growth models would help decision making on food production and planning on climate change adaptation for a region of interest. Still, crop growth models that simulate biophysical processes in crop production, e.g., phenology of crop and exchange of energy and matter, tended to predict crop growth at a specific site. Different approaches for spatial simulation of crop growth have been developed using existing models. For example, Elliott et al. (2013) developed pDSSAT for global assessment of climate change impact on crop production. Li et al (2013) used Oryza model to predict rice yield under different environmental conditions using a script.

It would be advantageous to improve existing crop growth model adding functionalities that support large scale data processing for spatial simulation of crop yield. For example, the DSSAT model heavily depends on file input and output process during simulation, which would limit crop growth simulation in a large area at high spatial resolution. Variables in a specific module of crop models are usually accessed using lists of arguments that passed to the given module, which make it challenging to alter the value of a variable in a data assimilation process.

For flexible data exchange between file and modules, modules abased on file streams and data storage mechanism that supports different data types would be useful. C++ supports these functionalities natively, which would make it easy to implement additional functionalities for flexible data management in crop growth simulation. The objective of this study was to redevelop a crop module of the DSSAT model using C++ to improve data exchange between modules of the crop model, which would be helpful for large scale data processing.

2. Design and Implementation

The CERES-Rice Model (version 4.02), which is a rice module of the DSSAT model, was implemented in C++. Subroutines for processing of input parameters, which is written in Fortran, were implemented using classes of file stream and string stream in C++ (Fig. 1). Local variables in each subroutine were grouped into data storage units in the structure data type, which allows management of a group of data items in different types, e.g., integer, floating, or string.

Simulation outputs of the CERES-Rice model written in Fortran and C++ were compared to quantify differences in simulation results between two versions of the model. Outputs from rice growth simulation were analysed for 39 simulations from five experiment settings. The CERES-Rice models implemented in Fortran and C++ were compiled using an open-source compiler within GNU compiler collection under the OpenSUSE (version 11.4), which is a Linux operation system.

In C++ version of the CERES-Rice model, structure data type, which is one of composite data type in C++, was applied to manage state variables for soil, land, weather, crop, management, simulation control, and other input variables. Passing variables defined in structure data type allowed access and modification of the values for state variables defined in subroutines across modules. This design made it possible to update specific variables from the outside of the module where the variable was used.

Because a large number of variables are simulated in the CERES-Rice model, compatibility between two versions of models was evaluated using the variables analyzed in Thorp et al. (2012). These variables include anthesis date (ADAT), maturity date (MDAT), maximum leaf area index (LAIX), canopy weight at maturity (CWAM), cumulative evapotranspiration at maturity (ETCM). ADAT and MDAT were used as indicators for the numerical accuracy of crop phenology. Comparison between values of LAIX and CWAM indicated the accuracy in leaf area development and CWAM between models. ETCM was used as an indicator of numerical accuracy in water balance simulation. Differences in variables between models were quantified using mean difference (MD) and mean absolute difference (MAD) between models built using different compilers as follows:

$MD = (Var_m - Var_n) / N$ and (eq. 1)
$MAD = | Var_m - Var_n| / N$, (eq. 2)

where Var and N indicate simulation outputs of a variable and total number of treatment in simulations, respectively. m and n represent individual compilers used to compile the CERES-Rice model. t-value was also calculated to provide a simple indication of degree of agreement between simulation outputs.

Using C++ version of CERES-Rice model, MODIS product (MCD15A2 MODIS/Terra+Aqua Leaf Area Index/FPAR 8-Day L4 Global 1km) was assimilated to improve rice yield at Haenam, Korea. Weather data were obtained through Monsoon Asian Hydro-Atmosphere Scientific Research and Prediction Initiative. In our study, Odae rice cultivar was used. Its parameters were obtained from Shim et al. (2003) was used.

3. Results and Discussion

The C++ version of CERES-Rice model had similar simulation outputs to the Fortran version of the model (Table 1). However, simulated phenology dates, e.g., ADAT and MDAT, were different in four simulation settings. Thus, analysis of simulation outputs were performed for two groups depending on whether phonological date agreed between models built using C++ and Fortran or not.

Under experimental conditions where phenological dates were identical between Fortran and C++ version of the CERES-Rice models, simulation outputs for CWAM had smaller MAD between Fortran and C++ versions of the model. On the other hand, MAD of the LAIX output was less between Fortran and C++ versions of model.

In the case of experiment configuration where phenological dates were different between Fortran and C++ versions of CERES-Rice model, C++ version of the model had slower phenological changes than Fortran version of the model. Between Fortran version of the models, LAIX was considerably low and CWAM was identical between two models. Considering the fact that phenological dates differ by experimental conditions between C++ version and Fortran version of the CERES-Rice model, there was discrepancy between C++ code and Fortran code, which require further examination of source code for C++ version model.

When the values of LAI from MODIS product were used to adjust biomass of rice plant during simulation, the prediction error decreased (Fig. 2). For example, estimates of rice yield using data assimilation had greater value of R^2 (0.87) than those without data assimilation (0.65).

Our results suggested that it would be preferable to use the C++ version of the CERES-Rice model because it allowed flexible data management, which would facilitate data assimilation. Still, more validation and debug would needed for the C++ version of the CERES-Rice model because there was difference in simulated phenological dates and key variables in the model.

ACKNOWELDGEMENTS

This work was supported by the Rural Development Administration (ATIS number PJ010115022015 and PJ010549022015).

REFERENCES

Thorp, K.R., J.W. White, C.H. Porter, G. Hoogenboom, G.S. Nearing, and A.N. French. 2012. Methodology to evaluate the performance of simulation models for alternative compiler and operating system configurations. *Computers and Electronics in Agriculture* 81:62-71.

Shim, K-.M., J.-T. Lee, Y.-S. Lee, and G-Y. Kim. 2003. Traits of agro-meteorological disasters in 20th century Korea. *Korean Journal of Agricultural and Forest Meteorology* 5:255-260.

Rosenzweig, C., J. Elliott, D. Deryng, A.C. Ruane, C. Müller, A. Arneth, K.J. Boote, C. Folberth, M. Glotter, N. Khabarov, K. Neumann, F. Piontek, T.A.M. Pugh, E. Schmid, E. Stehfest, H. Yang, and J.W. Jones. 2014. Assessing agricultural risks of climate change in the 21st century in a global gridded crop model intercomparison. *Proceedings of the National Academy of Sciences* 111:3268–3273.

Elliott, J., M. Glotter, N. Best, D. Kelly, M. Wilde, and I. Foster. 2013. The parallel system for integrating impact models and sectors (pSIMS). Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery (XSEDE ’13), Association for Computing Machinery, 21:1–8.

Li, T., A.K. Raman, M. Marcaida, A. Kumar, O. Angeles, and A.M. Radanielson. 2013. Simulation of genotype performances across a larger number of environments for rice breeding using ORYZA2000. *Field Crops Research* 149:312–321.

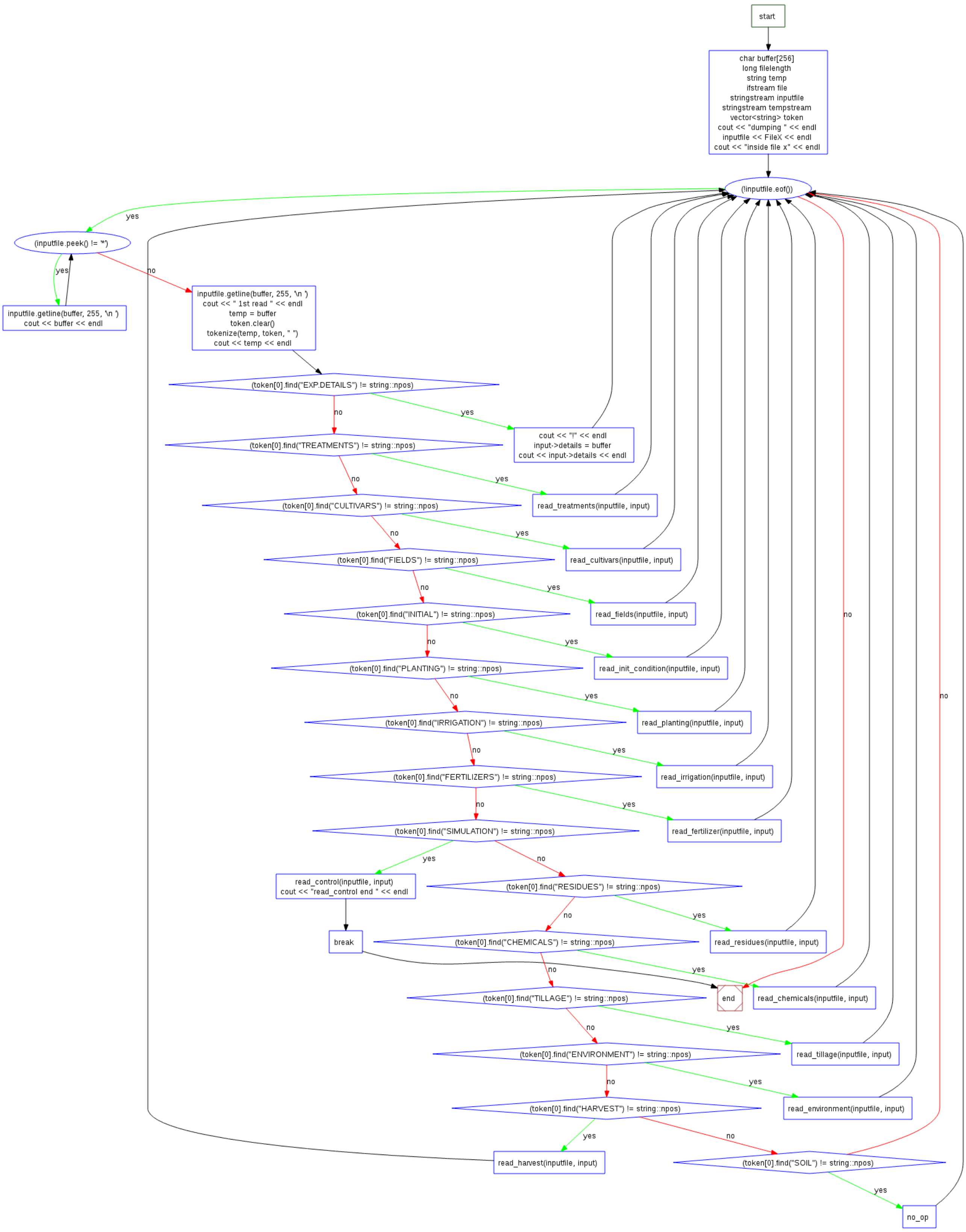

Figure 1. Flow chart of file input module using file and string stream in the C++ version of CERES-Rice model.

Table 1. Mean absolute differences between CERES-Rice models compiled using C++ and Fortran open-source compilers.

g++/ gofrtran[a]	Same[b] (N=35)	Different[b] (N=4)
ADAT	0	2.5E+00
MDAT	0	4.0E+00
LAIX	4.0E-04	1.6E-01
CWAM	2.7E+00	4.5E+02
ETCM	0	2.1E+02

a. g++ and gfortran represent open-source GNU C++ compiler and fortran compiler, respectively.
b. “Same” and “Different” represents experiment configurations in which phenological date was same and different between simulation using the model based on C++ and Fortran codes, respectively.

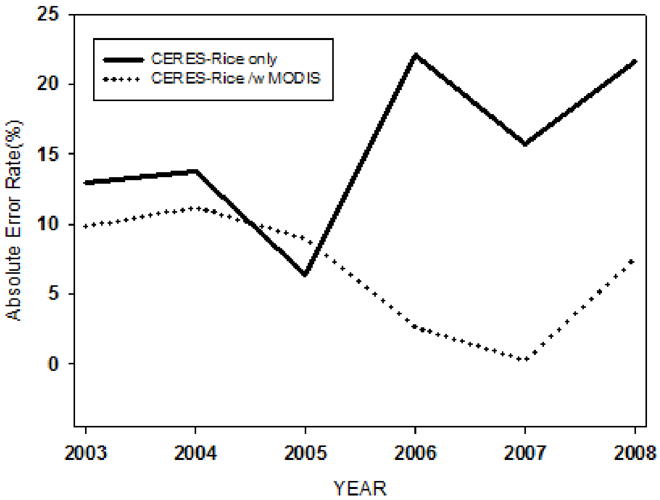

Figure 2. Absolute error rate between observed and simulated rice yield at Haenam using the CERES-Rice model and MODIS LAI product.