Corrigendum: The Design and Construction of an Ecosystem Simulation Workbench
(2019 IOP Conf. Ser.: Earth Environ. Sci. 515 012015)

R Molenaar¹, F Pangkerego¹ and H Wullur¹
¹Lecturer at the Department of Agricultural Technology, the Sam Ratulangi University, Manado, Indonesia.

The authors of the paper mistakenly included templated text (“start your abstract here”) within the abstract of the article, as a result of copying text from a template provided by IOP Publishing to conference proceedings authors that helps ensure their work is formatted in the correct way. The authors apologise for this mistake.
The Design and Construction of an Ecosystem Simulation Workbench

R Molenaar¹, F Pangkerego¹, and H Wullur¹

¹ Lecturer at the Department of Agricultural Technology, the Sam Ratulangi University, Manado, Indonesia.

Corresponding author’s email address: robertmolenaar@unsrat.ac.id

Abstract. Efforts to study ecosystems have been carried out by humans since ancient times in various ways or approaches. The approach of using a computer model and simulation nowadays is one of the increasingly possible alternatives, especially with the rapid advancement of information and communication technology. There are several ways to build computer models of an ecosystem to be used in simulations. Composite modeling is one method that can be used where several computer programs (main, sub-routines, functions, sub-programs) written using a procedural programming language, which as a whole are integrated into one final program that can be run on a computer. The approach in this research is different from what was explained earlier. Composite models are still used but each component of the model is kept separate which is not combined into one module when the simulation process of the ecosystem model is formed. Each module remains separate but will operate side by side and communicate with one another through an auxiliary module to allow conditions to work (Molenaar, 1998). This research was conducted to make a prototype of a composite computer simulation infrastructure model of an ecosystem. The outputs of this research proved that the approach seemed working well and able to be used as computer simulation infrastructure in studying various existing system start your abstract here.

Keywords: Design, Construction, Ecosystem Simulation, Workbench

1. Introduction

Efforts to study ecosystems have been carried out by humans since ancient times in various ways or approaches. The approach of using a computer model and simulation nowadays is one of the increasingly possible alternatives, especially with the rapid advancement of information and communication technology. The first paragraph after a heading is not indented.

There are several ways to build computer models of an ecosystem to be used in simulations. Composite modeling is one method that can be used. For example, Wight and Skiles used this approach to create an ecosystem model called SPUR (simulation of Production and Utilization of Rangelands) used in simulating ecosystem dynamics [1]. SPUR consists of several computer programs (main, sub-routines, functions, sub-programs) written using the FORTRAN procedural programming language, which as a whole are integrated into one final program that can be run on a computer.

Another approach is to use object-oriented programming (Object Oriented Programming) in making ecosystem models. For example, by use the object-oriented programming language SMALTALK, which is more convenient to use than procedural programming languages [2]. But the
use of object-oriented programming languages turned out to require the support of computer resources (memory) is quite large.

Another approach that is currently possible to use is different from what was explained earlier. Composite models are still used but each component of the model is kept separate which is not combined into one module when the simulation process of the ecosystem model is formed. Each module remains separate but will operate side by side and communicate with one another through an auxiliary module to allow conditions to work [3]. This research was conducted to make a prototype of a composite computer simulation infrastructure model of an ecosystem. Outcome of this research will also be in the form of scientific publications in national journals as well as models or designs of computer simulation infrastructure that are useful in studying various existing systems.

2. Research methodology
The first step in creating a structure for simulating a composite model is to establish a suitable development environment, such as the operating system and programming language, as well as various applications that are needed.

In this study an approach with a modular programming method has been used which has demonstrated its efficacy in solving the problem of modeling various complex systems [3]. The nature of this approach is dynamic, in the sense that each component of the system is made separately from one another. To support this, a C++ programming language environment will be used which is considered to be very efficient in utilizing computer resources, such as memory.

In its operation, all modules representing various system components will be run together under the control of one main module so that the overall model of the system being built can operate well in the simulation process. This condition is supported by the ability of the Windows 10 operating system that has multitasking capabilities.

2.1. Place and time
The study was conducted from April 2017 to December 2017 in the Laboratory of Agricultural Mechanization and Management Laboratory, Department of Agricultural Technology, Faculty of Agriculture, Sam Ratulangi University, Manado.

2.2. Research tools and materials
The tools used in this study are: 1. Intel Core i7-4500U, 1.80 GHz Processor Computer, 4GB memory, 230GB HDD, for programming, data processing and data analysis; 2. Operating system software used in this study is Microsoft Windows 10; 3. The programming language used is C++ which is available from the DEV C++ Integrated Development Environment (IDE) from Bloodshed. The stages of the research procedure carried out are as can be seen in Figure 1.
3. Results and Discussion
The complete projected simulation structure has a number of modules, each of which will be implemented as an independent process in a Windows 10 multitasking environment. The structure of each module in this composite simulation is divided into two parts: the system part and the program part (Figure 2).

Each module is further divided into three subsections, called declarations, initializations and iterations, which contain codes that correspond to the various phases of the simulation activity. Any information needed about the overall simulation structure is stored in the system module section. Basically, the system part acts as an envelope that surrounds the program part and allows it to function properly in the overall simulation structure. The program part contains all information relating to the process that a particular module will apply, and which is internally specific to the given module. For example, if the module is intended to implement an ecosystem model, part of the program will contain all information relevant to that model.
The available hardware and system resources are shared equally between modules, but only one process will be given access to these resources at a time. The most important module in the functional structure here is the Simulation Manager. Basically, this module governs the flow of simulation by deciding which module to process while others delay their activities. So, every time the module completes its task or has used the allocated run-time, the module will stop and wait for a signal from the Simulation Manager before starting the next job.

In the current simulation module design, the order of execution specified by the Simulation Manager is predetermined by the designer and is coded into the structure. However, ideally, the simulation flow should be kept as dynamic and flexible as possible. In the end, the sequence of simulations will be left open, and it is up to the Simulation Manager to decide which process should be run, according to the set of priorities that will be set.

At present, module synchronization is achieved by using the Critical Section as the object of flow synchronization as a facility available in the C++ language. It is this object that responsible for giving the opportunity for one process to run while another is given the right to proceed for a certain time interval. In total, the overall simulation structure uses up to a number of modules as needed. With the exception of the Simulation Manager, each module is associated with one Critical Section object.

4. Conclusion
From the results of this study it can be concluded that the design of composite computer simulation structures of ecosystem models produced in accordance with their intended purpose. The design of the structure referred to is able to accommodate the ecosystem model composite simulation adequately. Further development of ecosystem model development that will use the intended simulation structure is needed.

References
[1] Skiles J W and Schulbach C H 1996 Rehosting a mini-computer model on a super-computer Simulation 66(1) 43-58
[2] Baveco J M and Smeulders A M W 1994 Object for simulation. Smalltalk and ecology Simulation 62(2) 42-56
[3] Molenaar R 1998 Design and implementation of biosystem control and tools for biosystems simulation [Dissertation] Canada: Department of Agricultural Engineering, McGill University, Montréal QC