Environmental Modelling & Software xxx (2015) 1–19

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

A set of software components for the simulation of plant airborne diseases☆

Simone Bregaglio [a, *], Marcello Donatelli [b]

[a] Università degli Studi di Milano, Department of Agricultural and Environmental Sciences, Cassandra Lab, via Celoria 2, I-20133 Milan, Italy
[b] Consiglio per la ricerca in agricoltura e l'analisi dell'economia agraria, Centro di ricerca per le colture industriali (CRA-CIN), via di Corticella 133, I-40128 Bologna, Italy

ARTICLE INFO

Article history:
Received 15 May 2014
Received in revised form
27 March 2015
Accepted 26 May 2015
Available online xxx

Keywords:
Plant–pathogen interactions
Scenario analysis
Plant epidemic development
Biotic yield losses

ABSTRACT

Models to evaluate the impact of plant diseases on crop production under current and future climatic conditions are increasingly requested by different stakeholders. This paper presents four software components – *InoculumPressure*, *DiseaseProgress*, *ImpactsOnPlants*, *AgromanagementDisease* – which implement models to simulate the dynamics of generic polycyclic fungal epidemics and interactions with crop physiological processes. The software architecture adopted allows extending the components with alternate approaches to reproduce specific pathosystems or compare predictive capabilities. As proofs of concept, (i) the components are coupled with two crop simulators to reproduce wheat brown rust and rice blast epidemics and their impacts on leaf area and yield formation; (ii) spatially distributed sensitivity analyses are performed for rice in China and wheat in Europe to investigate model behaviour; (iii) a preliminary evaluation against observations of rice blast severity is performed in Northern Italy. The components are explicitly targeted to the modelling of crop–pathogen interactions to perform scenario analysis.

Software availability

The software development kit is available at http://components.biomamodelling.org/Default.aspx?productname=Diseases.

1. Introduction

Crop yield losses due to plant diseases represent a determinant of actual yield levels and are currently estimated at around 16% of worldwide food production (Oerke, 2006; Garrett et al., 2013). The impact of plant diseases in a changing climate is uncertain and controversial (Chakraborty and Newton, 2011), given the high number of interacting environmental and management factors contributing to the severity of plant disease epidemics (Coakley et al., 1999). Available studies indicate that changes in climatic conditions may lead to variable impacts on pathosystems, requiring that the future assessments of host–pathogen interactions are made via environment-specific evaluations (Ghini et al., 2012). It was demonstrated that climate change can lead to unfavourable conditions for pathogens that have now adapted (e.g., *Erysiphe graminis*, Hibberd et al., 1996), to shifts in the geographical distribution of the epidemics (e.g., *Phytophthora cinnamoni*, Bergot et al., 2004), or to the establishment of more aggressive pathogenic strains (e.g., *Puccinia striiformis*, f. sp. *tritici*; Milus et al., 2009) because of the high evolutionary potential of phytopathogen populations (McDonald and Linde, 2002; Pariaud et al., 2012). This increasingly draws attention to the need for reliable simulation models to estimate the impact of plant disease epidemics under current conditions and in consideration of climate change projections.

For decades, process-based modelling has been seen as the only viable method to draw realistic forecasts of the dynamics of plant diseases (Scherm and van Bruggen, 1994) and to provide consistent estimates of yield losses due to biotic stress agents. The availability of process-based plant disease models has been recognized by the scientific community as a priority (Magarey and Sutton, 2007; Pangga et al., 2011; Luck et al., 2011). However, technological constraints have limited so far the access to disease modelling to a number of users, and have not allowed the development of a framework which would ensure the re-use and extension of

☆ Thematic Issue on the Agricultural Systems Modeling & Software.

* Corresponding author. Tel.: +39 02 50316578; fax: +39 02 50316575.
E-mail address: simone.bregaglio@unimi.it (S. Bregaglio).

http://dx.doi.org/10.1016/j.envsoft.2015.05.011
1364-8152/

available models. One exception is highly empirical approaches, often specific to a single disease or even to a specific site. As a result, spatially distributed applications of simulation models to forecast crop production and/or assess climate change impact have mainly been focused on the direct effects of temperature change (Teixeira et al., 2013), use of nitrogen and water (Liu et al., 2013) and carbon dioxide (Hirata et al., 2013) on crop physiology, under the assumption of constant impact of plant diseases across years and environments (Donatelli and Confalonieri, 2011). In cases where an explicit coupling of disease and crop models was realized, the focus on specific pathosystems (e.g., Bastiaans, 1991; Luo et al., 1995) restricts the possible of reuse of these tools beyond the specific modelling applications they were built for. Even when generic plant disease models are available, the target to specific crop simulators (e.g., STICS-MILA, Caubel et al., 2012, 2014) constrains the adoption of a multi model approach to improve the reliability of the reproduction of the biotic impacts on crop yield. In fact, unless a dedicated architecture of the conceptual model and of the software framework are developed, enabling linking to multiple modelling solutions and allowing third party extensibility may require excessive resources, which would often make re-implementation more convenient.

The adoption of component-oriented programming as common practice in the development of agricultural modelling frameworks (e.g., APSIM, Holzworth et al., 2014; OMS3, David et al., 2013) could help to fill the gap, given the possibility of isolating knowledge of plant disease in discrete, extensible, and interchangeable software units. The software design of such components should not be driven by a specific framework architecture, but should include design traits representing a compromise between generality and specificity, in order to enhance the cross-framework capabilities for reuse (Donatelli and Rizzoli, 2008). A requisite to favour the diffusion of software components to a wide audience is the discretization of algorithms in fine-granularity model units (Meyer, 1988; Liu et al., 2002; Hocking et al., 2002), which in turn increases the transparency and the ease of maintenance of the modelling system. Another main functionality of framework-independent components is the increased ease in interfacing external tools to perform model sensitivity analysis (SA) and/or automatic calibrations (AC), which are common and good practices in the application of agro-environmental models. SA is used both to support model development (Jakeman et al., 2006) and to investigate model sensitivity to inputs and parameters (Bennett et al., 2013), whereas AC is commonly used to improve model performances in reproducing observed and measured data (Peña-Arancibia et al., 2015). Similar experiences in other domains of agricultural and environmental sciences have demonstrated the efficiency of component-oriented programming in supporting formalization of knowledge related to i.e. source pollution loads (Argent, 2005), the impact of agricultural management practices (e.g., Donatelli et al., 2006), simulation of water soil dynamics (Leavesley et al., 1996), and the qualitative aspects of crop production (Cappelli et al., 2014). Given that most environmental problems benefit from a multi-disciplinary analysis (Whelan et al., 2014), the availability of framework-independent software components strongly fosters the development of modelling solutions that integrate single-discipline approaches (Laniak et al., 1997; Donatelli et al., 2014).

Two main factors have so far limited the development of generic modelling frameworks for plant disease simulation: i) the lack of appropriate data, often collected to develop empirical models for use under specific conditions, and ii) the complexity and variability in the epidemiology of diseases, including differences between host cropping systems. Once the goal of data collection has been clarified, however, data can be effectively collected to parameterize models; furthermore, disease-specific modelling approaches can be added to generic core libraries, to allow their extension via sets of modelling options. Hence, there is the need for a modelling framework capable (i) to provide alternate options to build modelling solutions usable for operational applications, and (ii) to be further extended, autonomously by third parties, with new approaches for specific processes to facilitate models comparison.

The aim of this paper is to present four software components (called *Diseases*) implementing models to simulate the dynamics of a generic plant fungal airborne disease epidemic and its impact on crop production. The target requirements of this work are the development of modelling tools which can be used to explore situations where no reference data are available, for example conditions of climate change, and the extendibility of the components with alternative approaches to allow the simulation of specific pathosystems.

2. Description of the *Diseases* components

2.1. Software architecture

The *Diseases* components are *InoculumPressure*, *DiseaseProgress*, *ImpactsOnPlants* and *AgromanagementDisease* and consist of software libraries implementing input/output data structures and models to simulate a generic polycyclic fungal plant disease epidemic and to quantify its impact on crop production. The software architecture of the components is developed adopting three software design patterns, i.e., generically reusable abstractions of code solutions to commonly occurring problems within a given context (Gamma et al., 1995; Ludìk and Pitner, 2015):

- The *Composite* design pattern, to compose from simple models into higher-level models to represent part-whole hierarchies, while keeping the same interface.
- The *Strategy* design pattern, which allows the definition of the logic to switch at run-time between different models.
- The *Bridge* pattern, separating the data and communication model (data structures and interfaces) from the implementation of algorithms (simulation models), to allow interchanging model components via programming of applications against the data and interface library.

Each component consists of two independent and reusable software units, making available specific functionalities and providing access to their services via a semantically explicit interface (Donatelli and Rizzoli, 2008). Fig. 1 shows the Unified Modelling Language (UML) component diagram of the *Diseases* components, re-elaborated to highlight the main functionalities of each software unit.

The first software unit ([*Name_of_the_component*].Interfaces) contains the Application Programming Interface (API) of the component and the domain classes (Del Furia et al., 1995), which are data structures containing the input/output variables of the domain being modelled. To maximize ease of re-use, the API of each component implements the Create-Set-Call pattern design (Cwalina and Abrams, 2006), where the objects are first created via a default constructor, then some attributes are set, and finally the model is called (*Estimate* method in Fig. 1). Two overloads (i.e. variants) of the *Estimate* method are present in the API, which share the domain classes and the IStrategy[*Name_of_the_component*] parameters. One overload allows to run the test of pre- and post-conditions on each input/output variable of the model called. The programming interface used for models is the same for all the models belonging to a component. Available domain classes are States (state variables), Rates (rates variables), Auxiliary (intermediate variables), Exogenous (driving variables), ExternalStates and

Please cite this article in press as: Bregaglio, S., Donatelli, M., A set of software components for the simulation of plant airborne diseases, Environmental Modelling & Software (2015), http://dx.doi.org/10.1016/j.envsoft.2015.05.011