Designing modular frameworks for crop modelling: implementation and guidelines for use

Adam M1,2, Corbeels M2, Ewert F3, Leffelaar P1, van Keulen H1, Wery J2
1Plant production System group, Wageningen University, The Netherlands; m.adam@cgiar.org
2UMR System # 1230, CIRAD-INRA-SupAgro, 34060 Montpellier Cedex 1, France
3Crop Science Group, Institute of Crop Science and Resource Conservation (INRES), University of Bonn, Katzenburgweg 5, D-53115 Bonn, Germany

Keywords: model structure, uncertainty, software design patterns, good modelling practices, crop growth and development.

Introduction

Adoption of advanced software engineering techniques in crop modelling in the past decade has led to the construction of modular frameworks, consisting of libraries of models from which selections can be made. Advantages of a modular structure include the possibilities for: (i) interchange of code among models, (ii) testing of alternative hypotheses, (iii) use of simple or comprehensive modules as required for a particular application, and (iv) sharing of expertise. Although these advantages are undeniable (Acock and Reynolds, 1989), and have been illustrated, such as within the APSIM (Agricultural Production Systems Simulator, Keating et al., 2003) framework (e.g. McMaster and Hargreaves, 2009), little research has explicitly addressed the process of module comparisons or model adaptation within such frameworks. Modular frameworks provide technical possibilities to link modules, even if these links are meaningless in term of crop physiology or interaction between plant and soil. Therefore there is a need to support selection of modules in a consistent way between the agronomic domain and the software domain (Roux et al., 2010). The selection of the modules must be governed by the objectives of a specific simulation exercise. Thus, in addition to the implementation part of the framework (e.g. software engineering), there is a need to further focus on the model building process, and more specifically on the decision-making process of selecting one module rather than another (modelling part of the framework) and incorporating that module into the model structure (i.e. module assembly). This selection process is based on explicit criteria or approaches to guide model development. The aim of this article is to discuss the choices made in the development of modular frameworks in crop modelling at the implementation and modelling levels, and to discuss how such frameworks can contribute to the advancement of modular crop modelling, but also address its limitations.

Materials and Methods

CROSPAL (CROp Simulator: Picking and Assembling Libraries, Adam et al., 2010), APES (Agricultural Production and Externalities Simulator, Donatelli et al., 2010) and APSIM (Keating et al., 2003) are a few examples of modular frameworks, illustrating how modularity has been applied so far in crop modelling. Reflections on differences and similarities of the software design adopted to build these modular frameworks helped in identifying what is essential to create the libraries, without creating “Yet Another Modelling Framework” (Van Evert et al., 2005) and which are the consequences of their differences.
The assembly of the appropriate modules is based on criteria or systematic methodology for module selection. We suggest the use of three main methodologies to define and facilitate the assembly of modules: (i) uncertainty matrix, (ii) model comparison, and (iii) expert knowledge elicitation. The uncertainty matrix distinguishes different types and sources of uncertainties in order to facilitate uncertainty classification. We apply this methodology to select a soil nitrogen module (mineralisation explicitly described or not) to simulate crop growth in response to nitrogen management and emphasize the importance of accurate simulation of nitrogen uptake. Model comparison enables the investigation of the effect of the level of detail incorporated in process-based crop growth models on simulated potential yields under a wide range of climatic conditions (Adam et al., 2011). Finally, expert knowledge elicitation is used to define a new crop model for grain legume (pea, *Pisum sativum* L.) from an existing cereal (durum wheat, *Triticum durum* L.) model.

**Results and Discussion**

The software design of all three frameworks studied (Table 1) allows modularity and flexibility in adapting the model structure. The choice for one specific design mostly depends on the expertise of the future user of the modular framework. Use of an XML file (Table 1) to configure a model (i.e. define its structure), as possible in APSIM, provides total flexibility to the user to select any module, no matter whether the different modules “fit” together conceptually. Technically, everything is possible once individual modules have been properly implemented and modifications in the configuration of the structure of the model escape any consistency check in the agronomic domain. In APES, the use of the composite strategy provides less freedom to the user, as the developer defines this composition within the component (IStrategy) on the basis of his or her own opinion on the anticipated future modelling exercise and/or application. However, the selection of a specific model structure still remains the responsibility of the user through the use of model options. In CROSPAL, the choice for the use of the abstract factory (a way to encapsulate a group of individual factories that have a common theme, Table 1) relies on the logic used to assemble the crop model. The way module are selected and assembled is the consequence of the vision of the developer in the agronomic domain (e.g on crop functioning) and should correspond to the different criteria included in the graphical user interface, criteria defined with the help of uncertainties studies.
### Table 1. Comparison of the different software designs adopted in different crop modelling frameworks (including definitions of technical terms)

<table>
<thead>
<tr>
<th>Level of granularity</th>
<th>CROSPAL</th>
<th>APES</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic crop processes</strong></td>
<td>Strategy</td>
<td>Strategy</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>Component</strong></td>
<td>Abstract factory and criteria with a GUI</td>
<td>Composite strategy (IStrategy: interface)</td>
<td>Generic configuration</td>
</tr>
<tr>
<td><strong>Crop</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Crop models</strong></td>
<td>Definition of new concrete factories</td>
<td>Components linked via wrapper (using the ModCom framework)</td>
<td>GCROP</td>
</tr>
<tr>
<td><strong>Soil-crop</strong> (i.e. crop simulator)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Factory**, in object-oriented computer programming, a factory is an object for creating other objects. It is an abstraction of a constructor, and can be used to implement various allocation schemes. **GCROP**, the generic crop model template of APSIM; **GUI**, graphical user interface; **ModCOM**, a software framework to assemble simulation models (Hillyer et al., 200); **Strategy design pattern**, define a family of algorithms, encapsulate each one, and make them interchangeable; **Wrapper**, classes enabling combination of other classes that could not be combined, because of incompatible interfaces; **XML**, Extensible Markup Language

Use of uncertainty matrix, model comparison and expert knowledge elicitation stressed the importance of the documentation of the modelling process. These uncertainty assessments should be seen as the basis of the logic that enables one to go from the objective of the simulation to the “appropriate” crop model (Figure 1). The definition of this logic is based on a stylised decision-making process rather than based on pragmatic decisions (e.g. data availability, expertise...). The use of the uncertainty matrix emphasized the importance of explicitly defining the unknown. The use of model comparison enables one to tackle the issue of the required level of detail and highlights the risks of over-simplification of processes when data are scarce (Adam et al. 2011). And finally, the integration of expert knowledge in the development of the framework emphasizes the importance of explicitly describing the underlying assumptions through the use of conceptual modelling and the future potential of visual tools such as declarative modelling. Using these three methodologies, we derived an explicit description of the conceptual model, including (i) the domain of application, (ii) the type of model required, (iii) the relations and assumptions underlying the choices, and (iv) the verification of the conceptual model.

![Figure 1. Approaches used to guide the selection of the model that represents the system under study.](image)

While technical advances have stimulated substantial progress in the crop modelling field, especially in providing modular frameworks that allow easy coupling of different models at a higher scale for use in integrated assessment studies (Van Ittersum et al., 2008) or for further understanding of crop physiology (Hammer et al., 2002), conceptualisation of the systems...
remains an essential step. Consequently, the crop modeller should act as an interface between the developers and the end users: he or she must understand a minimum of all disciplines involved in the process of model development (Roux et al., 2010), integrating knowledge from the users (i.e. agronomists) and the developers (software engineers) to bridge the gap.

References


