Geophysical Research Abstracts, Vol. 9, 07755, 2007
SRef-ID: 1607-7962/gra/EGU2007-A-07755
© European Geosciences Union 2007

An object-oriented framework for a process-based soil-nitrogen model component

P. Neuhaus, C. Klar and K. Schneider

Department of Geography, University of Cologne, Germany (peter.neuhaus@uni-koeln.de / Phone: +49 211 470 7805)

Whereas a large number of existing ecosystem models are programmed following the traditional procedural programming paradigm (e.g. FORTRAN, C), modern software development paradigms like Object-Oriented Programming (OOP) become more and more prevalent. Within the GLOWA-Danube project, the object-oriented, distributed and multi-disciplinary decision support system DANUBIA is developed. The Soil Nitrogen Turnover (SNT) Model is one component within this complex system. Using the example of the SNT model component, the use of OOP in ecosystem modelling is demonstrated and its costs and benefits are discussed.

The development of the model-framework followed a process of Object-Oriented Analysis and Design (OOAD), with the graphical Unified Modeling Language (UML) as a tool for drafting, documentation and communication among the developers. Standard OOP-techniques such as encapsulation and interfaces have been applied to organize the code into communicating processes and data-pools and to separate scientific content from technical complexity as well as to make the data- and program-flow transparent and manageable. Advanced OOP-techniques such as design patterns, reflection and dynamic class loading were employed to further increase the usability of the model-framework especially for scientific personnel.

OOP is well suited for the development of an ecosystem model, since it provides a stringent and transparent structure, and an easily manageable, communicable, modular, scalable and flexible code. The encapsulation of each process in its own object has significantly reduced the side-effects among different processes and supports the analysis of interactions and feedback mechanisms. Furthermore, model development, testing and validation as well as model extensions have been simplified.