

Landscape Management Framework (LMF) - development and application of a new concept for a dynamic landscape management model

David Windhorst^a, Philipp Kraft^a, Hans-Georg Frede^a, Lutz Breuer^a

^a *Justus-Liebig-University Giessen Institute for Landscape Ecology and Resources
Management (ILR)*

Abstract:

The evaluation of current and future land use and management impacts on ecosystem processes often requires the application of a wide range of specialized models covering different environmental research disciplines. We promote the newly developed Landscape Management Framework (LMF) as a way to facilitate and ease the consideration of land use and management actions within such integrated modeling approaches. LMF takes advantage of existing modeling approaches, provides easy to handle interfaces to external models, and mediates between them. Virtual model stakeholders (agents) within LMF follow a set of user defined decision rules according to binary and fuzzy logics to allow semi-optimal and customized decisions to be made in order to achieve optimal solutions and management timing even during unprecedented events. Preliminary results of LMF are promising and will be further tested with case studies to follow.

Keywords: landscape management; modelling; framework; decision support.

1 INTRODUCTION AND JUSTIFICATION

The landscape as we know it, is always composed of a variety of elements and related processes that are studied by different scientific disciplines. In the past each of these disciplines has developed sophisticated models that describe these processes. Whenever modellers of different environmental research disciplines seek to pool the respective system knowledge that has been gained by the development of specialized models, they often encounter difficulties in mediating the flow of information between the models.

To promote the re-usability and applicability of such specialized models in a wider scope, frameworks have been developed to ease the exchange of information between them (Argent, 2004; Argent et al., 2006). Problems occurred when trying to make the models communicate within the framework and to anticipate the type of information required for the interaction between the models via the framework. Rather rigorous rules have been necessary to bind the flow of information within the frameworks.

To escape from the fetters of a fixed framework dependent model development the works of Jones et al. (2001) and Donatelli and Rizzoli (2008) promote rather flexible bonds between different models through a standardized interface between model components. To date, the large majority of models do not comply with such a generally accepted standard, and neither do they share a common programming language. Thus we are left with the incompatibility of great models and great frameworks.

To overcome these problems LMF goes one step further and resolves the precondition of a common interface and programming language and postulates the usage of a wrapper around the existing models. This allows the information to flow between models and between different programming languages, in this case via Python. Wrapper generators interfacing Python with Java (Jython), Fortran (f2py) or C++ (SWIG) are available for all common operating systems and offer the means to provide a common ground for the wide variety of models used in the scientific community.

Besides the structure to implement different models, a framework needs to facilitate the transfer of information between models. To anticipate all rules necessary to mediate the flow of information between them is an extremely complex matter, as the set of rules necessary to account for all eventualities soon becomes unmanageable. It is especially difficult to anticipate all the requirements to control the flow of information based on human decisions. Each implementation comes with a unique set of necessary rules to process and control the available information within the framework. The degree of available information in the framework thereby increases with the number of implemented models contributing to the common knowledge base (e.g. a hydrological model contributes data on soil water content; decomposition model delivers input data on nitrogen availability) on which the decision maker can base its decision. In contrast to most common approaches LMF does not seek to anticipate all eventualities in advance. It rather provides the fundamentals to adapt and extend the existing rules to meet the unique requirements of each setup of the framework.

Finally, to control and conduct the flow of information between the models within a framework representing an environment influenced by human decisions, it is mandatory to introduce virtual model stakeholders (subsequently referred to as *agents*) into the system.

The agents within LMF are capable of mimicking the human decision making process in a resource-limited environment based on a set of user defined rules and management options. The aim of this paper is to introduce the design and the functionality of LMF, thereby focusing on the ability to incorporate agents into the desired model environment, and highlighting the benefits of the chosen path.

2 LANDSCAPE MANAGEMENT FRAMEWORK (LMF)

The newly developed 'Landscape Management Framework (LMF)' offers the means to incorporate virtual model stakeholders (agents) which mediate between different models representing different compartments of the real world within a framework. The decision-making processes of these agents regarding different land use options can be subdivided into three sets of decisions, which need to be addressed: "What is the land used for?", "Where is the land use allocated to?" and "How and when is the land managed?".

It is assumed that the external constraints ("What is the land used for?") controlling the regional demand for land use services are part of a user defined land use scenario, provided externally and not simulated by LMF. The question "Where is the land use allocated to?" will be answered either by an external land use allocation model (e.g. CLUE, Veldkamp and Fresco (1996); SITE, Schweitzer et al.

(2011)) or the internal 'land use allocation module', an additional model component of LMF which is currently under development. The land use allocation module will be responsible for predicting the site specific location of a land use in the real environment and assigns plots to an agent, who will manage the plots in the course of the following model run.

Like in the model of van Ittersum et al. (2008), we introduced the agents to be the core objects of LMF, representing the main actors whose actions affect the environment. All agents are thereby embedded in a collective environment with a shared market, equal spatial extend and the same resource limitations through which they communicate.

Within LMF the agents are organized by the 'landscape management module'. The agent decides "How and when is the land managed?". Therefore they formulate *management tasks* to arrange and allocate the necessary resources to the actual executor (e.g. employees and/or machinery) of the upcoming management assignment. A certain management option (e.g. cultivate maize) consists of a series of successive mandatory or optional tasks, always offering the possibility to choose between implemented alternatives (e.g. choosing between manure or mineral fertilizer).

2.1 Management tasks

In analogy to Martin-Clouaire and Rellier (2009) all *management tasks* have a state of execution which controls the life cycle of a task (Figure 1). The constraints controlling the execution of a waiting task are evaluated at each time step by the decision algorithm described in section 2.3 of this paper. The kind of system parameters used in the constraints (execution terms) are only restricted by the availability of information in the framework. For example, the execution of an upcoming task could be triggered by the phenological status of the plant if a plant growth model is applied within the framework.

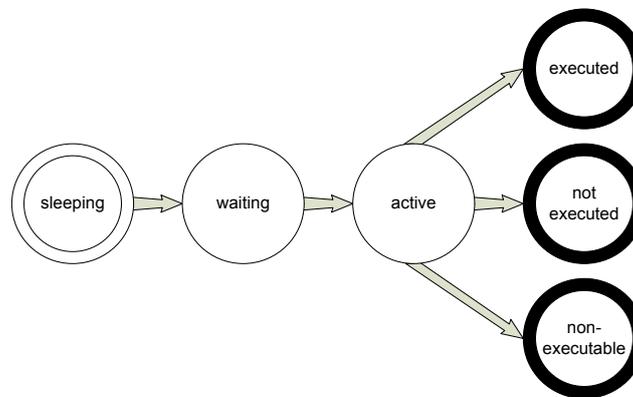


Figure 1 Task life cycle. Six states of execution are distinguished during the life time of a task: 1. sleeping (on initialization), 2. waiting (next tasks to be executed), 3. active (ready to be executed) and three final states executed (successfully processed), not executed (an alternative option was chosen) and non-executable (execution became impossible – if this task was mandatory a new management plan will be generated).

Once a task is executed the management effect is directly mediated to the other models within the framework using the signal/slot mechanism. This is a well established and powerful programming technique, used to convey information between two independent components (Dalheimer, 2002). Each management task sends, on execution, an associated signal (e.g. fertilization or harvest operation) and a model connected to this signal via a corresponding slot (e.g. hydro-geochemical or plant growth model) can react accordingly (e.g. add fertilizer to the upper soil layer or terminate the plant growth).

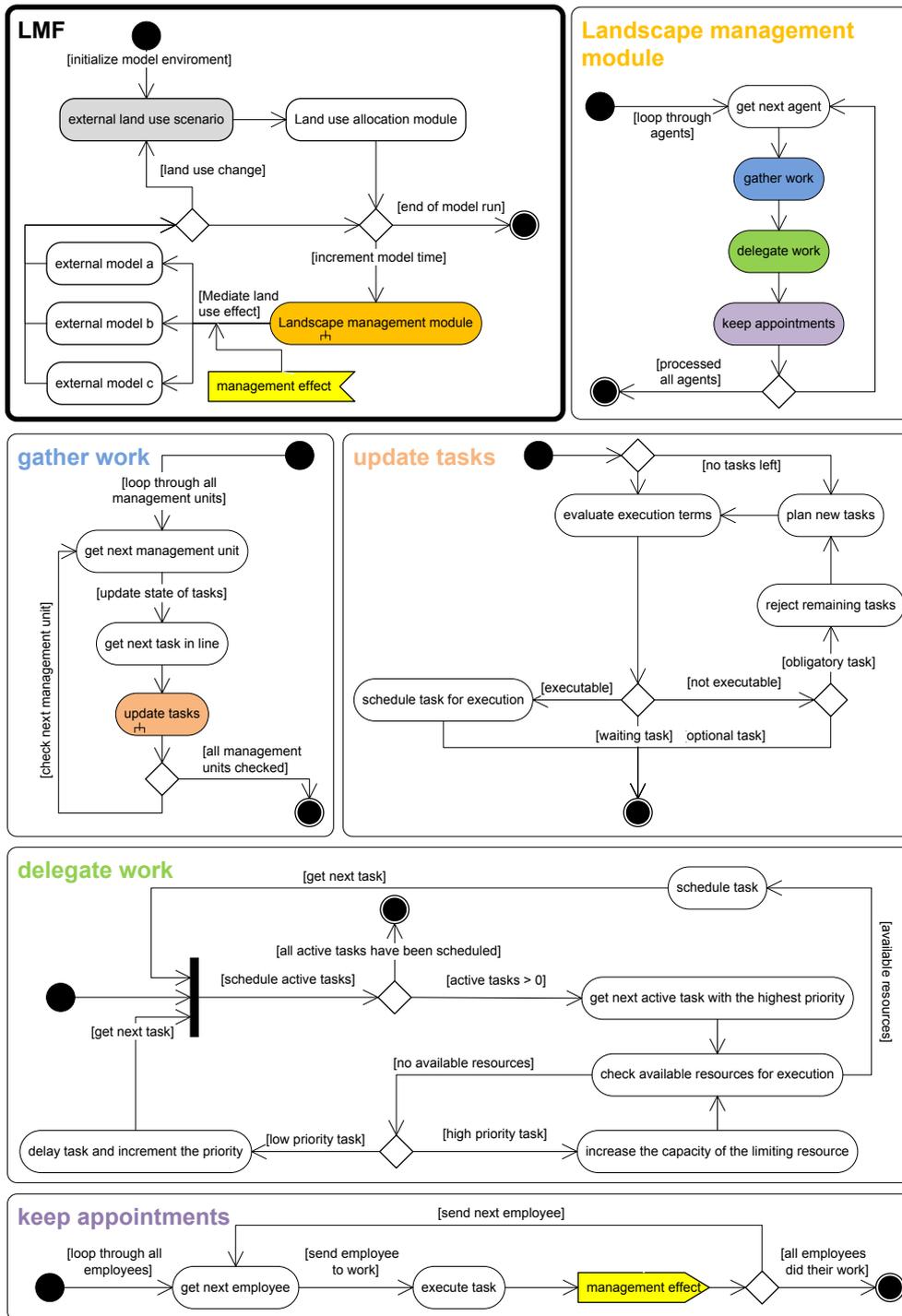


Figure 2 Activity diagram visualizing the processes and steps involved in a LMF-model run following UML notation

2.2 Model run

The steps performed within LMF during one model run can be summarized as follows and are illustrated in more detail in Figure 2. During initialization the land use is allocated to the responsible agents. Those agents start to take care of their management units in accordance to the current state of the system and communicate the management effect back to the other models deployed within LMF during each time step (by default on a daily basis).

During each time step every agent within the 'landscape management module' first checks every management unit to see if a management task has become executable and then allocates the necessary resources to the active task and schedules it for execution. Finally the scheduled appointments are carried out by the executors of the task.

A comprehensive description on how to set up the framework and its components and how to integrate external models into the framework via the interface provided by LMF and vice versa will be available in the documentation of LMF (<http://www.uni-giessen.de/cms/ilr-download>). Under the same address a preliminary version of LMF and its documentation is currently available.

2.3 DECISION ALGORITHM

The decision if a task will be executed is based on the feasibility to carry out the task under the actual environmental conditions. If conditions are favourable for the task, LMF simulates the expectation of the agent. LMF also forecasts conditions for carrying out the task, if they become better by waiting, the task will be delayed.

The feasibility μ of an agent to perform task T under the condition set x at the time t is modeled as a fuzzy constraint (Zadeh, 1978). The feasibility for one condition x_i is defined using a classical trapezoid function, based on a user supplied database. The overall feasibility for all conditions is defined as the combination of all feasibilities using the Fuzzy-AND operator minimum:

$$\mu_T(t) = \min(\mu_T(x_i(t))) \quad (1)$$

Fuzzy logic (Zadeh, 1978, 2008) has the advantage that states between TRUE and FALSE can be distinguished. This is convenient if you want to base the decisions on the environmental state of a system which is rarely perfect. The rules provided by Zadeh (1978, 2008) thereby allow to evaluate several environmental conditions at once.

If the task is feasible at time step t ($\mu_T(t) > 0$), the expectation of the agent is the trigger for the execution. If the agent expects better conditions for the next day, the task will be postponed. In the case of less suitable conditions in the future, an immediate execution is advised. Stable conditions are treated according to the rule "sooner rather than later". The expectation of the agent is modeled by calculating the trend $\Delta\mu_T$, as:

$$\Delta\mu_T(t) = \mu_T(t + 1) - \mu_T(t) \quad (2)$$

The feasibility of the next time step, can either be estimated based on the driving data for the next time step $\mu_T(t + 1)$ or by extrapolation of the last day's trend $\Delta\mu_T(t - 1)$.

To illustrate this approach we assume a field melioration which is ideally carried out between the 50th and the 55th day of year (DOY), but at least between DOY 45 and DOY 60. Furthermore, a medium soil water content between 12.5% and 27.5% with an optimum at 20% is needed to perform the task (Figure 3). Rainfall of the next day is added to the soil to estimate the development of soil water content, while the estimation of the date is trivial.

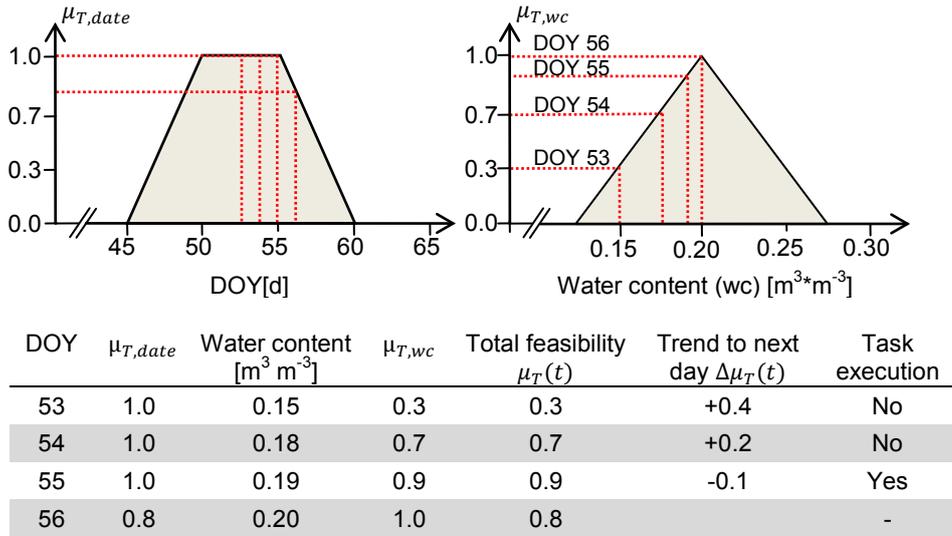


Figure 3 Illustration of the concept of the ‘decision algorithm’ used to trigger tasks depending on the system parameters day of year (DOY) and water content (wc). If there is no further improvement of the trend, the process is eligible for execution. The total feasibility μ_T and the expected feasibility trend $\Delta\mu_T$ are calculated according to eq. 1 resp. eq. 2

Figure 3 shows the decision algorithm on four consecutive days for the example mentioned above. For the days 53 and 54, the task is feasible to carry out, but the agent expects better conditions on the next day ($\Delta\mu_T > 0$). As a result, the task is postponed. On day 55, the forecast module reports a decreasing trend for day 55 to 56 ($\Delta\mu_T < 0$), therefore the task is triggered for execution.

In addition to execution terms used to trigger a task, termination conditions can be set to abort a management action (e.g. if a certain time window has past or a plant exceeded a certain development stage). The abortion changes the state of a task to ‘non-executable’. In the case that this task was obligatory all following tasks will be set to ‘not-executed’ and a new, adapted management plan will be developed. If it can be foreseen that the termination condition will be met in the next time step the overall trend will be set to negative per default.

3 CONCLUSION AND REMARKS

The major objectives of this paper were to outline the ideas of the developed concept of LMF and to accentuate the functionality of the implemented framework and its components.

The requirements for a spatial explicit modeling framework simulating the landscape management on a stakeholder level have been encapsulated in an object-oriented manner taking current approaches as antetype. The structure and approach in use were greatly influenced by the work of Martin-Clouaire and Rellier (2009), Donatelli and Rizzoli (2008) and van Ittersum et al. (2008). The prevalent binary logic in these approaches was successfully extended with the concept of fuzzy logic (Zadeh, 1965, 2008). A trend analysis of observed parameters thereby allows to trigger a given management action even under semi-optimal conditions. The implemented routines of the model are mostly replaceable and extendable to serve the requirements of the user.

In the final state, the LMF, including the fully developed ‘land use allocation module’ and the ‘landscape management module’ shown here, will have the capacity to combine a large variety of models representing different compartments

of the environment. During first tests in the implementation phase, LMF has shown promising capabilities to simulate and anticipated management actions and adapt its decisions to the current state of the system. In the course of two virtual case studies with a coupled plant growth and a hydrological model (results not shown here due to page limits), the ability of LMF to predict and optimize agricultural management practices on the landscape scale was additionally thoroughly tested.

Nonetheless it has to be acknowledged that the advantages of the presented concept come at a certain cost. The implementation of the wrapper for external models and the connection of the signals emitted by LMF and the external models to the receiver (slot) require certain programming skills. Although LMF comes with a basic user database for standalone operation, a considerable amount of work and system knowledge will be required to extend and adapt the underlying rules and management options for LMF when it is coupled with other models.

Despite those restrictions we believe that the capabilities to consider and evaluate different management actions, the broad scope of applications and the flexibility of LMF are well worth the effort and fulfill the need to deploy independent models generally not compatible with any framework in a common model environment.

REFERENCES

- Argent, R.M., 2004. An overview of model integration for environmental applications—components, frameworks and semantics. *Environmental Modelling & Software* 19, 219–234.
- Argent, R.M., Voinov, A., Maxwell, T., Cuddy, S.M., Rahman, J.M., Seaton, S., Vertessy, R.A., Braddock, R.D., 2006. Comparing modelling frameworks – A workshop approach. *Environmental Modelling & Software* 21, 895–910.
- Dalheimer, M.K., 2002. *Programming with Qt*. O'Reilly Media, Inc.
- Donatelli, M., Rizzoli, A.E., 2008. A design for framework-independent model components for biophysical systems, in: *Proceedings of the iEMSs Fourth Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSs 2008)*. Presented at the iEMS'08, Barcelona, Catalonia, pp. 727–734.
- van Ittersum, M.K., Ewert, F., Heckeley, T., Wery, J., Alkan Olsson, J., Andersen, E., Bezlepina, I., Brouwer, F., Donatelli, M., Flichman, G., Olsson, L., Rizzoli, A.E., van der Wal, T., Wien, J.E., Wolf, J., 2008. Integrated assessment of agricultural systems – A component-based framework for the European Union (SEAMLESS). *Agricultural Systems* 96, 150–165.
- Jones, J.W., Keating, B.A., Porter, C.H., 2001. Approaches to modular model development. *Agricultural Systems* 70, 421–443.
- Martin-Clouaire, R., Rellier, J.-P., 2009. Modelling and simulating work practices in agriculture. *International Journal of Metadata, Semantics and Ontologies* 4, 42–53.
- Schweitzer, C., Priess, J.A., Das, S., 2011. A generic framework for land-use modelling. *Environmental Modelling & Software* 26, 1052–1055.
- Veldkamp, A., Fresco, L.O., 1996. CLUE: a conceptual model to study the Conversion of Land Use and its Effects. *Ecological Modelling* 85, 253–270.
- Zadeh, L.A., 1965. Fuzzy sets. *Information and Control* 8, 338–353.
- Zadeh, L.A., 1978. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1, 3–28.
- Zadeh, L.A., 2008. Is there a need for fuzzy logic? *Information Sciences* 178, 2751–2779.