7.2 DESIGNING A WRAPPER FUNCTION TO ALLOW A STANDARDIZED COMMUNICATION CHANNEL BETWEEN THE MODELS

Background and key outcomes

The MIND STEP suite of models is set up as a modular framework where functionality can be added with additional models and data and where the models can be executed in flexible combinations. The latter requires embedding the models in an overarching structure following protocols developed in other work packages, which specify interfaces and the structure of input and output data for connecting the models. These protocols were put forward in MIND STEP deliverables 3.1. "Specification of model requirements: Protocols for code and data" and 3.2. "An overarching IDM model structure: Interfaces within the MIND STEP model toolbox".
Based on the concepts proposed in these deliverables, the bio-economic farm-level model FarmDyn was selected as core model, to which other, newly developed tools, could be linked in a modular fashion, such that they could be activated and executed within the FarmDyn environment. Still, newly developed methods, for instance to create the FarmDyn database by accounting for behavioural aspects of farm-level decision making, have to be executed separately. The same applies to linkages with market- and sector-level models as described in deliverable 5.2 "Report on improvements to the current EU and global models." One example is the use of FarmDyn results to update marginal abatement cost curves in the MAGNET model for dairy farms the EU Member States. This requires a separate execution of FarmDyn and MAGNET, using their respective environments.
Already in the conception phase of MIND STEP, it became clear that establishing a workflow between new and existing models would be greatly facilitated by using wrapper functions for the included models. One advantage of such wrappers is that they permit the execution of each model from an external environment, like the R programming language. The wrapper concept reduces the work to combine the models as it does not require changes to their programming environments and ensures that scenario settings and data are standardized and well documented.
Several simulation models in the MIND STEP toolbox, like FarmDyn, GLOBIOM, and IFM-CAP, feature a graphical user interface (GUI), realized with the „GAMS Graphical Interface Generator" GGIG (Britz 2014). This GUI facilitates, for instance, choosing the included modules or the database to use. In the case of the FarmDyn model the GUI permits constructing parts of the model database, the selection of modules, the steering of the solving algorithm, and the exploitation of results. In addition, the parallel execution of many model instances is also governed by the GUI. As several models are supported by the GUI, the wrapper function was programmed to take advantage of the most commonly used GUI features, starting with the example of the FarmDyn model.

Methodological developments: Applying the wrapper concept to the MIND STEP toolbox

In general terms, a wrapper can be understood as a function in a given programming environment that calls another, more complex function, provides the required function arguments, and executes it. The complexity of the wrapped function is reduced for the user by providing many of the requiring arguments as defaults, thus limiting the additional user

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 817566.

input. A simple example from the R programming language would be a function that uses a basic function to calculate the mean of a dataset, where the user can decide if not available data points should be treated as zeroes or not at all. Such a "mean(data, removeNotAvailable)" function would take two arguments: the data for which the mean should be computed and the information whether non-available data (NA) points should be removed. If for some reason a program requires a general solution for the treatment of NA values, a wrapper could be a function that calls the original mean function but sets the argument removeNotAvailable to "True". In the R programming language, such a wrapper function "**mean_alwaysremoveNA**" would be implemented like this:

```
mean_alwaysremoveNA ← function(data) {
        return(mean(data, removeNotAvialable = True))
        }
```

In this case, the user has only to provide the data while other arguments are passed to the underlying function automatically.

Similarly, the FarmDyn wrapper "runFarmDynfromBatch" requires the user to provide a set of arguments (location of FarmDyn on the computer, paths to the GAMS software, general user settings, scenario settings) and a farm dataset, which then executes one or multiple instances of FarmDyn by calling the GUI from R (see Figure 1).

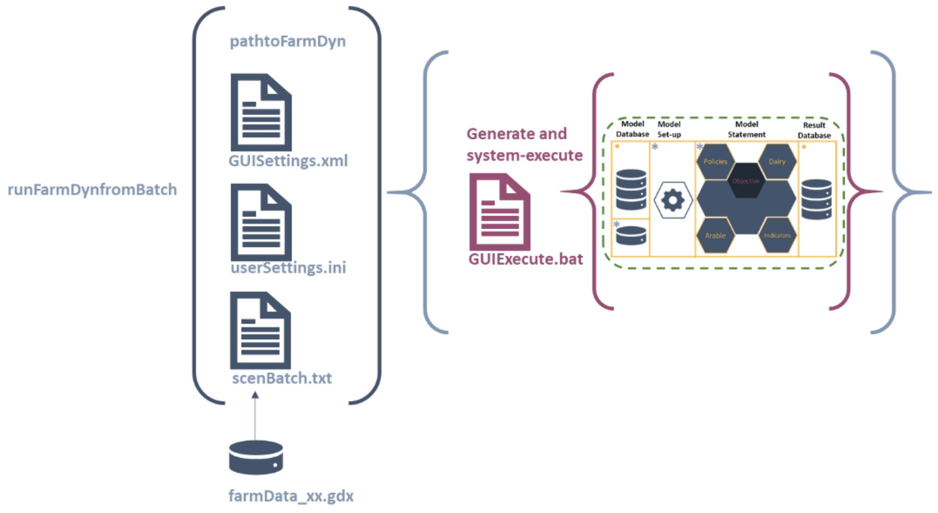

Figure 1 Schematic view of the wrapper "runFarmDynfromBatch"

Routines for the generation of the needed datasets from commonly used statistical sources and a range of standard results reporting functions are bundled in the R-package (or library) farmdynr. This package can be found in https://gitlab.iiasa.ac.at/mind-step/farmdynr . In R, the command "install.packages("https://gitlab.iiasa.ac.at/mind-step/farmdynr")" will install the FarmDynR package. It has to be noted that the FarmDyn model and a datasets named farmData_xx.gdx has to be found in the appropriate location as identified by the FarmDyn manual.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 817566.

Conclusions

MIND STEP has developed a concept for wrapper functions to execute models in the MIND STEP toolbox from an external environment, such that inputs and outputs can be exchanged between models and results can be combined. The proposed concept is based on the observation that most models in the MIND STEP toolbox take advantage of the GGIG program (Britz, 2014) for model execution and visualization, which provides advanced model steering functionalities and, most importantly for the purposes discussed here, a batch execution facility.

MIND STEP internal surveys on programming practices and modellers experiences, together with personal communication, showed that most modellers in MIND STEP are at least to some extent familiar with the R programming language. The R environment permits functional programming, efficient handling of complex data structures, and a standardized way to develop program libraries in the form of R-packages. Based on such packages, GAMS-specific data formats can be easily processed.

Based on these observations, a wrapper for the execution of the FarmDyn model exploiting the GGIG facilities was programmed in R. In its current state, it executes the FarmDyn model (or any other GGIG-driven model) from the GUI to create a set of files that contain a large number of default settings. Once these files are generated, simple changes can be made rather easily and then used to call the proposed wrapper function, which permits the parallel execution of several instances of the FarmDyn model from an R program flow. The proposed approach is therefore conceptually applicable to several models in the MIND STEP toolbox in addition to FarmDyn (IFM-CAP, AGRISPACE, GLOBIOM) as it exploits functionalities provided by R and GGIG.

The wrapper function was bundled with routines for the generation of the needed databases from commonly used statistics as well as functions to report and visualise results in an R-package named famdynr, which can be download at https://gitlab.iiasa.ac.at/mind-step/farmdynr.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 817566.