19th International Congress on Modelling and Simulation, Perth, Australia, 12–16 December 2011
http://mssanz.org.au/modsim2011

Evaluating and improving OpenMI as a model integration platform across disciplines

M.J.R. Knapen [a], **S.J.C. Janssen** [a], **O.R. Roosenschoon** [a] **and P.J.F.M. Verweij** [a]

[a] *Alterra, Environmental Sciences Group, Wageningen University and Research Centre,*
Droevendaalsesteeg 3, 6708 PB Wageningen, The Netherlands
Email: rob.knapen@wur.nl

Abstract: For decision makers in the domains of agriculture and environment, for instance in government agencies, farming, and environmental NGOs it is beneficial to evaluate ex-post or to asses ex-ante the impacts of their choices. Modelling and modelling tools can be helpful by providing a simplified representation of reality, simulating potential contrasting pathways into the future and improving the understanding of interdisciplinary cause-and-effect relationships.

To research these interdisciplinary relationships, models developed by different scientific disciplines and often operating at different scales can be integrated into model chains that cover processes across disciplines. Integration allows the single disciplinary models to complement each other and thereby provide comprehensive and balanced assessments across scales and disciplines.

In order to assemble models into an operational model chain multiple levels of integration have to be taken into account. First and foremost methodological or conceptual integration has to take place, which focuses on aligning different scientific methodologies and identifying required model improvements necessary for meaningful linkage. Secondly semantic integration has to take place to achieve the use of a common language to describe the models and the data they consume and produce in order to avoid failed integration due to ambiguous or misunderstood terminology. It also helps in reaching shared understanding and goals between the modellers involved.

Thirdly, and the main focus of this paper, technical integration has to take place. Here the aim is to ensure repeatability and reproducibility of model chain runs and to optimize use of computer hardware for model simulations. Technical integration itself can be achieved by different approaches (i.e. manual, scripting, building or using a proprietary framework, using an open framework based on standards). The choice of approach depends on the institutional and project context and on the preferences of the researchers involved. This paper will further focus on framework based technical integration. From the many available modelling frameworks (e.g. OMS, TIME, KEPLER, FRAMES, MODCOM, OpenMI, etc.) the emphasis will be on OpenMI, the Open Modelling Interface and its use and usefulness as a readily available, generally accepted and open standards based framework.

OpenMI is a open source software standard for dynamically linking models at runtime, which can potentially be used in many domains, but is currently mainly applied in the water and environmental domains. This paper examines the use of OpenMI in several multi-disciplinary large projects that worked on integrated models. These projects operated in the disciplines of agriculture, land use, nitrogen cycling, forestry, hydrology and economics. The overall objective is to investigate the strengths and weaknesses of integrated modelling according to open frameworks based on standards in general, and OpenMI in particular, based on feedback from both software developers and modellers that contributed to the aforementioned projects on the use of OpenMI, combined with the authors' knowledge and experience. Recommendations for improvements of OpenMI specific and integrated modelling in general will be presented.

Preliminary findings point to the existence of tension between building a model with a stand-alone purpose and building a model suitable to be linked to other models. Making a model linkable turns out to be an effort that is neither a primary goal for the modeler or for the framework developer. This is not helped by the typical OpenMI framework Software Development Kit (SDK) implementation tending to have a fairly heavy weight impact on the models, where recent studies promote a more lightweight approach to frameworks to make their adoption easier. Steps taken at Alterra in the development of the OpenMI 2.0 SDK for Java to improve this situation will also be mentioned.

Keywords: *Modelling frameworks, OpenMI, model integration, technical integration, interdisciplinary*

1. INTRODUCTION

Evaluating ex-post or assessing ex-ante the impacts of their choices with the help of modelling and modelling tools is useful for decision makers in the domains of agriculture and environment. Models represent parts of reality as simplified quantitative or qualitative systems. Together with modelling tools they can help simulate potential contrasting pathways into the future and improving the understanding of interdisciplinary cause-and-effect relationships. An ex-post evaluation occurs after such a choice has been made, while an ex-ante assessment tries to simulate the potential impacts of choices before these are made. In ex-post evaluation, data is likely to be available or can be collected on relevant variables in the period after the choice took effect. In contrast, an ex-ante assessment tries to shed some light onto the future and data is not available.

Ex-ante assessments through models and modelling tools could provide valuable insights on potential choices for complex societal and environmental problems (e.g. climate change, achievement of the Millennium Development Goals (UN, 2005)). A prominent example of the use of an ex-ante assessment is the assessment of the likely impacts of climate change on the biophysical environment and society (IPCC, 2000) by the Intergovernmental Panel on Climate Change. An example on a lower spatial scale is the FARMSCAPE project (Carberry, et al., 2002), in which farmers, advisory services and researchers jointly applied a simulation tool to assess the potential for alternative management strategies of cropping systems on Australian farms. Such ex-ante assessments need to involve multiple disciplines and cover multiple scales.

Single-disciplinary models cover only a few processes from a single domain, be it economic, agricultural or environmental and lack descriptions of some relevant processes. These models generally do not cover the relevant multiple scales to handle all assessment questions. Mono-disciplinary models can complement each other and thereby provide comprehensive and balanced assessments across scales. Therefore, it becomes necessary to integrate models from different disciplines, sometimes operating at different scales into model chains, covering processes across disciplines. In order to arrive at an operational model chain for applications in integrated assessment procedures, semantic, methodological and technical integration of models is required. Ideally, model integration leads to a model chain, that can be trusted, is transparent, and can be used and understood by a community of researchers, not only the individual modeler.

Semantic integration means speaking a common language and achieving a shared understanding between all models and modellers working together (Jakobsen and McLaughlin, 2004; Hinkel, 2008; Scholten, 2008). The methodological integration focuses on aligning different scientific methodologies and identifying required model improvements necessary for meaningful linkage. Methodological integration in a model chain requires that the data produced by one model are a meaningful input to another model, usually operating data at a different temporal and spatial scale (Liu, et al., 2008). Finally, technical integration means automating data exchanges between models, making them jointly executable, without human intervention. Technical integration aims to ensure repeatability and reproducibility of model chain runs and to optimize use of computer hardware for model simulations (e.g. batch running, grid computing, cloud computing, high performance computing).

This paper focuses on technical integration, and touches slightly on aspects of semantic integration, while leaving methodological integration aside. Technical integration can be achieved in different approaches. First, the most straightforward method is a manual and ad-hoc linking of models through output files of one model that are used as input files to another model after conversion, reformatting and transformations in, for example, spreadsheet programs through manual work. The EURURALIS project (www.eururalis.eu) is one of the many projects that apply this type of model integration. It is perfectly suitable when the integration does not have to be repeated often. Second, scripts (i.e. Python, Perl) can automate some of the manual steps in the conversion of output files to input files. Third, models are integrated in a proprietary method, where one model encapsulates the other model, making it possible to share input files and variables in memory, to implement feedback loops between models and to run models as one large monolithic model (SWAP - http://www.swap.alterra.nl/, MODFLOW – http://www.modflow.com/, WOFOST - http://www.wofost.wur.nl/). Fourth, instead of building a large monolithic model, a proprietary approach can be used to loosely link the models (IMAGE - http://themasites.pbl.nl/en/themasites/image/). In this approach, models become substitutable and the linking mechanism can be more transparent and better documented. The proprietary approach for linking the models has to be built and is specific for the models integrated or the organization that develops and maintains it. Five, instead of a proprietary approach to link models, an available and generally accepted modelling framework can be used. Such a modelling framework can be based on open standards, making it easily accessible to a larger community. In such model framework, the models are linked as independent components, exchanging data in a standardized way. Different modelling frameworks have been developed, each with their own set-up and philosophy. The choice of one of the five approaches (i.e. manual, scripting, proprietary large model, proprietary framework, open framework based on standards) to technical integration depends on the institutional and project context and on the researchers involved.

2. MODELLING FRAMEWORKS

In this paper we will focus on one particular modelling framework, OpenMI. Based on its use in several multi-disciplinary large projects that worked on integrated models, its strengths and weaknesses are investigated and recommendations for improvements made. The projects all operate in the disciplines of agriculture, land use, nitrogen cycling, forestry, hydrology and economics.

Following trends in software engineering and commercial software development and people trained in such modern methods entering the environmental science disciplines there have been several attempts to develop standardized approaches to modelling and model integration in the environmental problem domain, resulting in a number of modelling frameworks. A modelling framework brings together suites or libraries of modules with the intention to standardize features such as data manipulation and analysis, exchanges between models and data sets, structure an coding of modules, and visualization of model outputs. The term *framework* is typically used to refer only to the underlying classes and libraries, and *environments* to refer to systems that use these software frameworks to support module development, model construction, and execution (Argent, et al., 2006). A further differentiation that can be made is that between the definition of a framework, the *specification* or interface, and its *implementation*. When not further specified the term framework is used in its broadest sense in this paper.

Modelling frameworks have been conceived of and developed for some time, with a varying range of features (Argent, 2004; Jagers, 2010). Developing a framework, starting with writing its specification from a set of requirements, providing a core implementation and growing it into an environment with readily usable modules for e.g. database access, analysis and visualization, is a time consuming activity. Selecting to invest and re-use a framework therefore partly becomes a strategic decision and aspects like international recognition start to play a role. In this regard the four levels of model development and application ((i.) specific single project use, (ii.) used for a range of problems, (iii.) well documented, published use in case studies, readily available for application, (iv.) black box usage, accepted solution) defined by Argent (2004) might as well apply to the framework development and maturity. When compared to other software framework solutions, e.g. Hibernate for object-relational mapping between object classes and database tables (http://www.hibernate.org/), that can be and is widely being used in a black box fashion, model integration frameworks are clearly one or more steps away from reaching such level of maturity and acceptance.

3. THE OPEN MODELLING INTERFACE (OPENMI)

The OpenMI (http://www.openmi.org/) is a data and model integration framework, designed to take independent data and computing systems and provide a standard means of describing how time series are communicated between the systems. It has been developed from the need to answer integrated hydrological catchments management questions within the EU 5th framework program project HarmonIT (Gijsbers, et al., 2002). The OpenMI is available under the terms of the open source license LGPL (SDK implementations might apply the even less restrictive MIT open source license) with the aim of easier dissemination.

The OpenMI is under development and under the supervision of the OpenMI Association, which has as its main objectives to promote its development, use, management and maintenance. It is an entirely open international group of organizations and people, with a small core team that supports, responds to and is guided by a growing active worldwide user community.

The original version (1.4) of the OpenMI provided standardized interfaces to define, describe and transfer data between software components that run sequentially, based on a pipes and filters architecture (Gregersen, et al., 2007). After the initial release of the OpenMI version 1 in 2005 the OpenMI-LIFE project demonstrated its usability at the operational level on real world scale problems (Schade, et al., 2008), both hydrological and outside the hydrological domain (Van Ittersum, et al., 2008), integrating agricultural and economical models from farm field to world scale). The OpenMI is seeing a significant increase in the number of applications that leads to new requirements. To improve the OpenMI and advance it to the next version several European research institutes bundled activities and, after a modeling community review period, released the OpenMI standard version 2 in December 2010. Some of the major changes in the new version include better compatibility with the standards of the Open GIS Consortium (OGC), removal of limitations for working with non time-step based models, cleaning up of the interface specifications – e.g. replacing links and data operations with direct input – output references and adapters, allowing setting and varying boundary conditions for individual models, the introduction of base interfaces and optional extension interfaces to allow adaption to future requirements.

OpenMI was chosen as common platform by Alterra, Wageningen UR in research projects targeting model linking, as (i.) it is relatively lightweight, requiring few modifications to the models and works on the basis of well-defined and documented interfaces, (ii.) it is based on an open standard, which is further developed by an multi-institutional organization and which benefits from contributions from the larger research community, (iii.) it provides flexibility to implement the standard in any programming language and to extend and adapt the implementation for the specific project, (iv.) changes and additions to the standard be can proposed to the OpenMI Association for future developments and (v.) it multi-institutional organization facilitates cooperation internationally and joint development of projects with OpenMI.

4. EVALUATING OPENMI

OpenMI has been applied as a modelling framework in several multi-disciplinary large projects that worked on integrated models. Each of these projects was European Union sixth Framework Programs (EC, 2011). These projects operated in the disciplines of agriculture, land use, nitrogen cycling, forestry, hydrology and economics, and are listed in the table below.

Table 1. Characteristics of the examined projects

Project	Domains	Project Purpose	Modelling languages
SEAMLESS	Agronomy and economics	Model linking	Java, C#, GAMS
SENSOR	Land use	Results visualization	Java
EVOLTREE	Genomics and ecology	Model linking	Java, C++, Visual Basic
NITROEUROPE	Nutrient cycling and soil science	Results visualization	C#, Fortran
EFORWOOD	Forestry and forest ecology	Model definition, linking	Java
Splash!	Hydrology	Results visualization	Java, Delphi

To evaluate the use of OpenMI in these projects and assess its value for model integration undertakings two half-day workshops were organised. For one we invited six software developers that were involved in the projects and worked with the modellers on the integration of their models through the use of OpenMI. For each project we asked the developers to give a short presentation about the goal of the project, the role OpenMI played in the project, strength and weaknesses of OpenMI based on experiences in the project, and about lessons learned and suggested improvements to OpenMI.

After the presentations a discussion was held about the outcomes, particularly focussing on the strengths and weaknesses, lessons learned and suggested improvements. The information collected from this and the presentations was gathered and combined into a first concept report, which then was handed to the modellers involved in the projects and whom we invited for a second workshop.

In this second workshop five modellers were asked to respond to both the concept report and to our initial findings and conclusions, approaching it from their own perspective. For the most part it was a round-table discussion that provided extensive feedback and triggered interaction between the modellers. Typically the software developers present at the first workshop all knew each other and worked together on one or more occasions, while for the modellers it in some cases was a first time to interact about the subject.

5. RESULTS

5.1. Roles of OpenMI

Based on the feedback from the developers and the modellers gathered during the workshops two distinctive approaches were used in applying the OpenMI in the projects: (a) using models directly and modifying the model source code to make it OpenMI compliant (figure 1), and (b) building a wrapper component to provide the OpenMI compliance while leaving the model largely untouched (figure 2). Using the model directly allows for more detailed integration, but usually takes more time to implement and requires the use (in the model) of the same or at least compatible programming languages. The wrapper approach is also useful to make models comply with the OpenMI standard in regards to data definitions, exchange formats (e.g. from in-memory data exchange to file or database oriented exchange), and model execution sequences. Building generic wrappers makes it possible to re-use this often-required functionality.

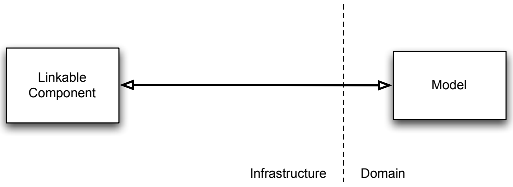

Figure 1. Option A

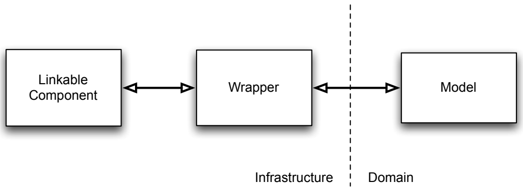

Figure 2. Option B

Option A was used by three of the projects, option B by two, and the SEAMLESS project applied both methods to integrate models. A variety of models from different domains were linked with the use of OpenMI in the projects, with a large diversity in the spatial resolution the models operated on (e.g. from farm field to world market scale), the size of the models (from EU market equilibrium model to very small wood processing steps with many feedback loops), and the programming language used for the model (Java, C#, GAMS, Visual Basic, NSM, CAPSIS – the latter two being complete modelling environments that were made OpenMI compliant).

5.2. Strengths and weaknesses

The following provides an overview of the aspects mentioned at the workshops that are considered to be a strength or benefit of using the OpenMI. Comments mentioning the same aspect were grouped and the number of projects for which the comment applied are included.

Strengths:

- Open source standardisation; provides clear and fixed definition on how things are supposed to work; potential for interoperability. (5 projects)
- Open source standardisation; provides clear and fixed definition on how things are supposed to work; potential for interoperability. (3 projects)
- Re-use existing OpenMI framework experience and knowledge. (2 projects)
- Level of available documentation. (2 projects)
- Solution for infrastructure (not a project goal to develop it). (2 projects)

Besides strengths a number of weaknesses were highlighted. These are summarized below.

Weaknesses:

- Regarded as complex or overkill by modellers. (5 projects)
- Missing (expected / promised) environment and standard shared modules, e.g. for data analysis and visualisation, multi-thread execution, programming language bridges. (5 projects)
- Missing required features: OGC support, qualitative data exchange, raster data, multi-part geometries, loops, N:M connections between inputs and outputs, support for semantic integration. (4 projects)
- Cumbersome to use, requires work-around for models outside the hydrology domain. (3 projects)

Although many strengths and weaknesses were mentioned during the two workshops some distinct clusters of comments emerged. The participants (both developers and modellers) liked the use of a standard and the benefits it offers, which are clarity in definitions, available documentation, support and samples to work from, interoperability, improved software maintenance, re-use of experience and skills. Having such an infrastructure in place is a benefit and typically not a primary goal of an integrated environmental modelling project to build it, but instead to use it. The weaknesses of the current OpenMI in this regard are that it is regarded as complex, cumbersome to use and lacking features or standard functionality (that could provide the added value to researchers working on model integration). An often-encountered misunderstanding is the difference between the OpenMI standard (the interface definitions) and the OpenMI Software Development Kit (SDK) as a reference implementation of the standard in a programming environment.

6. CONCLUSIONS AND RECOMMENDATIONS

The interviewed modellers and developers subscribe to the idea of using standards for integrative modelling and the potential benefits this presents.

However solely the definition of the standard is not enough, a full implementation that handles all the boilerplate work and provides all the right tools and components is expected. The evaluated projects all were domain content orientated and not set-up and properly budgeted to contribute to development of a generic model integration framework. With the standard specification available, some benefits are already experienced, but the standard functionality needs to be available as well.

A further cause of tension is the difference between building a model with stand-alone purpose and building a model suitable for linking to other models. The later usually is not the primary goal of the modeller working on the model, and neither of the software developer working on the framework. Making a model linkable clearly is an effort that falls between two stools. Similar tension exists between developing models and integration solutions specifically for a project versus developing them in a more generic way. Making a strong (supra-) institutional strategy is helpful, which must also be supported by strong project acquisition goals.

A team-effort between software developers and modellers to create an integrated model with use of a software framework requires the software developers to learn about the modelling and the domain context and on the other hand the modellers need to pick up some of the software engineering science and principles that they might not be accustomed to. Common best-practices of the software engineering community like using version control, issue management, unit tests, continuous integration, or even UML and (software) domain modelling can be relatively new to the modeller (Knapen, et al., 2007; Verweij, et al., 2010) and accounted to the burden of having to use the framework. Thus, the complexity of model linking in a rigorous and repeatable solution might overwhelm the modeller and software engineer when confronted with the multitude of tools and approaches from software engineering and the complex data types, large amounts of data and scaling issues from the modelling domains.

Advantages of using an existing and standards based model integration framework are (i.) the short-circuiting of a lot of discussions, commonly when working with researchers, since the standards have to be followed. The alternative is to spend extra resources on improving and adapting existing functionality. (ii.) It provides the opportunity to quickly start with integration, run the model chain and start acting on the results by improving the models and making them more suited for the integration. (iii.) The adoption of a standard proves to lead to better documentation and construction of the model making them more comparable and easier to follow for researchers from other domains. The models do not automatically become interchangeable, in practice a project tends to build a wrapper layer around each model and a layer on top of a framework for the purpose of integration and matching the framework set up to the specifics of the project (i.e. modelling languages, model scales, desired end-use). Such layers can at a later stage be used to improve the models and the framework itself.

Considering the large diversity of projects, domains (i.e. agriculture, land use, forestry) and types of models described in this paper, the OpenMI standard can meaningfully be applied outside the hydrology discipline from where it originates. Some of the extensions and adaptations required to fit project needs (i.e. more flexible data types, better support for non-time step models) and the framework weaknesses (i.e. too complex and cumbersome to use outside the hydrology domain) are already being addressed in version 2.0 of the OpenMI standard specification (Donchyts, et al., 2010), as well as in its SDK implementations, which focus more on making its usage more lightweight and less invasive for the models, e.g. by the use of annotations and introspection.

Overall, OpenMI is a useful operational solution for integrating existing and new models across domains and scales in a technical sense, often based on wrapper development and in close interaction between modellers and software engineers. Future attention has to be on developing and sharing standard functionality for use with model chains, achieving easy re-use of available OpenMI compliant models, and moving towards simple, unobtrusive and ubiquitous uses of this standard in the field of environmental modelling.

REFERENCES

Argent, R. M. (2004). An overview of model integration for environmental applications -- components, frameworks and semantics. Environmental Modelling & Software 19 (3) 219-234.

Argent, R. M., Voinov, A., Maxwell, T., Cuddy, S. M., Rahman, J. M., Seaton, S., Vertessy, R. A., and Braddock, R. D. (2006). Comparing modelling frameworks - A workshop approach. Environmental Modelling & Software 21 895-910.

Carberry, P. S., Hochman, Z., McCown, R. L., Dalgliesh, N. P., Foale, M. A., Poulton, P. L., Hargreaves, J. N. G., Hargreaves, D. M. G., Cawthray, S., Hillcoat, N., and Robertson, M. J. (2002). The FARMSCAPE approach to decision support: farmers', advisers', researchers' monitoring, simulation, communication and performance evaluation. Agricultural Systems 74 (1) 141-177.

Donchyts, G., Hummel, S., Vanecek, S., Groos, J., Harper, A., Knapen, R., Gregersen, J., Schade, P., Antonello, A., and Gijsbers, P. (2010). OpenMI 2.0 - What's New? In: Proceedings of 2010 International Congress on Environmental Modelling and Software, Ottawa, Canada.

EC (2011). CORDIS Sixth Framework Programme, http://cordis.europa.eu/fp6/dc/index.cfm?fuseaction=UserSite.FP6HomePage.

Gijsbers, P.J.A., Moore, R.V., and Tindall, C.I. (2002). HarmonIT: towards OMI, an open modelling interface and environment to harmonise European developments in water related simulation software. Hydroinformatics 2002. Fifth International Conference on Hydro- informatics: Cardiff, UK. IAHR, 7 pp.

Gregersen, J. B., Gijsbers, P. J. A., and Westen, S. J. P. (2007). OpenMI: Open modelling interface. Journal of Hydroinformatics 9 (3) 175–191.

Hinkel, J. (2008). Transdisciplinary Knowledge Integration - Cases from Integrated Assessment and Vulnerability Assessment. PhD Thesis. Wageningen University, Wageningen. 198 pp.

IPCC (2000). Special Report on Emissions Scenarios. Intergovernmental Panel on Climate Change (IPCC), UNEP and WMO: 27 pp. http://www.grida.no/climate/ipcc/spmpdf/sres-e.pdf

Jagers, B. (2010). Linking data, models and tools: an overview. International congress on environmental modelling and software, Fifth biennial meeting, Ottawa, Canada. 2010.

Jakobsen, C. H., and McLaughlin, W. J. (2004). Communication in Ecosystem Management: A Case Study of Cross-Disciplinary Integration in the Assessment Phase of the Interior Columbia Basin Ecosystem Management Project. Environmental Management 33 (5) 591-605.

Knapen, M.J.R., Verweij, P.J.F.M., and Wien, J.J.F. (2007). Applying enterprise application architectures in integrated modelling. In Oxley, L. and Kulasiri, D. (eds) MODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2007, pp. 74-80. ISBN : 978-0-9758400-4-7.

Liu, Y., Gupta, H., Springer, E., and Wagener, T. (2008). Linking science with environmental decision making: Experiences from an integrated modeling approach to supporting sustainable water resources management. Environmental Modelling & Software 23 (7) 846-858.

Schade, P., Lang, G., and Jürges, J. (2008). OpenMI Compliant Import of Initial and Boundary Data into a numerical 3D Model. In: Proceedings of the iEMSs Fourth Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSs 2008) Vol. 2, pp 1086-1093.

Scholten, H. (2008). Better Modelling Practice: an ontological perspective on multidisciplinary, model based problem solving. PhD Thesis. Wageningen University, Wageningen. 313 pp.

Sutton, M. A., Reis, S., and Bahl, K. B. (2009). Reactive nitrogen in agroecosystems: Integration with greenhouse gas interactions. Agriculture, Ecosystems & Environment 133 (3-4) 135-138.

UN (2005). Investing in Development: a practical plan to achieve the Millennium Development Goals. UN Millennium Project. United Nations: New York. http://www.unmillenniumproject.org/documents/overviewEngLowRes.pdf

Van Ittersum, M. K., Ewert, F., Heckelei, T., Wery, J., Alkan Olsson, J., Andersen, E., Bezlepkina, I., Brouwer, F., Donatelli, M., Flichman, G., Olsson, L., Rizzoli, A., van der Wal, T., Wien, J.-E., and Wolf, J. (2008). Integrated assessment of agricultural systems- a component based framework for the European Union (SEAMLESS). Agricultural Systems 96 150-165.

Verweij, P.J.F.M, Knapen, M.J.R., De Winter, W.P., Wien, J.J.F., Te Roller, J.A., Sieber, S., and Jansen, J.M.L. (2010). An IT perspective on integrated environmental modelling: The SIAT case. Ecological Modelling 221 (2010) 2167-2176.