Integration of Models for Low Carbon Economy

Getachew F. Belete\textsuperscript{a}, Alexey Voinov\textsuperscript{b}

\textsuperscript{a,b} University of Twente, Department of Geo-Information Processing, Netherlands, \texttt{a getfeleke@gmail.com, b aavoinov@gmail.com}

Abstract: Designing the transition to low carbon economy is a very complex task that touches upon a wide variety of climate-energy-economic systems. We need to explore the various possible climate mitigation scenarios at different temporal and spatial scales. However, due to the diversity of the involved disciplines it is difficult to find one complete and unified modeling approach that works equally well in all those different domains. As a result we have to select ‘appropriate’ models, which represent only specific aspects of the scenarios and assemble them ‘coherently’. In this research we have identified some challenges in integrating multidisciplinary models; and have developed a conceptual design for a multidisciplinary model integration framework that can harmonize the technical, semantic, and dataset aspects of interoperability.

Keywords: integration; model; semantic; dataset; framework.

1. INTRODUCTION

A wide variety of computer-based models and tools have been developed to describe and characterize different systems but for complex problems “no single application is available nor is it advisable to develop one for practical reasons. Integration of models and tools may be a solution” (Stoorvogel 1995). Similarly for interdisciplinary assessment such as needed in studying the transition to a low carbon economy, integrating models from different disciplines is necessary since no single disciplinary model can cover all the relevant processes and systems involved, including analysis of climate, energy, hydrology, land use, economic development, human behavior, resource allocation, etc.. Another advantage of the integration approach over using one integral model is that it gives the promise of reuse of existing legacy models and the knowledge contained in them. “Integration is faster and far less expensive than reengineering”(Madni & Sievers 2013) legacy systems. However this doesn’t belittle situations in which (no matter how time taking and expensive) we prefer the benefits of reengineering through improved implementation.

Various model integration techniques and frameworks already exist. However, for analysis of complex systems such as those involved in climate mitigation research or low carbon transitions the integration technique should go beyond software-based component coupling. We need to develop an integration framework in which stakeholders and scientists can explore various climate mitigation scenarios using different models and different model types, including conceptual and quantitative models that may not be immediately implemented as software. We should create reliable semantic and conceptual understanding among models, modelers, and stakeholders. Our integration approach should accommodate both existing and new models to work in concert and support transdisciplinary studies.
Here we are identifying the most crucial integration points at different levels and are proposing a conceptual design of an integration framework that can serve the needs of such transdisciplinary research.

2. CHALLENGES AND ASPECTS OF MULTIDISCIPLINARY MODEL INTEGRATION

For analysis of low carbon economy scenarios we need to link models, which come from different disciplines, may be both qualitative and quantitative, may have different temporal and spatial scales, and are built using different technologies, under different paradigms and assumptions. The linking mechanism should be able to execute models in a transparent way, since the output needs to be understood, trusted, and used by a community of researchers and stakeholders, and not only by the developers.

Currently a number of model integration techniques, frameworks, and standards exist (Armstrong et al. 1999; Gregersen et al. 2007; Peckham et al. 2013). A common problem of many of model integration techniques is that they are mostly concerned with the software side of integration, where “models are merely treated as software components that are to be made to work together and talk to each other” (Voinov & Cerco 2010). However, integrating models of various types and from different disciplines requires more than that. The first strategy we follow is to divide the integration effort into three different levels; namely technical, semantic, and dataset integration levels (Knappen et al. 2013).

Technical integration of models stands for “automating data exchanges between models, making them jointly executable, without human intervention” (Knappen et al. 2013) and is usually resolved in the software engineering level. It tries to ensure repeatability and reproducibility of model chain runs and to optimize the use of computer hardware for model simulations. Johannesson et al. (2000) point out that for technical integration of models data-oriented, application interface-oriented, method-oriented, portal-oriented and process-oriented approaches have been used and turned to be helpful. Knappen et al. (2013) suggest that technical integration of models can be achieved in five different approaches: soft linking, using scripts, proprietary monolithic, proprietary loosely linking, and with open standards.

In building integration frameworks technical integration is commonly achieved mainly by implementing some standards on model interfaces (Brown et al. 2002; Janssen 2009; Peckham et al. 2013). Then, once a model implements such a standard, we can consider a computer-based model as a composition of two major parts: (1) the interface which defines the inputs, outputs and parameters of a model; and (2) implementation which defines the model equations (Athanasiadis et al. 2011) and intentions. For implementing interoperability standards, sometimes for small or simple models rewriting the model with the preferred language and with a particular interface may be feasible but for large models we should not do that because: (1) it is time consuming; (2) error prone; (3) rewritten version of the model will not see the benefits of future improved versions of the original model; (4) code reuse is one of best practices of software engineering. However “treating models only as software in solving the integration challenge may give birth to ‘integronsters’ - constructs that are perfectly valid as software products but ugly or even useless as models” (Voinov & Shugart 2013).

One possible way to safeguard us from ‘integronsters’ is semantic mediation, which is the process of enforcing and supporting semantic consistency along data flow paths (Villa et al. 2009). Models have underlying concepts; and integration of models should be done only among those models that have matching input-output flows that can be conceptually linked to each other. Scholten (2008) describes the importance of semantic integration of models as “means of speaking a common language and achieving a shared understanding between all models and modelers working together”. This semantic mediation will be based on a common vocabulary presented in an ontology that formally describes
concepts involved in the messages that models exchange and defines the relationships among those concepts (Rizzoli et al. 2008).

Regarding the scope of ontology, only the specification of the interfaces between the models have to adhere to the shared ontology - the knowledge in the models themselves does not have (Athanasiadis et al. 2006). This can be achieved by specifying ontology for model parameters, inputs, outputs, and states. Otherwise the scope of semantic integration work could be unmanageable for interdisciplinary models. Depending on the participating models and the complexity of the message the semantic mediation can be either simple meaning mapping or something that requires complex reasoning. This can also apply to conceptual models developed by stakeholders. We should also be able to check their semantics with respect to the chosen ontologies to make sure that they remain consistent with the other models that may be relevant to the study.

Another method to avoid ‘integronsters’ is checking the consistency of datasets exchanged between models. A dataset is a collection of related data, which is composed of separate elements. Dataset integration of models is the process of converting “data produced by one model to become a meaningful input to another model, usually operating at a different temporal and spatial scale” (Knapen et al. 2013). Such task requires executing such operations as aggregation, conversion, projection, estimation, scaling, etc. There is no generic dataset integration technique that can convert the output of one model into input for another model. In integrated assessment, for example, this process is mainly concerned with statistical up- and down-scaling techniques to move data from a lower scale level to a higher one and vice versa (Rotmans 2002). Regarding selecting the appropriate scaling methodology Ewert et al. (2006) underline the importance of the research question; the behavior of the system; the understanding of the underlying processes, mechanisms and their interactions; and the availability of data.

Generally there is no single integration technique, which is best suited to all kinds of integration requirements. In some cases manual integration of models using ad-hoc soft-linking techniques or using some scripts for data conversion may be appropriate. In situations when we have large libraries of models and methods to choose from when building and integrated application, or when a two-way communication between models is required at each time step an integration framework that facilitates and checks the integration process of models could be a solution.

3. POINTS TO CONSIDER IN DESIGNING A TRANSDISCIPLINARY MODEL INTEGRATION FRAMEWORK

In designing a transdisciplinary model integration framework the following features become especially important.

a) Componentization and Structural Granularity

A component is an application developed for a clear purpose and that has a standard interface, which enables to utilize it even without knowing the internal implementation of the application. Models developed as components have the advantages of reusability, limited dependency on other components, interoperability, dynamic linking to other components at runtime, etc. (Peckham et al. 2013; Rizzoli et al. 2008). As mentioned above, for complex models componentization can be achieved by developing wrappers that serve as a bridge and gateway to the underlying model and handles every communication between the model and its users.

However this also means that we might be creating considerable redundancy and component overlap, when the same functional elements are present as parts of different components (Polhill & Gotts 2009). This in turn may result in ambiguities if these functional elements are using different
assumptions, data, and equations, but do not report this properly in the documentation available. For example, various components (models) may be calculating certain climatic variables (say amount of rainfall) to use them in other model functions (say, agricultural yield, forest biomass, grassland productivity, etc.). When integrating these components (say to calculate the total economic output of biomass producing sector) we will be in fact running this climate generator inside several components, not necessarily realizing that. The choice of the most appropriate structural part of a model that should be componentized remains a problem that may be difficult to solve, especially if we do not want to rewrite much legacy model code. Due to this avoiding subcomponent redundancy may be difficult but alerting the user about the occurrence of subcomponent redundancy can be done from meta-model information and semantic representation of the models.

b) Meta-model Annotation

To build semantic linking and dataset conversion among models the meta-models of participating models play a crucial role in understanding the participating concepts. Kühn et al. (2003) underline that for integration of models that describe interdisciplinary concepts “a necessary prerequisite is the integration of their underlying meta-models”. We need a strategy that enables models to express themselves to other participating models. Before the wrappers can make the model information available to other models, first we need to make sure that participating models are explained and described according to a generally accepted and understood meta-model standard. Designing such standards is only part of the challenge; enforcing them will be much harder. Model documentation has always been quite poor and in many cases inadequate. Developing incentives and requirements for standardized model documentation is much overdue and becomes especially important for model integration purposes.

c) Loosely-Coupled Communication

Coupling is relative and unavoidable when we assemble components to build a system. Loose coupling is a method in which components of a system have little or limited dependency among them. If the parts are over-tighten we compromise flexibility, and if we do not tighten enough the system it will lose robustness; so we have to maintain the balance by establishing the most appropriate coupling level among the parts (Erl 2008b). For example: the dependency among components can be minimized by establishing message based communication rather than direct point-to-point communication. However message based coupling results in larger computational overhead and is less efficient when complicated models exchange large amounts of information, in time and space. Hardly a “one-size-fits-all” approach can be recommended here. The choice of the coupling method will have to take into account the particular features of the models that are integrated.

d) Data Transformation

The data transformation task includes both semantic mediation and dataset conversion. In some of the cases the semantic mediation can be addressed by using variable mapping, i.e. when simple mapping of a variable is enough to create shared understanding between models. But in most of the cases ontology will be used for semantic mediation since ontology is commonly used as inter-lingua for semantic understanding among applications (Uschold & Gruninger 1996). On the other hand, the dataset integration process requires understanding the context of the specific environment and acting on it. Sometimes the task may also require action while having only incomplete knowledge, and with various levels of uncertainty. All such contextual based data transformation tasks should be facilitated by the accompanying meta-model information of the participating models.

e) Design-time and Run-time Composition

We distinguish two stages in the integration process. First at the design stage the semantic and dataset integration agents are defined to check the logic and consistency of the proposed model links.
While doing that they formulate the integration rule that is then applied during the run-time stage. This integration rule makes sure that the output from the joint execution of various models is meaningful (Gabrilovich & Finkelstein 2001). This is achieved by: (1) applying the proper data set conversions and consistency checks; and (2) by checking the semantic linking of models and identifying and maintaining a list of valid input-output relationships among models. The rule mainly defines with which models and how a certain model integrate while still generating scientifically valid output. During runtime when a model makes a call or requests a service from another model, the content and semantics of these calls are adjusted and corrected according to the integration rule formulated during the design time. The defined relationships, message interpretation and context based amendment on the message is done before the message crosses the boundary of another model.

4. CONCEPTUAL DESIGN OF MULTIDISCIPLINARY MODEL INTEGRATION FRAMEWORK

In developing a high-level conceptual design of multidisciplinary model integration framework (Fig 1) we match the aspects of integration defined in section 2 with the required features in section 3. We propose to base the framework on the Service Oriented Architecture (SOA), which is a design principle that assumes a standardized service contract, loosely coupled communication, flexible service composability, and reusability (Erl 2008a; Erl et al. 2009; Josuttis 2007). See more details on the architecture in Belete et al. (2014).

![High-level conceptual design of multidisciplinary model integration framework](image)

As discussed above the minimum requirements for a model wrapper are as follows:

- It should convert a model into a plug-and-play component;
- It should not be constrained to one programming language, models wrapped using different languages should not require language interoperability to communicate with each other;
- It should expose meta-model information for semantic checks and data transformation.

A promising technology that can meet these requirements are web services. A web service is a component which can be accessed by other programs over the web and which provides standardized machine-readable metadata information about available functionalities, input-output, and messaging format for communicating it. Web services are language interoperable and loosely coupled. When we link service-based models the service consumer model should have to maintain the service name, list of service parameters, and acceptable messaging format of the service provider model. The number
of participating models together with location information of each model can vary from time to time. To avoid information and effort duplication, and to facilitate loosely coupled communication among wrapped models technical integration agent is needed; which itself is a web service.

The technical integration agent is still the core of the integration process and it is dedicated to routing and messaging among service-based models. However now the message is amended by the information provided by the semantic mediation and dataset integration agents, which define the functionality that is needed to allow models to "talk to each other". Before routing a message to the next model, we apply the integration rule defined by the semantic mediation and dataset conversion requirements. Consider the case shown in Fig 2. The message received at stage 1 should be semantically amended as in stage 2 based on the semantic relationship identified during design time. At stage 3 the dataset should be organized in a format, which will be utilized by the next model. But assigning the two tasks to the technical integration agent will make it monolithic and responsible to a number of unrelated tasks. Implementing separate concerns in different functions helps to have independent functions that work together towards achieving a common goal. Being independent functions the development, maintenance, failure and success of each section will be more manageable. As a result separate semantic and dataset integration agents are identified.

<table>
<thead>
<tr>
<th>Stage 1</th>
<th>Stage 2</th>
<th>Stage 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output message from Model 1 where a, b, and c are variable names</td>
<td>Suppose from the semantic network we know that a=d, b=e, c=f, and Medium= Average, then we modify the message to</td>
<td>We have to rearrange the order of message content in a way Model 2 expects</td>
</tr>
<tr>
<td>&lt;a&gt;100&lt;/a&gt; &lt;b&gt;Medium&lt;/b&gt; &lt;c&gt;300&lt;/c&gt;</td>
<td>&lt;d&gt;100&lt;/d&gt; &lt;e&gt;Average&lt;/e&gt; &lt;f&gt;300&lt;/f&gt;</td>
<td>&lt;e&gt;Average&lt;/e&gt; &lt;d&gt;100&lt;/d&gt; &lt;f&gt;300&lt;/f&gt;</td>
</tr>
</tbody>
</table>

Fig 2: Data transformation process

Generally, when we link models using the framework the integration process can be described by the following workflow:

- The technical integration agent will call on the semantic mediator to handle the semantic interpretation using available meta-model information and model interface ontology. If the rate of greenhouse gases production is named as ‘Rate’ by model A and ‘Emission’ by model B the semantic mediator is expected to match the two variables using the semantic relationship defined in the ontology. After semantic mediation the message will be passed to dataset integration agent for dataset integration.
- The dataset integration agent will identify the type of dataset conversion needed and will return the information directly to the technical integration agent which will formulate the integration rule to be applied at run-time.
- Finally at run-time, the technical integration agent will collect the message from one model, amend it according to the integration rule and pass it on to the next model in the model chain.

5. CONCLUSION AND DISCUSSION

Investigation of ways of transition to low carbon economy requires collaboration in several disciplines such as hydrology, climate, energy, economy, behavioral science, land use, policy, etc. Therefore for analysis of transition scenarios we need to integrate models, which are interdisciplinary, may be
conceptual, qualitative and quantitative, may have different temporal and spatial scales, may be built using different technologies, under different paradigms and assumptions, etc. However integration of models has technical, semantic, and dataset aspects of integration, which goes beyond linking software components.

Although manual ad-hoc technique has been in use to solve several integration challenges, integration frameworks can facilitate to address integration requirements that need dynamic two-way communication of models at different time steps. Componentization of models for interoperability, establishing limited dependency among models via loosely-coupled communication, flexible run-time composition of different componentized models, meta-model annotation for contextual description of models, and data transformation are the main characteristics that needs to be considered in designing multidisciplinary model integration framework. We have also shown how all these required features can be attained in a framework using SOA design approach. Our next step is implementation of the framework in a case study to figure out unforeseen difficulties, which will be used to improve the design of the framework.

Besides those identified high-level features during design and implementation of the framework, usability should be emphasized. Our users can be scientists, decision makers, and stakeholders with various backgrounds and goals. Availing meta-model information about each model, query capability to navigate alternatives models in the framework, and providing user-defined flexible linking of models can improve the usability of the framework. Participation of model developers in documenting meta-models, and participation of users in evaluating the system becomes quite crucial to improve usability and acceptance of the approach.

ACKNOWLEDGMENTS

G. F. Belete and A. Voinov was supported by COMPLEX – Knowledge Based Climate Mitigation Systems for a Low Carbon Economy Project, EU 7th Framework Program, Theme [env.2012.6.1-2], Grant agreement 308601.

REFERENCE


Erl T (2008a) SOA design patterns Pearson Education.


Belete G.F. et al. Integration of Models for Low Carbon Economy