Jul 1st, 12:00 AM

Integronsters' and the special role of data

Alexey Voinov

'Integronsters' and the special role of data

Alexey Voinov

University of Twente Faculty of Geo-information Science and Earth Observation (ITC)
P.O. Box 6, 7500 AA, aavoinov@gmail.com

Abstract: In many cases model integration treats models as software components only, ignoring the fluid relationship between models and reality, the evolving nature of models and their constant modification and re-calibration. As a result, with integrated models we find increased complexity, where changes that used to impact only relatively contained models of subsystems, now propagate throughout the whole integrated system. This makes it harder to keep the overall complexity under control and, in a way, defeats the purpose of modularity, when efficiency is supposed to be gained from independent development of modules. 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 and useless as models. We argue that one possible remedy is to learn to use data as modules and integrate them into the models. Then the data that are available for module calibration can serve as an intermediate linkage tool, sitting between modules and providing a module-independent baseline dynamics, which is then incremented when scenarios are to be run. In this case it is not the model output that is directed into the next model input, but model output is presented as a variation around the baseline trajectory, and it is this variation that is then fed into the next module down the chain. The Chesapeake Bay Program suite of models is used to illustrate these problems and the possible solutions.

Keywords: Modularity; Calibration; Integrated modeling; Chesapeake Bay; Module linking; Components

1. INTRODUCTION

As our impacts on the environment become more dramatic more interest is drawn to analysis of systems that span over several traditional scientific disciplines. These systems are more complex and have to be described by models that may have various components characterized by different scales, resolutions, and developed under different assumptions and paradigms coming from different scientific traditions and backgrounds. Integrated modeling is the method that is developing to bring together diverse types information, theories and data originating from scientific areas that are different not just because they study different objects and systems, but because they are doing that in very different ways, using different languages, assumptions, scales and techniques.

There may be two ways of doing integrated modeling. One is to build the model as a whole. Here the modeling team collects data and information from various scientific fields, processes it, and translates it into one formalism. This is how some of the well known integrated models were built. For example, the Club of Rome models resulted in the famous World 3 model by Meadows et al. (1979), which integrates information about agriculture and food production, industry, demographics, non-renewable resources, and pollution (http://www.whole-systems.org/world3.html). Another example is the Global Unified Metamodel of the Biosphere (GUMBO) (Boumans et al., 2002, http://ecoinformatics.uvm.edu/projects/the-gumbo-model.html). Here knowledge about geology, global climate, sociology, economics, atmospheric processes, and ecosystems is integrated in one model, designed to run various scenarios of global change. A significant feature of these models is that they are developed and maintained by the same team as a whole.

The other approach to integrated modeling is to put together already built models. For example, the modeling suite developed by the Chesapeake Bay Program (Cerco, 2000; Linker et al., 2000; Wang and Johnson, 2000) consists of three major parts:

- The atmospheric transport model that produces atmospheric deposition predictions for nutrients and other constituents. The Airshed Model is based on the Community Multi-Scale Air Quality modeling system (CMAQ). The latest CMAQ (2009) code runs on a 12 km fine grid in the Chesapeake region, with a 36 km grid used for the continental scale boundary conditions;
- The watershed model, a highly-modified version of HSPF (Linker et al. 2000, Bicknell et al. 1996), that produces loadings that come from the land into the estuaries;
- The estuary model, which is, itself, a combination of three linked models, a hydrodynamic model (Johnson et al. 1993), a eutrophication model (Cerco and Cole 1993), and a sediment diagenesis model (DiToro 2001). This Water Quality and Sediment Transport Model (WQSTM) is a three dimensional model of the tidal Bay comprised of 57,000 cells. Currently it represents transport processes, eutrophication processes, and living resources such as submerged aquatic vegetation and benthos.

This modeling system has been developed over the past 25 years and went through many phases. The models are linked loosely, so that is there is no formal software mechanism involved. Mostly output from one model is sent as input into the other model as a data file. Yet still, the models work in concert and are used for decision making purposes as a suite. Here integration is performed at the level of existing models that are taken from different fields of science as modules.

Since there are already numerous legacy models that have been carefully designed and tested in numerous applications by skilled researchers who are specialists in their corresponding fields of science, it makes perfect sense to try to reuse their products as building blocks for more complex systems. We only need to make sure that they can be linked together in a meaningful way matching the variables, scales and resolutions. Currently there are several efforts to develop the standards and software tools that would provide for this kind of integration. For these purposes, models are merely treated as software components that are to be made to work together and talk to each other.

The two approached to integrated modeling bear their own caveats and complications, which we consider below.

2.   INTEGRATED MODELS

Building integrated models is a rather conventional approach as we can see from their history. In most cases such models were built by integrating knowledge from different fields of science in the form of data, concepts, functions, approximations, etc. The models themselves were built from scratch and little care was given to future reuse of the pieces that went into the model formulation. The models were not designed for reuse or modification by others outside of the team.

At some point it became clear that the model components could be reused or modified for future applications, and models started to be packaged as modeling systems. The Modular Modeling System (MMS, Leavesley, et al., 1996) was one of the first attempts to make modules available for reuse. Another example of modular modeling is the Library of Hydro-Ecological Models (LHEM, Voinov et al., 2004). Here modules were developed as a mix of Stella and C++ code, together with the Spatial Modeling Environment (SME) used to stitch the modules together and take care of all the input and output.

A more recent and sophisticated development is the Object Modeling System (OMS) by the US Department of Agriculture (David, et al., 2002; Kralisch, et al., 2004; Ahuja, et al., 2005). OMS requires rewriting modules in Java or C# to be then inserted into the system library.

Another framework in this same category is the MIMOSA developed by CIRAD (Müller, 2009, http://mimosa.sourceforge.net/). It is a modeling and simulation platform, which can

support the whole model building process, starting from the stage of conceptual models and up to running and analyzing simulations. The specification uses ontologies and an extensible set of formalisms to present system dynamics, initialize and visualize the model. The simulation kernel is based on DEVS principles (Discrete Event System Specification; Zeigler, 1976).

There are also proprietary platforms that bear promise in supporting integrated modeling such as Extend and Simile.

Note that all these frameworks are not really designed to integrate legacy code as is, but rather they serve as tools for future model development, offering input, output and project support tools and common library standards, into which different modules are invited to be contributed. The future of these frameworks will very much depend upon how wide their standards will be accepted within the modeling community and whether there will appear a critical mass of contributed models. So far, unfortunately, there are very few examples of frameworks being used beyond their respective development teams. Since each module has to be rewritten in the language of the framework, there is less worry about the consistency of models that these frameworks are producing. All the scaling and linkage issues are expected to be solved when modules are tailored, adjusted and added to the framework libraries.

3. INTEGRATING MODELS

The alternative approach is to support model integration at the level of existing models, allowing these models to talk to each other directly.

Some of the most advanced attempts of such model integration come from physics and engineering. For example the Common Component Architecture (CCA) is developed by the Department of Energy and Lawrence Livermore National Lab teams (Bernholdt, 2004). The design places minimal requirements on components and facilitates the integration of existing legacy code into the CCA environment by means of the Babel (2004) language interoperability tool, which currently supports C, C++, Fortran 77, Fortran 90/95, and Python. The CCA is being applied in a variety of disciplines, including combustion research, global climate simulation, and computational chemistry. At this time CCA is adopted as the integration engine by the Community Surface Dynamics Modeling System (CSDMS; Peckham, 2010).

The US Environmental Protection Agency (EPA) has been developing the FRAMES (2009, 2009a) (Framework for Risk Analysis in Multi-media Environmental Systems) system to manage the execution and data flow among the science modules. 3MRA (2009) (Multi-media, Multi-pathway, Multi-receptor Risk Analysis) (Babendreier and Castleton, 2005) is a collection of 17 modules that describe the release, fate and transport, exposure, and risk associated with various contaminants. FRAMES was developed as the framework that would allow these modules to communicate with each other.

The Open Modeling Interface and Environment (OpenMI, 2009) developed by a consortium of European universities and private companies, is a standard for model linkage in the water domain (Moore, et al., 2005). The OpenMI standard defines an interface that allows time-dependent models to exchange data at runtime. When the standard is implemented, existing models can be run in parallel and share information at each time-step.

All these systems are primarily about software issues and the major concern is how to link models as software components. There always seems to be a software solution that we can find, and the promise is that we will be able to make models work together and exchange information at run time. Actually, models are more than just software and there are issues that may not be easy to deal with at the software level. This takes root from the long lasting discussion about modeling as science or art. Let me try to explain what I mean by using some metaphors.

3.1 "Ugly" constructs

Fig. 1A. Disney's Ariel

There is certainly some beauty in the most well-known models from physics. Agren and Bosatta (1990) have called for ecologists to use similar esthetic criteria to guide them towards useful theories in their domains. They asked: "Are we like construction engineers who mainly care about getting the house built, or like artists who try to capture the essence of the house in our paintings?" (p.213). I would argue that similar principles should apply to model coupling. In Fig.1A we can see an example of a somewhat esthetically pleasing effort in module integration. Coupling modules in a different way can produce results that are way less attractive for an average eye (Fig.1B). It may be hard to explain why one coupling is nicer than the other one: technically the linkage process in both cases has followed quite similar rules. By following only technical principles we may be giving birth to 'integronsters' - constructs that are perfectly valid as software products but ugly and useless as models.
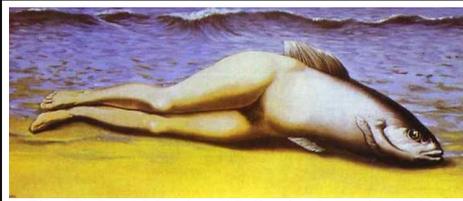

Fig. 1B. René Magritte. "Collective Invention" 1934

3.2 Skewed geometry

Geometry and spatial resolution of a system is very important. When building a model we tend to imply much about the spatial representation that we use. In many cases the system boundaries are not quite carefully spelled out and the spatial resolution and system topology are assumed as a given. This works nicely as long as the modelers themselves are the ones who use the model. If the model is packaged as a module and offered for reuse and recombination, all these hidden and unsaid features become crucial. We can imagine a geometrically consistent system as in Fig.2A, combined with another geometrically consistent system as in Fig. 2B. The result of this integration shown in Fig. 2C makes little sense in our normal Euclidian space. Even while the connection was made in a reasonable way: a lower leftmost column was connected to the upper leftmost column and so on. The overall orientation of the object may have not been part of the module description. The result, while technically correct, becomes meaningless.
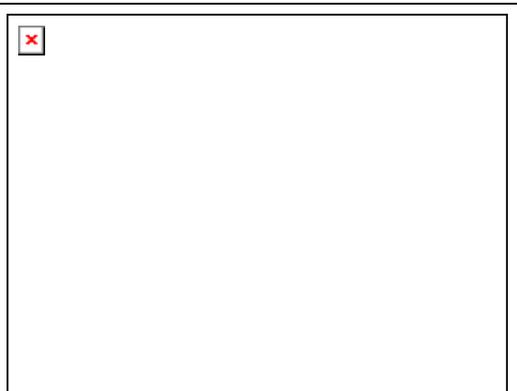

Fig.2. M.C. Escher's Belvedere, 1958, made of two geometrically consistent components twisted in space when put together.

3.3 Mismatched scales

In environmental sciences we find various combinations of processes and systems, and each one is characterized by its own specific scale and resolution in space, time and complexity. While linking in time and space is mostly a technical problem and can be resolved by appropriate software and documentation, it is less clear how to communicate information from one complexity level to another, how to estimate system responses across scales or levels. The problem is that each model is built for its particular goals and these goals and scales are also specific for the field of application and discipline. For example, economic models mostly operate in terms of global equilibria, which may remain unchanged for months or even years. They also deal with variables that may be averaged across regional scales. It would be inappropriate to blame the economists that their models do not get even close to the resolution and complexity of, say, hydrologic models. If we need to make these two types of models work in concert as components, we find that the complexity of these components is in different dimensions, which would

Fig. 3. Bull in a china shop

be hard to compare. In a way one model component in this case can be compared to a bull in a china shop: it simply does not belong there. Yet for a transdisciplinary system analysis, we need to integrate the two modules together.

There have been many attempts to couple models from different levels of organization ranging, for example, from leaf to ecosystem (Anderson et al., 2003). One approach connecting such models is to extrapolate or average the results obtained from a detailed level to a higher level (Ewert et al., 2006). In other cases we may want to focus on the critical thresholds, when the more detailed model will be analyzed beforehand for conditions that can be like a red flag for the rest of the system. For example, the hydrology model may be needed only to generate the extreme, flood events, which are important for the economy model. By estimating various important relationships from model simulations at lower levels, and then using the derived parameters as inputs for higher-level models, we can reduce the computation time. However, we may be missing some complex interactions and feedback mechanisms (Ewert et al., 2006).

3.4 Confusion of tongues

Different scientific disciplines speak different languages. This makes it only harder to integrate the models that they produced. There is much hope that the ontological approach (e.g. Hřebíček, Kisza, 2008; Rizzoli, et al., 2008; Athanasiadis, et al., 2008; Janssen et al., 2008; http://www.apesimulator.org/OntologyBrowser.aspx) can help to find a common way for communicating information. But ontologies emanating from different scientific fields are likely to be different and additional effort will be needed to synchronize them. Otherwise we are still likely to experience the problems described in the story about the Confusion of Tongues (Fig.4). This transdisciplinary synchronization of ontologies is hard to automate and standardize, especially when we are trying to use legacy models.
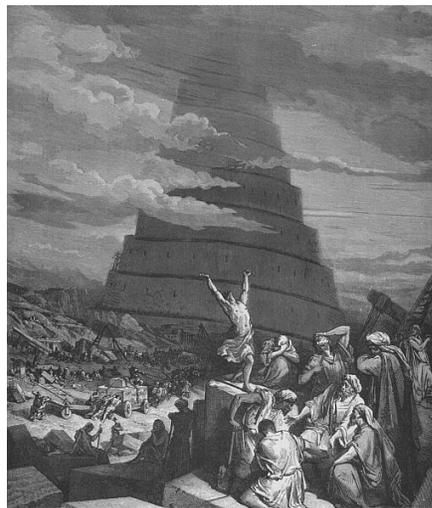

Fig. 4. Gustave Doré's "Confusion of Tongues".

3.5 Do it together

Most of the above listed problems can be resolved if the modeling process is conducted as a community effort (Voinov et al., 2010). A community for model building, such as the CIEM (Community for Integrated Environmental Modeling - http://groups.google.com/group/commiem), which is based on open source and follows some principles similar to the ones developed by OGC (Dibner, Arctur, 2008), is certainly a very promising endeavor.

According to wiki-answers it is Geoffrey Haley who said that "When a collection of brilliant minds, hearts and talents come together ... expect a masterpiece". This may be true but it is still hard to imagine how a painting like Mona Lisa could be painted by Leonardo working together with Michelangelo and Raphael in a community effort. More likely they would have produced some kind of a modular product as in Fig.5. Do we really expect masterpieces from community efforts or rather they should be focusing on routine work, serving the needs and goals of particular applications?

We see that model integration and integrated modeling may need efforts that go beyond software development and require research in community building, social networking, semantics and modeling methodology.

4. MODEL INTEGRATION AND DATA

Fig.5. Mona Lisa as a product of community efforts.
(http://www.walyou.com/blog/2010/04/27/18-greatest-mona-lisa-painting-remakes/)

Another important issue that distinguishes models from mere software is how they relate to data. This is especially evident for environmental and socio-economic applications, where in fact models are rarely based on sound theory and mostly represent empirical generalizations. Even in hydrology, where there is a solid theoretical background, the calibration phase turns out to be essential. No matter how undesirable calibration is, especially when we need to run scenarios for global change and explore the domains that are not well covered by data, using models without calibration is risky, to say the least. Jetten et al. (1999) compared the performance of some 14 catchment models and found that uncalibrated use of models is not advisable, especially for small and medium scale catchments. Similar conclusions were drawn by Bormann et al. (2007): when calculating scenarios that were out of the calibration domain, different models produced quite different results.

The promise of module integration has been most prevalent for process-based models. The idea is that by representing the processes correctly in a model, we can then couple models of these processes together and produce simulations of more complex systems, where processes are linked and exchange material and information. This becomes less attractive, if we need to deal with empirical models, which are always dependent on data. The problem is that in environmental sciences there are hardly any really process-based models.

What is a truly process-based model as against an empirical, regression one? In any process-based model we actually utilize certain empirical generalizations, rather than true process description. For example, there is hardly an adequate detailed biophysical molecular description of the photosynthesis process to be found among the models of vegetation growth, instead some variations of Michaelis-Menten kinetics are applied, which are already empirical generalizations of the process. Nevertheless these models claim to be process-based. As we go to larger systems, such as landscapes, we will need to employ even more generalized formalizations (Voinov, et al., 1998). We tend to forget that what we assume to be descriptions of processes are in fact only empirical generalizations of these processes. For almost all so-called process-based models, they actually describe processes only at a certain level of abstraction, and become empirical beyond that. This is probably why most of the process-based models still need to be recalibrated when applied to new areas and study cases. The more different the environmental conditions – the more recalibration needed.

Silberstein (2006) claims that "modellers who focus on their model without continual reference to real data are not really scientists but artists. They have their place, and indeed their ideas may well turn out to be useful, but their activities are not science if they are not base on observations". But if calibration is still an essential task when integrating models, the promise of building new models of complex systems from modules, like from Lego blocks, becomes less feasible. If we still need to do calibration, we need to remember that calibration of complex models is even more complex. As more components are brought together, the calibration of the whole model becomes only more difficult.

An integrated model is made out of two or more relatively independent components. Each of these components can operate on its own and in many cases has been developed

independently by separate groups of researchers. The promise of integration in this case is that legacy code and models can be reused to analyze more complex systems, while analysis is simplified since the overall system can be studied and modeled in portions. Suppose we have model A for one sub-system and model B - for another. If a system is a composite of these two sub-systems then instead of building a whole new model C to present it, it should be possible to use an integration of the two existing models and model the system with an AB model, where the two components A and B exchange information as they run. For simplicity let us assume that there is only a one-way flow of information from A to B.

What should be the calibration process in this case? If A and B existed before, they were most likely calibrated previously with observation data. If not - still it should be much easier to calibrate smaller and simpler components, therefore we may assume that A and B are calibrated separately. However, obviously, after integrating A and B we should not expect that the resulting output from the AB model will match data as well as output from B. Note that for our system it is the output from B that matters, since B generates output for the integrated model AB.

The reason that AB will generate somewhat different results than the calibrated standalone B model is that now the forcings for B come from the calibrated model A, instead of being taken from data. No matter how well A is calibrated, its output is likely to deviate from the data, and will steer AB away from what B previously was generating. The results can be still improved by some further calibration of B as part of the integrated system. In this case this will look like a calibration of the AB model as a whole. This will also likely include some refinement of overall model performance by model B calibration, in fact, compensating for some deficiencies in the calibration of model A. Moreover, we may be even tempted to further tweak some of the parameters in A to get a better match. This is exactly what is done in integrated models: they need adjustment and re-calibration after the components are put together. However by doing this we lose much of the advantages of the modular architecture in the integrated AB model. We now need to deal with the calibration of the full model, and whenever any changes are made to any of the components, the other components need to undergo new calibration as well.

For example, for the Chesapeake Bay decisions are mostly based on predictions for the future state of the system in terms of such indicators as the area of hypoxia, or suitability of habitat for living resources, while most of the management and mitigation is carried out for the watershed, where the nutrient and solids loads are generated. Therefore the integration of the three models is crucial. But whenever the watershed model gets updated, it produces different output. As a result, every time the watershed model is changed, the estuary model needs to be re-calibrated. In theory, as the watershed model moves from one phase to the next one, its output is supposed to become 'better' in terms of matching the data, the model is supposed to be better calibrated. Sometimes this is indeed the case, in other cases the upgrades are driven by the need to include new features important for management, or by the need to change the resolution in the model. The output in this case becomes 'different' but not necessarily 'better' in terms of a better fit to data. Again and again the estuary model needs to undergo tedious re-calibration, which is entirely caused by developments that are external to it, and do not necessarily have anything to do with improvements of the estuary model itself. Much effort and time is spent with the only benefit of keeping the components working together.

Voinov and Cerco (2010) point out that the fact that two components are integrated into one model, does not have to make the available calibration data no longer relevant. When the downstream model, the estuary model in our example, was first calibrated, it was forced with observed loads, and there is no reason to think that the output from the up-stream model, the watershed model, is more accurate than these observations. If the data was still embedded in the integration process, then there would be no need for model re-calibration every time modifications are made 'up the modeling stream'. Indeed, when new and 'better' modules are developed, the data are still invariable, and there is no reason why the same data set that was used for calibration of an up-stream model A cannot be used as forcing functions for the 'down-stream' model B.

When integrating models, the data can stand between the output of the upstream model and the input of the downstream model (Fig.6). When the first model is modified, the second model does not need to be re-calibrated, because still the first model is expected to represent the same data as before, and it is that data that feeds into the second model. In a way we employ a version of data assimilation, when data become part of the overall integrated modeling structure. Certainly, in many cases the output of the up-stream component is much more data-rich than what the observed data sets offer. For example, in case of the Bay model, the watershed model generates inputs for every single tributary, while data sets are available only for some. Yet still that is not a reason to exclude the available observations and replace them entirely by 'artificial' data. Wherever and whenever observed data are available they should be used and should have precedence over model generated data. What we suggest is a data-model fusion when integrating components.

But how do the two model components run in concert, if the data component sitting between them is invariant to the changes in the forcing functions that drive the upstream components? After all, most models are built to run scenarios, where certain parameters and forcing functions are modified to answer the 'what if' questions or make predictions for
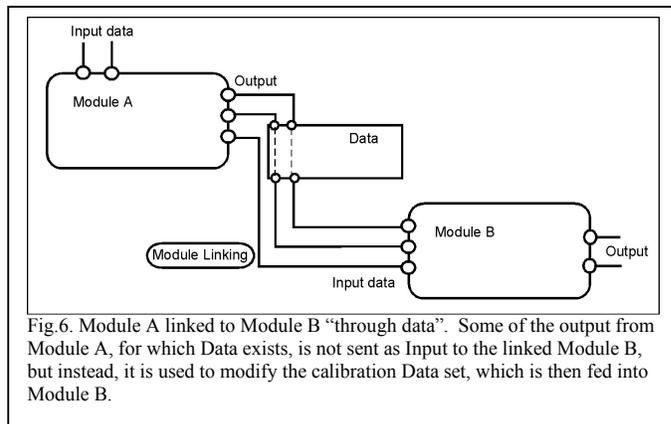

Fig.6. Module A linked to Module B "through data". Some of the output from Module A, for which Data exists, is not sent as Input to the linked Module B, but instead, it is used to modify the calibration Data set, which is then fed into Module B.

the future. How do we treat scenarios that produce results that are not available as part of the existing data sets?

The idea is to describe a scenario as relative to the calibrated base run, as a perturbation of this base run, which is then fed into the next component as an increment to the same data set that was used to calibrate that downstream component before (Voinov and Cerco, 2010). At any time step, the upstream component output is corrected based on the data set available, before it is fed into the next component as input. What is most important, we no longer need to undergo the tiresome re-calibration process, when changing the components 'above'. Only the scenario runs will need to be rerun, since, obviously, a modified upstream component will likely produce different set of perturbations, which will change the performance of the components 'below'.

Data, in fact, should be treated as an intrinsic part of any model, or even as models on their own. Data modules will exist along with model modules and when designing integrated systems we will choose which modules to use in which cases. This may simplify the calibration problem for cascade linking, when output from one module is fed further down the chain into the next module. Here data may serve as a constant 'reality check' between modules.

The situation becomes more complicated when we have circular linkage and modules exchange information in both directions. It is not clear where to place data in this case, and how the whole calibration exercise should be performed. I would argue that in such cases integration is becoming the less feasible the more empirical the modules are. CCA, DEVS and other software linking technologies are more suitable for linking modules that are process-based and less dependent on data and calibration. For more empirical models it should be safer to build integrated models as a whole, using some of the integrated frameworks described at the beginning of this paper. After all with empirical models it is always hard to draw the line for the best level of complexity. A simpler model can perform much better than a complicated one, and many processes and parameters that come with pre-existing components may turn out to be unimportant.

5. CONCLUSIONS

A lot has changed during the last 20 years since Agren and Bosatta's (1990) plea for more artistry and less construction-engineering or more science and less modelling in ecology. They were pointing out that it is not trivial to synthesize a larger system from pieces and then infer the system properties from the properties of the parts. With the modern module coupling tools it is getting only easier to put such larger systems together, but it is still questionable whether our products will be meaningful and useful.

While Silberstein (2006) insists that more data are needed and that models without data are not science, Andersen (2008) claims that there is already a deluge of data that can change the way science is done. Google and other similar efforts produce evidence that by analyzing huge arrays of data only, we can actually build new theories based only on correlation, ignoring causation. The new analytical tools that are going to be developed for petabyte computing in the "clouds" will require entirely different approaches to integration than the types of model integration that we were considering so far.

There are a lot of gains that can be made if instead of mechanistically plugging modules together we take into account the specific goals and features of the system, and approach the problem with some creativity. The considerations presented here by no means devalue the importance and need for module linking software tools, component interfaces and specifications. We are only reminding that models are more than just pieces of software and they require more to make them work together efficiently. For example, incorporating data sets in appropriate places within the integrated modeling system can help to keep components somewhat separate to avoid the propagation of perturbations and changes down the stream from one model to another. This can substantially increase the efficiency and accuracy of the integration process. We need data sets to be recognized as components on the same level as models. Such data components can then enter the integrated frameworks at various places, not only at the top, as input to drive the whole integrated model, and at the bottom, to compare with the output and to calibrate the model. Data components can be also used between components to test, adjust, and correct the data flows inside the integrated model. This will help to keep components more independent, and reduce the overall complexity of the calibration task for the whole integrated model.

ACKNOWLEDGEMENT

REFERENCES

3MRA, Multimedia, Multi-pathway, Multi-receptor Exposure and Risk Assessment, 2009. http://www.epa.gov/athens/research/projects/3mra/.

Ahuja, L.R., J.C. Ascough II and O. David, 2005. Developing natural resource models using the object modeling system: feasibility and challenges, Advances in Geosciences 4 pp. 29–36

Agren, G.I. and Bosatta, E., 1990. Theory and model or art and technology in ecology. Ecol. Modelling, 50: 213-220.

Athanasiadis, I.N., A.E. Rizzoli, M. Donatelli, and L. Carlini, 2008. Enriching environmental software model interfaces through ontology-based tools. Journal of Advanced Systemic Studies.

Anderson, M.C., W.P. Kustas and J.M. Norman, 2003. Upscaling and downscaling - a regional view of the soil-plant-atmosphere continuum, Agronomy Journal, 95(6), 1408-1423.

Anderson, C., 2008. The End of Theory: The Data Deluge Makes the Scientific Method Obsolete, The Wired Magazine, August, http://www.wired.com/science/discoveries/magazine/16-07/pb_theory

Babel, 2004. Lawrence Livermore National Laboratory homepage http://www.llnl.gov/CASC/components/babel.html

Bernholdt, D.E., B.A. Allan, R. Armstrong, F. Bertrand, K. Chiu, T.L. Dahlgren, K. Damevski, W.R. Elwasif, T.G.W. Epperly, M. Govindaraju, D.S. Katz, J.A. Kohl, M.

Krishnan, G. Kumfert, J.W. Larson, S. Lefantzi, M.J. Lewis, A.D. Malony, L.C. McInnes, J. Nieplocha, B. Norris, S.G. Parker, J. Ray, S. Shende, T.L. Windus and S. Zhou, 2004..A component architecture for high performance scientific computing, International Journal of High Performance Computing Applications https://e-reports-ext.llnl.gov/pdf/314847.pdf ACTS Collection Special Issue, 75 pp.

Bicknell B., J. Imhoff, J. Kittle, A. Donigian, R. Johanson and T. Barnwell, 1996. Hydrologic Simulation Program – FORTRAN User's Manual for Release 11, United States EPA Environmental Research Laboratory, Athens, GA.

Bormann, H., L. Breuer, Graff, T., J. A. Huisman, 2007. Analysing the effects of soil properties changes associated with land use changes on the simulated water balance: A comparison of three hydrological catchment models for scenario analysis, Ecological Modelling, 209: 29-40

Boumans, R., R. Costanza, J. Farley, M. A. Wilson, R. Portela, J. Rotmans, F. Villa, and M. Grasso. 2002. Modeling the Dynamics of the Integrated Earth System and the Value of Global Ecosystem Services Using the GUMBO Model. Ecological Economics 41: 529-560

Cerco, C. and T. Cole, 1993. Three-dimensional eutrophication model of Chesapeake Bay, Journal of Environmental Engineering 119 (6) pp. 1006–10025. Full Text via CrossRef

Cerco, C., 2000, Phytoplankton kinetics in the Chesapeake Bay model, Water Quality and Ecosystem Modeling 1 (1–4), pp. 5–49. Full Text via CrossRef

Cerco, C., D. Tillman and J.D. Hagy, 2010. Coupling and comparing a spatially- and temporally-detailed eutrophication model with an ecosystem network model: an initial application to Chesapeake Bay, Environmental Modelling & Software 25 (4), pp. 562–572.

David, O., S.L. Markstrom, K.W. Rojas, L.R. Ahuja and I.W. Schneider, 2002. The Object Modeling System. In: L. Ahuja, L. Ma and T. Howell, Editors, Agricultural System Models in Field Research and Technology Transfer, CRC Press, pp. 317–330.

Dibner, P., D. Arctur, 2008. Addressing Cultural and Institutional Barriers to Data and Model Interoperability. . In: M. Sànchez-Marrè, J. Béjar, J. Comas, A. Rizzoli and G. Guariso (Eds.), Proceedings of the International Environmental Modelling and Software Society (iEMSs), p.711.

DiToro, D., 2001. Sediment Flux Modeling, John Wiley and Sons, New York.

Ewert F., H. van Keulena, M. van Ittersuma, K. Gillera, P. Leffelaara, R. Roetter, 2006. Multi-scale analysis and modelling of natural resource management options

FRAMES, 2009a http://www.epa.gov/ATHENS/research/modeling/3mra.html.

FRAMES, 2009b http://www.epa.gov/athens/research/projects/3mra/index.html.

Janssen, S., Andersen E., Athanasiadis, I. and van Ittersum, M., 2008. An European database for integrated assessment and modeling of agricultural systems. In: M. Sànchez-Marrè, J. Béjar, J. Comas, A. Rizzoli and G. Guariso (Eds.), Proceedings of the International Environmental Modelling and Software Society (iEMSs), p.719.

Jetten, V., de Roo, A., Favis-Mortlock, D., 1999. Evaluation of field-scale and catchment-scale soil erosion models. Catena 37_1999.521–541.

Hřebíček, J., K. Kisza, 2008. Conceptual Model of Single Information Space for Environment in Europe. In: M. Sànchez-Marrè, J. Béjar, J. Comas, A. Rizzoli and G. Guariso (Eds.), Proceedings of the International Environmental Modelling and Software Society (iEMSs), p.735.

Leavesley, G.H., S.L. Markstrom, M.S. Brewer, R.J. Viger, 1996. The Modular Modeling System (MMS) - The Physical Process Modeling Component of a Database-Centered Decision Support System for Water and Power Management. Water Air and Soil Pollution 90(1–2):303–311

Linker, L., G. Shenk, R. Dennis and J. Sweeey, 2000. Cross-media models of the Chesapeake Bay watershed and airshed, Water Quality and Ecosystem Modeling 1 (1–4), pp. 91–122.

Meadows D.H., D. H. Meadows, 1979. The Limits to Growth. Macmillan, 209 p.

Moore R., P. Gijsbers, D. Fortune, J. Gergersen and M. Blind, 2005. OpenMI Document Series: Part A Scope for the OpenMI (Version 1.0), HarmonIT.

Müller J.P. 2009, Towards a formal semantics of event-based multi-agent simulations, in : Multi-agent Based Simulation IX, Nuno D. & J. S. Sichman Eds., LNAI 5269, Springer Verlag, pp. 110-126.

OpenMI, 2009. The Open-MI life project website http://www.openmi-life.org/, 2009.

Peckham, S. 2010. CSDMS Handbook of Concepts and Protocols: A Guide for Code Contributors. http://csdms.colorado.edu/wiki/Help:Tools_CSDMS_Handbook

Rizzoli, A.E., M. Donatelli, I.N. Athanasiadis, F. Villa, and D. Huber, 2008. "Semantic links in integrated modelling frameworks," Mathematics and Computers in Simulation, vol. 78, pp. 412-423.

Silberstein, R.P., 2006. Hydrological models are so good, do we still need data? Environmental Modelling & Software 21: 1340-1352

Voinov, A., Cerco, C., 2010. Model integration and the role of data. Environmental Modelling and Software 25(8): 965-969

Voinov, A., H. Fitz, C., Costanza, R., 1998. Surface water flow in landscape models: 1. Everglades case study Ecological Modelling 108: 131 – 144

Voinov, A., C.Fitz, R. Boumans, R. Costanza. 2004. Modular ecosystem modeling. Environmental Modelling and Software.19, 3: p.285-304.

Voinov, A. A., C. DeLuca, R. R. Hood, S. Peckham, C. R. Sherwood, J. P. M. Syvitski (2010), A Community Approach to Earth Systems Modeling, Eos Trans. AGU, 91(13)

Wang H., B. Johnson, 2000, Validation and application of the second-generation three-dimensional hydrodynamic model of Chesapeake Bay, Water Quality and Ecosystem Modeling 1 (1–4) (2000), pp. 51–90.

Zeigler B. (1976). Theory of Modeling and Simulation (first ed.). Wiley Interscience, New York