Environmental Modelling & Software 55 (2014) 1–24

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Design of a component-based integrated environmental modeling framework

Gene Whelan [a,*], Keewook Kim [a,b], Mitch A. Pelton [c], Karl J. Castleton [c], Gerard F. Laniak [a], Kurt Wolfe [a], Rajbir Parmar [a], Justin Babendreier [a], Michael Galvin [a]

[a] U.S. Environmental Protection Agency, Office of Research and Development, Athens, GA 30605, USA
[b] U.S. Department of Energy, Oak Ridge Institute for Science and Education, Oak Ridge, TN, USA
[c] Pacific Northwest National Laboratory, Richland, WA, USA

ARTICLE INFO

Article history:
Received 3 June 2013
Received in revised form
3 January 2014
Accepted 10 January 2014
Available online 1 February 2014

Keywords:
Integrated environmental modeling
Multimedia modeling
IEM
Risk assessment
FRAMES

ABSTRACT

Integrated environmental modeling (IEM) includes interdependent science-based components that comprise an appropriate software modeling system and are responsible for consuming and producing information as part of the system, but moving information from one component to another (i.e., interoperability) is the responsibility of the IEM software system. We describe and discuss the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES), a component-based IEM system, from the standpoint of software design requirements which define system functionalities. Design requirements were identified in a series of workshops, attended by IEM practitioners, and reported in the development of a number of IEM software systems. The requirements cover issues associated with standards, component connectivity, linkage protocols, system architecture and functionality, and web-based access, all of which facilitate the creation of plug & play components from stand-alone models through a series of software support tools and standards.

Published by Elsevier Ltd.

Software/Data availability

FRAMES-2
Software Developer Mitch Pelton and Karl Castleton
Address PNNL, Battelle Boulevard, Richland, WA 99354 USA
Tel +1 509 372 6551 and +1 970 248 1837
Fax +1 509 372 4995
E-mail Mitch.Pelton@pnnl.gov and Karl.Castleton@pnnl.gov
First available 2005
Hardware requirements 700 Mhz CPU 512 MB RAM
Software requirements Windows 95 Or Newer
Availability Open source (no license, source code available upon request)
Cost Free
Program language Visual Basic and C++
Program size 700 MB on disk after installation
Software Access (download FRAMES) http://iemhub.org/resources/133/

1. Introduction

Many complex environmental problems can benefit from a multi-disciplinary analysis provided by integrated modeling, and only by implementing systems thinking and integrative approaches that complement traditional single-discipline approaches will we be better able to solve challenging environmental problems (Babendreier et al., 2007; Gaber et al., 2008). Laniak et al. (2013) and Whelan and Laniak (1998) chronicle the evolution of integrated technologies that view the environment from a holistic, systematic viewpoint. A number of researchers (e.g., Gaber et al., 2008; Jakeman and Letcher, 2003; Laniak et al., 2013; MEA, 2005; Parker et al., 2002) have articulated the need to

- solve increasingly complex real-world problems involving the environment and its relationship to human systems and social and economic activities.
- create cost-effective, harmonious, higher-order systems thinking and holistic, equitable solutions that reflect the inherent complexity of environmental systems.
- develop and organize multidisciplinary knowledge through a science-based structure that explains, explores, and predicts

* Corresponding author. Tel.: +1 706 355 8305; fax: +1 706 355 8302.
E-mail address: Whelan.Gene@epa.gov (G. Whelan).

1364-8152/$ – see front matter Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.envsoft.2014.01.016

environmental-system responses to natural and human-induced stressors.
- break down research silos and join scientists from multiple disciplines with decision-makers and other stakeholders to solve problems in which the social, economic, and environmental considerations are highly interdependent.
- foster increased knowledge and understanding of the system, reducing the perception of 'black-box' modeling, and increase awareness and detection of unintended consequences of decisions and policies, resulting in a movement toward transdisciplinarity (Tress et al., 2005) and participatory modeling (Voinov and Bousquet, 2010).

Integrated environmental modeling (IEM) concepts have been around for over 40 years, paralleling developments and advancements in the computer industry (e.g., Abbott et al., 1986; Cohen, 1986; Crawford and Linsley, 1966; Metcalf and Eddy, 1971; Parker et al., 2002; Patterson et al., 1974). Environmental assessments supported by modeling evolved from single-stressor-, single-pathway-based analysis toward fully integrated systems analysis of stressors that impact human and ecological end-points of concern (Laniak et al., 2013; Whelan and Laniak, 1998). IEM is a systems analysis approach with inter-dependent science-based components (models, data, modules, and assessment methods) that, together, are the basis for constructing an appropriate modeling system (Gaber et al., 2008). A model is the mathematical representation of a process or concept (Sippl and Sippl, 1980) coded into a computer language for execution on a computer (e.g., source code and executable). An IEM module consists of a model and model description, user interface (UI) for model-specific input, and pre- and post-processors to convert model input/output (I/O) for recognition by the system (Buck et al., 2002).

IEM is represented by software-based computational systems (platforms or frameworks) that describe how science-based components will be organized/linked and used to address a specific problem coherently (Gaber et al., 2008). IEM systems focus on transferring information between components by capturing a conceptual site model (CSM), establishing local metadata standards of I/O and models/databases, managing data/information flow among models and throughout the system, facilitating quality control of data/information exchanges (e.g., units checking, units conversion, inter-language transfers), performing warning/error handling, and coordinating sensitivity/uncertainty analyses. An environmentally based CSM, for example, represents an environmental system and the biological, physical, and chemical processes that determine transport of contaminants through environmental media to environmental receptors within the system (ASTM, 2008); this identifies linkages of concern for a particular problem and clarifying what needs to be done. Although most IEM applications have typically supported the standard chemical risk paradigm (problem statement, fate and transport, toxicity, and risk assessment) (EPA, 2005, 2000, 1989, 1986), IEM systems have wide applicability for addressing non-chemical-based problems such as quantitative microbial risk assessments (Hunter et al., 2003; Haas et al., 1999) or radioactive risk assessments (DOE and U.S. Department of Energy, 1997; Eslinger et al., 2006; Kincaid et al., 1998).

Several computational software systems have been designed and implemented to facilitate execution of a modeling system, as noted in Table 1. Table 1 is not the most comprehensive list and should not be construed as making judgments as to which software systems are more worthy than others to be on the list. There are multiple opinions as to when a modeling infrastructure becomes a "modeling framework," what and how many components it should contain, or even if the ones on the list are worthy of being considered frameworks. Just checking off a list of implemented functional features does not make it a modeling framework. For example, if extensibility beyond the software's original scope is the prerequisite, then to what degree? Therefore, each has it own unique strengths and weaknesses; and no one system can meet the needs of every problem or user. For example, some: are designed for high performance computers (HPCs), while others are for personal computers (PCs); allow users to develop custom-designed assessments, while others require system developers, not users, to integrate components; are permanently connected and not easily modified; link to GIS visualization while others do not; or focus on ecosystem services while others address chemical, microbial, or radioactive problems. The system that best meets one's needs is determined by its design and functionality. Hence, this paper describes the design and functionality of a software system from a common community perspective, specifically focusing on the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES).

2. Materials and methods

A description of FRAMES is discussed in Section 3 in the context of software design requirements, characteristics a piece of software possesses to function adequately for its intended purpose. They are sometimes called attributes (Whelan and Nicholson, 2002), and a good requirement is testable. Although there is no universally accepted list of requirements for IEM systems, we attempted to compile one by reviewing available documents (Brooks, 1987; Gause and Weinberg, 1989; HarmonIT, 2002; IEEE, 1998; Peckham et al., 2013; Whelan and Nicholson, 2002; Wiegers, 1999) and synthesizing discussions from a series of workshops involving IEM developers, modelers, and practitioners (EPA, 2008, 2007; Gaber et al., 2008; iEMSs, 2012, 2010; Laniak et al., 2013; Moore et al., 2012; SOT, 2012; Whelan and Nicholson, 2002). Nineteen IEM software requirements are summarized, covering areas associated with 1) standards, component connectivity, and linkage protocols, including semantic and pragmatic mediation, data-transfer compatibility, units/unit conversions, temporal and spatial discretization, interoperability between systems, and mass conservation; 2) system architecture and functionality, including programming language interoperability, plug & play and intra-system security, interoperability across operating systems, component ownership/familiarity, execution management, multiple logical entry points, graphical user interfaces and visualization capabilities, exception handling, service components, and system documentation and help; and 3) web-based connections and GIS connectivity. They are identified and defined as follows:

1. *Semantic Mediation* refers to information exchange content (Wang et al., 2009), the meaning of data (Wang et al., 2009), and how differences are accommodated between model variable names and the standards utilized to describe I/O (e.g., name, type, cardinality, range, etc.), model (e.g., analytical, numerical) metadata characteristics, and pedigree of data, where applicable.
2. *Pragmatic Mediation* refers to information exchange context (Wang et al., 2009), the use of data (Wang et al., 2009; Sheth, 2001), and how the relationships between variables are accommodated [i.e., intended use of metadata about data (Sheth, 2001)].
3. *Units and Unit Conversions* refer to the ability to identify and track units associated with all variables and their values.
4. *Data-transfer Compatibility* refers to the protocol for passing information between modules.
5. *Plug & Play and Intra-system Security* refer to the system's capability to allow components to be added or removed in a relatively easy manner, allow for transparent implementation of the component within the system, and control unauthorized modifications to components in the system.
6. *Service Components* refer to tools or utilities that encapsulate a special feature that can be automatically instantiated by the system (e.g., sensitivity/uncertainty, data storage options).
7. *Temporal Discretization* refers to the ability to describe and transfer temporal information in discrete components, typically at medium interfaces (i.e., boundary conditions), thereby describing the allocation and distribution of time. These include single values versus time series; time series start and end (e.g., Julian day, specific Gregorian calendar dates, time zero with an open-ended ending time, etc.); description of values within the time series (point values versus step functions, even-increments versus variable-stepping, etc.); and correlating times for runtime execution between models.
8. *Spatial Discretization* refers to the process of dividing geometry into finite elements (Random House, 2013), and describing and transferring spatial information. For IEM systems that link legacy components, description and transfer are typically at model interfaces, responding to differing scale and