Development of an Open Platform System
for Environmental Pathway Analysis

David J. LePoire, John J. Arnish, Emmanuel Gnanapragasam, and S.Y. Chen
Argonne National Laboratory, 9700 S. Cass Ave, Argonne, IL, 60439
Corresponding Author: David LePoire (dlepoire@anl.gov)

Introduction

Most existing computer codes for environmental pathway modeling have been developed to satisfy a specific objective (e.g., perform analyses to demonstrate regulatory compliance). These codes were written in various computer languages and software environments over time and often are not mutually compatible. In recent years, largely driven by advancements in industrial software development, a new concept based on “modularization” has emerged. This approach entails the development of common “modules,” or components, that can be shared by and used in different applications that have certain common needs. For instance, an air dispersion model can be written as a common component to be shared by several different applications, each with the need to model air dispersion of some material release. When this concept is fully developed, modeling applications would become an exercise of selecting, integrating, and applying a consistent combination of appropriate pre-existing modules for a specific problem. Although modularization holds promising advantages over the traditional approach, a number of issues do exist. These issues must be fully addressed and resolved before the approach can be accepted as a new paradigm for environmental modeling. This paper discusses these issues and provides recommendations and a course of actions for future development.

Description

Like the traditional approach toward code development, the modularization approach for developing a complete modeling package usually

* Work supported by the U.S. Department of Energy under Contract No. W-31-109-ENG
consists of four generally distinctive elements (Figure 1):

- **Application Layer** – Users utilize the application with specific data to communicate understanding or demonstrate compliance.
- **Presentation Layer** – A user-friendly interface facilitates modeling and understanding through guided input and results presentation.
- **Model Layer** – Models are encoded by developers to describe physical phenomena.
- **Data Layer** – Data are collected to support parameters and analysis using models.

While the traditional method integrates these elements into a single code, the modularization approach instead aims at building a code “system” consisting of lineated components that can be used and reused for different purposes.

If a next-generation risk-modeling environment is to be successful, it must address a range of needs and issues: flexibility, software dissemination, quality assurance, life-cycle development, maintenance, and platform reliance. The various layers and issues associated with an open environmental architecture have been explored and applied to data, models, and user interface elements of the RESRAD code [1]: (1) data layer – a nuclide web service has been developed for XML communication between an internet server and models utilizing the data (2) model layer – the RESRAD-OFFSITE and DUST models have been integrated by factoring the RESRAD model into components and allowing model replacement, and (3) presentation layer and application layer – the user interface for this combined model utilizes interface components designed to support model integration and data sharing.

**Results**
Using standard commercial software tools, a flexible open architecture for environmental pathway analysis has been explored and demonstrated. This architecture allows for distributed component and data maintenance and distribution across multiple platforms while still maintaining the verification and validation of the integrated system.

To further develop this open architecture paradigm, an interagency committee would specify the standards for communicating between codes and establish the minimum validation/verification process required for individual components. Code developers at specific agencies or institutions would maintain control over their own source, but use the specified standards for “publishing” their codes for inclusion in other applications. While this approach would work best with new code development, methods for “wrapping” legacy code are available that allow reuse of existing code [2]. This structure would make it possible to embed distributed components within everything from simple spreadsheet applications, to commercial geographical information system (GIS) packages such as ArcView™, to a visual programming environment, if that was desired.

The use of distributed objects using standard commercial development and communication tools would even make it possible to maintain object components and data on different distributed servers and ensure that the architecture is “future-proof;” (i.e., the latest general information technologies can be utilized).

References

Figure 1 Example of how a next-generation environmental modeling system might be structured.