

EMDS 5.0 and Beyond

**Steve Paplanus, Bruce Miller, Philip J. Murphy, Keith Reynolds
and Michael Saunders**

Abstract The EMDS Consortium plans, designs, and oversees software development and integration for the EMDS suite of tools. This chapter presents an overview of the Consortium's plans for the next version of EMDS. Our objective is to release an EMDS 5.0 that will support web-services and Microsoft's workflow foundation. The 5.0 release will include major updates to the NetWeaver[®] engine, the Priority Analyst[®] engine, and the EMDS core.

Keywords Architecture · Workflows · Web services · Engine services · Data services · Business logic · NetWeaver · Criterium decisionplus · VisiRule

S. Paplanus (✉)

University of Redlands, 1200 East Colton Avenue, Redlands, CA 92373, USA
e-mail: Steve_Paplanus@spatial.redlands.edu

B. Miller · M. Saunders

Rules of Thumb, Inc., 11817 Cedar Mill Road, North East, PA 16428, USA
e-mail: bjmillier@gmail.com

M. Saunders

e-mail: mcs5@psu.edu

P. J. Murphy

InfoHarvest, Inc., PO Box 25155 Seattle, WA 98165-2055, USA
e-mail: philip_murphy@infoharvest.com

K. Reynolds

U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station, 3200
SW Jefferson Way, Corvallis, OR 97331, USA
e-mail: kreynolds@fs.fed.us

M. Saunders

Department of Entomology, Penn State University, 501 Agricultural Science and Industries
Building, University Park, PA 16802, USA

1 Introduction

Development work on the Ecosystem Management Decision Support (EMDS) system was begun by the Pacific Northwest Research Station (U.S. Department of Agriculture, Forest Service) in 1995. For the initial versions of EMDS through version 3, system implementation was done by the Environmental Systems Research Institute¹ (Redlands, CA) under contract with the Forest Service through 2002. Stewardship of EMDS was transferred to the Redlands Institute at the University of Redlands in 2005 under a memorandum of understanding between the University and the Forest Service. The EMDS Consortium was organized shortly thereafter as a private, non-profit research and development group to continue system development. Original parties to the Consortium included the Redlands Institute and the Pacific Northwest Research Station, as well as two private companies (Rules of Thumb, Inc. and InfoHarvest, Inc.), that had been instrumental in providing the core decision-support technologies underlying EMDS version 3.0. The most recent addition to the Consortium was Logic Programming Associates (LPA) LLP of London, UK in 2010, which made available a new suite of powerful Prolog programming tools for integration into the EMDS architecture.

From its inception, members of the Consortium have regularly collaborated as an applied research team, using EMDS for knowledge engineering, logic modeling, system assessment, alternative analysis, and prioritization. These decision support projects ranged from local (e.g., forests) to landscape (e.g., watersheds,) to regional and national analysis scales. This applied research serves as a testing and proving ground for EMDS functionality, and as a feedback mechanism for design improvements. The objective of the present chapter is to lay out the Consortium's vision for the next release of EMDS (version 5.0) in terms of major new features and functionality that will take the current, well-established, desktop system for individual users to a powerful, industrial-strength enterprise system for natural resource agencies and large organizations.

2 Background

“[An Overview of the Ecosystem Management Decision-Support System](#)” (this volume) describes the core functionality of the current EMDS system, including the central roles played by its logic and decision engines in providing spatially enabled decision support for environmental analysis and planning. “[NetWeaver](#)” and “[Criterion DecisionPlus](#)” provide introductions to the commercial development systems, NetWeaver Developer (NetWeaver) and Criterion DecisionPlus (CDP), which are used to design the respective logic and decision models used in EMDS

¹ The use of trade or firm names in this publication is for reader information and does not imply endorsement by the US Department of Agriculture of any product or service.

applications. “An Overview of the Ecosystem Management Decision-Support System” also explained that although the EMDS system *per se* is public domain freeware, and is capable of running existing NetWeaver and CDP models, both NetWeaver and CDP are commercial products that must be purchased from their parent companies in order to build new or edit existing logic and decision models used in EMDS.

3 EMDS Version 4.2 Enhancements

The current EMDS software release, version 4.2 (October 2012), represents a comprehensive re-engineering of EMDS. Improved functionality in the 4.x architecture includes:

- an ArcGIS^{®2} 10.x add-in component,
- support for printing and exporting data and graphs,
- installers for both 32-bit and 64-bit systems, and
- an increase in the maximum number of features per analysis to 10,000.

New capabilities for the 4.2 product line include:

- support for both ArcGIS[®] versions 9.3 and 10.x and on Microsoft Windows[®] 32-bit and 64-bit operating systems,
- an increase in the maximum number of features per analysis to 16,000,
- modularization of the interfaces to the NetWeaver^{®3} and Priority Analyst^{®4} engines, and
- integration of VisiRule.⁵

Despite all these improvements, EMDS 4.x suffers from four key limitations:

- **It has a fixed analysis workflow.**⁶ A study area is chosen, a NetWeaver model run, and a CDP model can be executed to generate a prioritization. From the EMDS 4.x user interface, there is no way for an analyst to change that order, or execute one of LPA’s VisiRule models.
- **It processes a single feature at a time.** If analysis requires properties of groups of features, such analysis has to be done through geoprocessing of the features datasets before loading EMDS, so no dynamic updates are possible in the course of the analysis.

² ArcGIS is a product of the Environmental Systems Research Institute, Redlands, CA.

³ NetWeaver is a product of Rules of Thumb, Inc., North East, PA.

⁴ The Priority Analyst is an engine for Criterium DecisionPlus, a product of InfoHarvest, Seattle, WA.

⁵ VisiRule is a product of Logic Programming Associates, LLP, London, UK.

⁶ A workflow is an organized collection of activities (often, but not necessarily, sequential) that are executed in program code to accomplish some overall task.

- **It is single user only.** There is no way to collaboratively build or share an EMDS project in real time.
- **The number of features it can process is limited.** Due to current architecture in terms of how EMDS is integrated with ESRI's ArcMap, it does not reliably handle feature datasets.

4 EMDS Version 5.0 Design Objectives

The goal of this latest version of EMDS is to transform it into a complete design, analytical, and dynamic *scenario planning*⁷ framework (the current sense of the EMDS 4.2 framework is described in “[An Overview of the Ecosystem Management Decision-Support System](#)”). The purpose of this new version is to:

- Enable EMDS to support one or more analysts and managers to work on one or more projects concurrently across a wide variety of *clients*.⁸ The current desktop implementation allows for a single user to work on a project, and there is a limited import/export feature to allow another user to view or alter the current project.
- Allow for more flexible analysis for users of EMDS. Over the years, we have received many requests to add flexibility to the EMDS workflow. For example, users often wanted to create a spatial selection and then run just a CDP analysis. Currently, they would have to create a dummy NetWeaver model and run it before they could do the CDP analysis. In another case, any further geoprocessing done on the results was done outside of EMDS without any tracking. With the new framework, we allow the end user to select from a common set of pre-defined workflows to perform a much wider variety of operations than is possible with EMDS 4.2.

We will be migrating from a traditional desktop architecture into a distributed service-oriented framework. This framework for version 5 will be of *Type II*⁹ as defined by the SEI team (i.e., the Carnegie Mellon Software Engineering Institute, SEI). Future versions will move toward a *Type III*¹⁰ architecture to give additional flexibility to the systems.

⁷ Scenario(-based) planning can inform decision makers by presenting scenarios that combine known or assumed facts with other plausible future conditions to explore the implications of alternative future states of a system.

⁸ A client is the part of an application, generally remote, that the user sees and interacts with.

⁹ Type II frameworks are “typified by allowing users to customize services in a finite number of commonly understood ways based on shared, community-wide assumptions about what is needed”. (Phase I: Strategic Analysis of Problem; SEI team; <http://www.frames.gov/partner-sites/iftcss/phase-i/>).

¹⁰ Type III architectures are “typified by supporting the customization of services by users for specific, unique operational situations that may or may not be shared, community-wide ways of solving a particular problem”. SEI team; <http://www.frames.gov/partner-sites/iftcss/phase-i/>.

This framework will support true *provenance tracking*,¹¹ allowing programs to expose each step of the process and to change conditions dynamically and then view the results. Scientific provenance tracking is defined as having the knowledge of all the steps in producing the result—from design through acquisition of data, manipulation of data, analysis performed, and any additional manipulations. From this information, a user will be able to reproduce a given result consistently, regardless of the complexity of the process. With the new EMDS 5.0 Framework’s provenance subsystems, we will be able to track all this information, excluding the model design, which is currently captured in the model building software, and the raw data acquisition.

In addition, the framework will support multiple users and will be *multi-thread safe*,¹² allowing for client applications to be written as standalone applications, ESRI ArcMap add-ins, or as web clients. A defined application programming interface (API)¹³ is planned that will allow for extending the framework with additional data formats or analytical and modeling engines by the end user. A workflow editor will allow the end user to perform some changes to the execution steps within the EMDS client. This workflow editor will be the same editor the EMDS developers use to create the pre-defined workflows, and will display all the relevant higher level activities the end user selects. This will allow for user-defined workflows for users who have the necessary rights. These workflows can be added to the pre-defined workflow library. All these changes will transform EMDS from being solely an ArcMap add-in into a true suite of products supporting multiple platforms.

The current design enhancement plan for EMDS version 5.0 includes the following objectives:

- **Workflow Foundation.** The EMDS Framework will be powered by workflows, based on the Microsoft Windows Workflow Foundation[®] (WF). Windows Workflow Foundation is part of the Microsoft .NET Framework and was introduced in version 3 of the framework. WF is a workflow engine, programming model, and set of tools that allows developers and end users to build workflows that coordinate people and software (Chappell 2009). Windows Workflow

¹¹ Provenance tracking formally documents the ownership(s), origin(s), uses and transformations of computerized data. Provenance tracking is of particular concern with electronic data, because data sets are routinely modified and copied without citation of the originating data set or further documentation of data modifications.

¹² In computer science, a thread of execution is the smallest sequence of programmed instructions that can be managed independently by an operating system scheduler. A piece of code is thread-safe if it only manipulates shared data structures in a manner that guarantees safe execution by multiple threads at the same time. Multiple threads can exist within the same process (i.e. the running application) and share resources such as memory, while different processes do not share these resources.

¹³ An application programming interface (API) specifies how software components need to interact with each other. In practice, an API is embodied as a code library that includes specifications for routines, data structures, object classes, and variables.

leverages the concept of activities, which can be simple, with only one activity to execute, or complex, and composed of multiple simple activities. One or more activities can be combined to form a workflow, which is the actual entity that is passed to the workflow engine. In this new workflow architecture, each functional unit of the EMDS platform will be engineered as one or more WF activities, enabling users to organize their analyses as customized process chains. Each engine (NetWeaver[®], Priority Analyst[®], and VisiRule[®]) will have a standardized, *pluggable wrapper*¹⁴ that will be exposed within the EMDS Framework as a series of workflow activities. A default set of pre-defined workflows will support traditional EMDS process patterns (e.g., Project > Assessment > Analysis > Scenario > Prioritization).

A library of pre-built workflow activity templates will also be provided to enable users to create their own customized analytical process chains with a minimal amount of coding. A workflow editor will enable users to create, save, and re-use workflows. *Provenance metadata*¹⁵ will be recorded for each workflow activity, enabling users to undo, redo, and pivot from any step and move along an alternate workflow sequence. All workflows will support both synchronous and asynchronous interfaces.¹⁶ In addition, all workflows can be exposed as web services, accessible by both desktop and web clients. The workflow platform will be integrated with Microsoft's Project Trident[®] workbench (Trident Team 2011). Translators¹⁷ may be built to facilitate integration between Trident and third-party modeling packages such as the IBM Web Process Server.

- **Pure Microsoft .NET[®] Implementation.** The EMDS core and the integration of the NetWeaver[®] and Priority Analyst[®] engines will be re-engineered in .NET[®] to improve system performance and stability.
- **Multi-core CPU.** To speed up in-memory calculations and operations, EMDS will be updated to support hardware systems with multi-core CPUs.
- **Relational database management system (RDBMS) Support.** Currently EMDS supports SQL Server, Microsoft Access, and SQL Server Compact edition. EMDS 5 will add support for Oracle and Postgres, while supporting the use of Oracle[®] and Microsoft SQL Server Spatial[®] RDBMSs.

¹⁴ Pluggable functions (aka "plug-ins") let the user extend the core functionality of an application via software components that add specific features otherwise not present in the application. When an application supports plug-ins, it enables customization and extension.

¹⁵ With provenance tracking, sufficient metadata about current status and configurations are gathered and stored so that operations can be rolled forward and back.

¹⁶ Synchronous operations require the software to wait for the called operation to complete before the current process continues whereas asynchronous operations allow the program to continue doing other things while the asynchronous operation runs. Asynchronous operations generally are more complex to implement, as the program must be notified of their completion so that the program can update accordingly.

¹⁷ Translators convert data and/or programming calls from one convention to another. These are similar to wrappers mentioned below.

- **Graphical User Interface Tools.** The next generation of EMDS will have a new **Project Manager** component for adding, deleting, and updating project metadata, and for importing/exporting multiple projects. A **Report Manager** tool will enable users to create, select, and re-use reports, including support for auto-updating the data behind reports and sending reports on a pre-determined schedule. A new web-based user interface component will enable users to view existing EMDS projects and modeling results in tabular, graphical, and spatial formats.
- **Actions that change the state of the System.**¹⁸ For end users, actions that change the state of the system will be one of the biggest additions to the platform. Supporting actions will allow for running scenarios that are based upon some activity or action that modifies the state of the current system—through altering analytic models or data or both—and then re-running the analysis to see how it affects analysis outcomes. New map comparison tools will be provided to evaluate the change in the systems wrought by such actions.

5 EMDS Version 5.0 Architecture Design

The new architecture of EMDS will transform the platform from a simple ArcMap add-into a complete multi-faceted platform that supports modeling, analysis, actions and scenario-based planning. With the new architecture, instead of a single monolithic application, the work of the EMDS system is now broken into discrete parts set within a systems framework (Fig. 1). There are two low level sections of the framework, which are the Engine Services Tier and the Data Services Tier.

5.1 Engine Services/Wrapping Tier

For the analysis and modeling engines, we now have a layer between the rest of the framework and the individual engines (Fig. 2). For each engine type, we have abstracted out a common set of functions each engine supports and have a query-able interface to call engine-specific functionality via .NET wrappers.¹⁹ The

¹⁸ For example, an EMDS project is created to analyze a set of watershed conditions. After running the models, a possible action is to reforest a stream bank. This would lower the water temperature, which is an effect on the model since it includes temperature. Fish species may be affected due to this change, and the watershed condition may be improved. Another example is for a model for forest fuels management. After running the analysis, one possible action is to remove the low lying brush. If this is done, the particular areas would have a reduced fire danger.

¹⁹ In computer programming a library is a collection of subroutines, usually external to the application. Wrappers are sparse amounts of programming code that translate a library's existing interface into a compatible interface. This is done to allow code or data formats to work together which otherwise cannot, or to enable cross language and/or runtime interoperability.

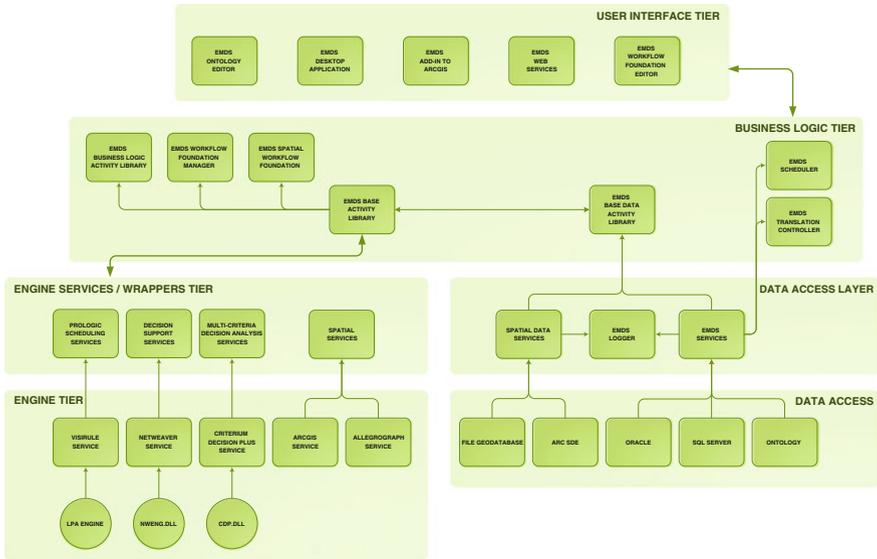


Fig. 1 Overview of the EMDS 5 service-oriented architecture

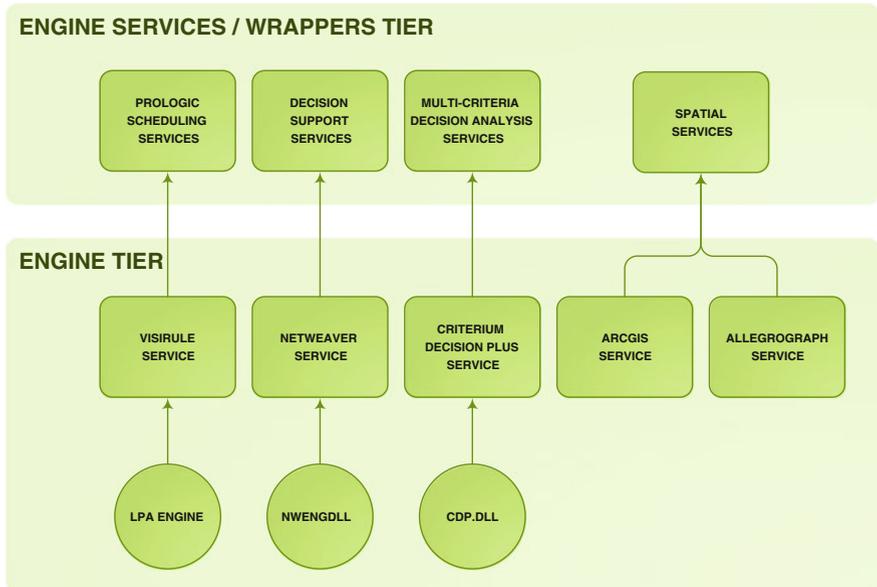


Fig. 2 Engine Services Tier. .NET wrappers provide the interface to each engine in this tier

wrappers provide the generic interface for each engine, as well as a queuing service and work-id management facilities to handle multiple user requests, even for engines that do not support multi-user or are not thread safe. All calls for analysis and processing of models will go through these interfaces. EMDS will support spatial, temporal, ontological,²⁰ scheduling, decision support, and multi-criteria analysis engines as default components within the new framework.

5.2 Data Services Tier

With the Data Services Tier, all data storage functionality is hidden behind services and objects with which the Business Logic Tier and Presentation Logic will interact (Fig. 3). For EMDS 5, we plan to support spatial, traditional data sources, and ontological data sources. Initial database support will be SQL Server, SQL Server Compact, and Oracle. For spatial storage, we will support file geodatabases, ArcSDE, SQL Server, and AllegroGraph. For ontological sources, initial support will be for Allegrograph, with future support of Oracle already planned. These are mapped to the Data Services and Spatial Data Services.

The key changes to the EMDS database structure will be driven by the need to fully support provenance recording, with assistance from the EMDS Logger service, and the fact that we need to support a more flexible structure than the old Project \Rightarrow Assessment \Rightarrow Analysis workflow process when we add the actions capability and the workflow engine. There will be a predefined set of analytical workflows, one of which will match exactly the existing EMDS workflow, along with several other optional workflow process paths to assist in analysis and what-if processing beyond the current limits of the system. Therefore, the database will not only need to keep track of the map and attribute data, but must also include additional metadata to allow the system to handle each of the different workflows within the same schema. The database schema has also been modified to allow for multi-level undo functionality within the system along with ability to view the history of the data changes.

5.3 Business Logic Tier

Both the Engine Services Tier (Fig. 2) and the Data Services Tier (Fig. 3) will interact with the Business Logic Tier (Fig. 4). The Business Logic Tier will expose a series of Windows Communication Foundation (WCF) REST Services

²⁰ Ontologies allow for the organization of entities, concepts about entities, and relationships between entities. This means we can describe the world or a portion which we wish to deal with in an agreed upon formal vocabulary that allows other people to have the understanding that the original creator of the ontology meant. Once the ontology is created, an ontology engine can be used to infer logical consequences based upon the facts contained within the ontology.

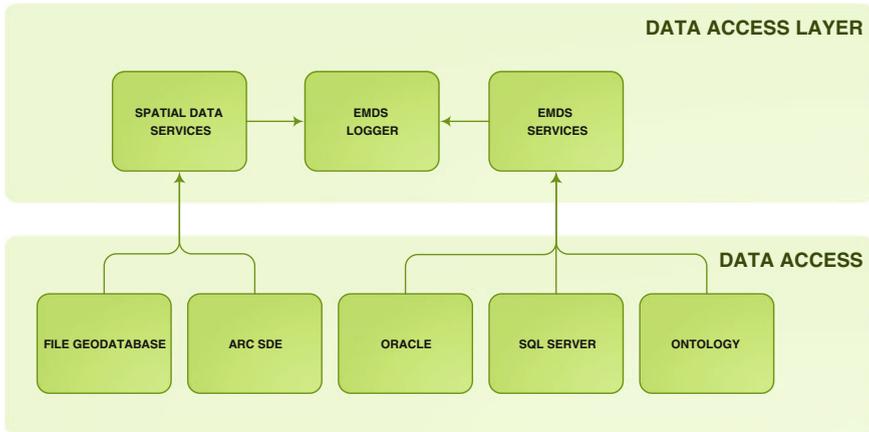


Fig. 3 Data Services Tier. Data storage functionality is hidden behind services and objects that are accessed by the Business Logic Tier (Fig. 4 of Chap. “Measuring Biological Sustainability Via a Decision Support System: Experiences with Oregon Coast Coho Salmon”)

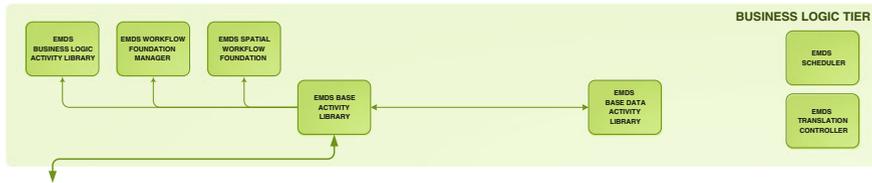


Fig. 4 Business Logic Tier. The Business Logic Tier exposes a series of REST services and workflow activity libraries that allow applications to easily tap into the power of the engines and database

(Flanders 2009) and Workflow Activity libraries to allow end applications to easily tap into the power of the engines and database. The EMDS Base Activity Library will contain the low level workflow activities²¹ and WCF REST Services to perform fine grain operations, such as create a new project, do a spatial UNION, or query for a subset of provenance information. This library of activities works along with the EMDS Base Data Activity Library, which handles the low level interactions for data access. These metadata on activities are

²¹ Low level workflow activities are the granular operations of the framework. Examples of these include reading and writing from the database, submitting queries to the ontology engines, running a ArcGIS Server service, or routing of messages between tiers.

saved inside the database, which can, in a future update, leverage a reasoning engine such as Allegrograph or LPA,²² and dynamically create a complex workflow based upon these activities and based on information stored in ontologies.

5.4 EMDS 5 Will Run on Workflows

Activities are chained together using Windows Workflow Foundation to create a complete workflow that is exposed in the EMDS Business Logic Activity Library and EMDS Spatial Workflow Foundation Library. An example would be the Run Priority Analyst Workflow defined in the EMDS Business Logic Library. This multi-step workflow is defined as follows:

- The CDP model is loaded via the EMDS Base Activity Library.
- Another activity is called in the EMDS Base Data Activity Library, which returns the records for the particular dataset.
- A SendandReceive activity in the EMDS Base Activity Library calls the Multi-Criteria Decision Analysis service in the Engine Services Tier. It is called and passes the model and dataset, and waits until the processing is completed and a dataset is returned.
- The EMDS Transaction Controller updates the provenance information.
- The result set is returned to the calling application.

The EMDS Transaction Controller (Fig. 4) is the main sub-system that allows the system and end user to access and manipulate the provenance information. This service handles undo requests, workflow branching due to actions, a true history of work done, user and application state, as well as handling any errors within the Business Logic Activity Libraries. The EMDS Scheduler handles the loading, editing, and processing engine for workflows. This component reads the activity workflows from the other activity libraries and runs the Windows Workflow Engine to perform the actual tasks.

²² Reasoning engines, which include Ontology Engines and Inference Engines, take a set of facts that is defined as the knowledge model of the system and given a set of conditions. The engine will then infer a logical conclusion or consequence of the action. LPA is a Prolog based Inference Engine system, in which you define several facts and the rules that are applied to it. From this, LPA can evaluate a query that is submitted to the system. LPA has a visual editor called VisiRule to allow for a graphical representation of facts. Allegrograph, an Ontology Engine, takes a standard ontology and allows you to create SPARQL queries to perform the evaluation of facts against.

6 Discussion and Conclusions

From the previous two sections, users of earlier versions of EMDS will hopefully have received a good sense of the major advancement that version 5 represents in terms of the power and flexibility to do environmental analysis and planning in a spatially enabled decision-support framework. Whereas the previous sections provided a more technical explanation of the new features and functionality of EMDS version 5, here we revisit that explanation in somewhat simpler language and in terms of the practical implications for the user experience.

The new workflows architecture moves EMDS from a static, one-size-fits-all, analysis paradigm to one virtually without limit. The new architecture allows the chaining together of any of the supported engines, in any order, to build very complex, maintainable workflows. The user is supported through a powerful workflows editor with the capabilities to flexibly create, maintain, and reuse existing workflows and their constituent workflow activities. EMDS 5 will support users new to the system with a library of workflow activities, lessening the experience necessary to build a solid EMDS application. Now, EMDS 5 will adapt to the user's process model rather than enforcing its traditional process model on the user.

EMDS 5 adds support for the popular relational database management systems Oracle[®] and SQL Server Spatial[®], among others. Through services, data will be able to reside independent of the EMDS deployment. Now EMDS will be able to use corporate data sources directly without the need to import the data into the stricter and harder to maintain EMDS format.

By using web services, EMDS 5 will move from being a strictly desktop application to one with great deployment flexibility. EMDS will still be available as a desktop application, but it will also be available in a server version in which EMDS clients communicate with the EMDS server to provide EMDS access to many remote users concurrently. Data storage and processing reside on the server, while the user interface can be another desktop application or even a web browser. In this context, EMDS modeling software can be developed and maintained centrally where the resources are, and applied where it makes most sense.

Through provenance tracking, EMDS application development and use will be much more interactive and will facilitate developing and using alternative workflow sequences. The utility is akin to having "go back" and "go forward" buttons on your web browser.

Coding in EMDS makes extensive use of the software technique of abstraction, which combined with the layered systems architecture, will make EMDS 5 easier to maintain and much more "plug and play," in that new features will be much easier to add. The technique of abstraction refers to the hiding of implementation details from the end user, and exposing a representation or interface that is easier to interact with. For example, if a new analysis engine is found that would enhance EMDS usability, it would not require changing the architecture of EMDS. It would, for the most part, require only writing the appropriate wrapper code (a generally small amount of code

that translates “engine-speak” to “EMDS-speak”) to talk to the new engine. Other external issues such as licensing may exist. Now though, EMDS is easily extendible to add new technologies and data sources.

Finally, readers may be curious to know when EMDS 5 will become available. As of June 2013, approximately 35 % of the version 5 architecture has been implemented. We expect that implementation will be at 65 % by October 2013, given current project support. Beyond that date, the funding picture is less certain, but we hope to have the final production system available through the Redlands Institute (University of Redlands, Redlands, CA) no later than October 2014.

References

- Chappell D (2009) The workflow way: understanding windows workflow foundation. <http://msdn.microsoft.com/en-us/library/dd851337.aspx>. Last accessed on 22 July 2013
- Flanders J (2009) An introduction to RESTful services with WCF. MSDN magazine. <http://msdn.microsoft.com/en-us/magazine/dd315413.aspx>. Last accessed on 22 July 2013
- Trident Team (2011) Project trident: a scientific workflow workbench. <http://tridentworkflow.codeplex.com/>. Last accessed on 22 July 2013