Decision support for forest management

Lecture 6a outline, advances in DSS technology

# Introduction

1. In this lecture, we look at some important ways in which DSS can be extended to provide better, more comprehensive services in support of DSS users.
2. First, though, we take a quick look at ways in which the EMDS design has been successful.
3. Next we review some of the current limitations of EMDS.
4. Finally, we look at some specific ways in which DSS functionality can be enhanced.
	1. This is drawn from the current design specification for EMDS 5
	2. But has broad implications for DSS design in general

# Why EMDS has been successful

1. Generality – a design framework
	1. Support for large, complex, abstract problems
	2. Many topics, any scale or scales
2. Transparency – more than just a pretty map
	1. Rational, repeatable, and fully documentable
	2. Transparent – intuitive explanation of model results
3. Simplicity – a V-8 moment
	1. Decomposition into two simpler components
4. Reasoning with incomplete information
	1. Derive priorities for missing information
5. Supports multiple interdependent spatial scales

# Current limitations of EMDS 4.2

1. It has a fixed analysis workflow.
	1. A study area is chosen, a NetWeaver model run, and a CDP model can be executed to generate a prioritization.
2. It processes a single feature at a time.
	1. If analysis requires properties of groups of features, such analysis has to be done through geoprocessing of the features datasets before loading EMDS.
3. It is single user only.
	1. There is no way to collaboratively share an EMDS project in real time.
4. The number of features it can process is limited.
	1. Due to current architecture (in terms of how EMDS is integrated with ESRI’s ArcMap), it does not reliably handle large feature datasets.

# Extending DSS with new technologies

## Workflows

1. A workflow is an organized collection of activities (often, but not necessarily, sequential) that are executed in program code to accomplish some overall task.
	1. Automation
		1. Example
	2. Customization
		1. Example
2. Predefined workflows
	1. Capture the business logic of a DSS
		1. What the DSS can do
		2. How it does it
	2. EMDS is implemented in workflows (see below)
3. Custom workflows
	1. Modifying the predefined workflows
	2. Extending the predefined workflows
4. What is a workflow component?
	1. Analogous to NetWeaver
		1. An interface to design workflows
		2. An engine that executes the workflow
		3. The workflow itself is analogous to a knowledge base
			1. A set of activities and their relations (e.g., the flow)
5. Workflow systems
	1. Windows Workflow Foundation
		1. Microsoft Science
		2. Open source
	2. Trident
		1. Customization of WWF

## Service-oriented architecture (handout)

1. Tiers
	1. Low level services
		1. Engine tier (see below)
		2. Data tier
	2. Abstracting the services
		1. Engine services
		2. Data Access Layer
	3. Business logic tier (see below)
	4. User interface tier (see below)
2. Abstraction
	1. This design abstracts out a common set of functions each engine supports
		1. Query-able interface to call engine-specific functionality via wrappers.
	2. Wrappers provide the generic interface for each engine,
		1. Also 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.
	3. All calls for analysis and processing of models go through these interfaces.
	4. EMDS will support spatial, temporal, ontological, scheduling, logic, and multi-criteria analysis engines as default components within the new framework.

## Engine tier

1. Visirule – a prolog engine that can be used for dynamic schedule optomization
2. NetWeaver – the logic engine
3. CDP – the MCDA engine
4. ArcGIS – a spatial service
5. Allelograph – an ontology engine
	1. Ontologies allow for the organization of entities, concepts about entities, and relationships between entities.
	2. We can describe the world (or a portion which we wish to deal with) in an agreed upon formal vocabulary that allows other people understand the reasoning of the original creator.
	3. Once the ontology is created, an ontology engine can be used to infer logical consequences based upon facts contained within the ontology.

## Business logic tier

1. This is essentially the “brains” of the system
2. Both the **Engine Services Tier** and the **Data Services Tier** interact with the Business Logic Tier.
	1. The Business Logic Tier exposes a series of Windows Communication Foundation REST Services and Workflow Activity libraries
	2. Applications can tap into the power of the engines and databases.
3. The **Base Activity Library** contains the low level workflow activities and WCF REST Services
	1. Performs fine grain operations, such as create a new project, do a spatial UNION, or query for a subset of provenance information.
	2. This library of activities works with the EMDS Base Data Activity Library, which handles the low level interactions for data access.
	3. These metadata on activities are saved inside the database
		1. In the future update, can leverage a reasoning engine such as Allegrograph or LPA, and dynamically create a complex workflow based upon these activities and based on information stored in ontologies.
4. **Business Logic Activity Library**
	1. Activities are chained together using Windows Workflow Foundation to create a complete workflow
	2. EMDS example: executing the Priority Analyst Workflow
		1. CDP model is loaded via the **Base Activity Library**.
		2. Another activity is called in the **Base Data Activity Library**, which returns the records for the particular dataset.
		3. A SendandReceive activity in the **Base Activity Library** calls the Multi-Criteria Decision Analysis service in the **Engine Services Tier**.
			1. It is called and passes the model and dataset, and waits until the processing is completed and a dataset is returned.
		4. The **Transaction Controller** updates the provenance information.
		5. The result set is returned to the calling application.
5. **Transaction Controller**
	1. The main sub-system that allows the system and end user to access and manipulate the provenance information.
	2. This service handles provenance tracking
		1. undo requests
		2. workflow branching due to actions
		3. history of work done
		4. user and application state
		5. any errors within the Business Logic Activity Libraries.
6. **Scheduler**
	1. Handles the loading, editing, and processing engine for workflows.
	2. This component reads the activity workflows from the other activity libraries and runs the Windows Workflow Engine to perform the actual tasks.

## User interface tier

1. EMDS Add-in (essentially the back-end engine)
2. Windows Workflow Foundation Editor
3. EMDS Desktop Application
4. Web Services
5. Ontology Editor

# Readings

Reynolds, K.M., and P.F. Hessburg. 2014. An overview of the Ecosystem Management Decision-Support system. Chapter 1 in Reynolds, K.M., P.F. Hessburg, and P.S. Bourgeron (eds). Making Transparent Environmental Management Decisions: Applications of the Ecosystem Management Decision Support System. Berlin: Springer. (**just section 1.11 as review**)

Paplanus, S., B. Miller, P. Murphy, K. Reynolds, and M.Saunders. 2014. EMDS 5.0 and Beyond. Chapter 14 in Reynolds, K.M., P.F. Hessburg, and P.S. Bourgeron (eds). Making Transparent Environmental Management Decisions: Applications of the Ecosystem Management Decision Support System. Berlin: Springer.

# Additional 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/). <http://tridentworkflow.codeplex.com/> (last accessed on 22 July 2013)