: What are the features of a model-driven DSS?

by Dan Power

Editor, DSSResources.com

This is the final column about features of computerized decision support systems. The focus is on model-driven decision support systems. The goal remains to identify from a user's perspective major, observable aspects, attributes or elements that distinguish one type of DSS from other types of DSS (Power, 2002) and from other computerized information systems.

In general, a model-driven DSS provides access to and manipulation of a quantitative model. DSS built using simple algebraic models provide the most elementary level of functionality. In general, model-driven DSS use more complex models, e.g., accounting, optimization and simulation, to provide decision support. In most implementations, model-driven DSS use the data and parameters provided by a decision maker to help in analyzing a situation. Model-driven DSS do not usually require large historical databases. Early versions of model-driven DSS were called model-oriented DSS by Alter (1980) and computationally-oriented DSS by Bonczek, Holsapple and Whinston (1981).

Alter (1980) identified DSS built using accounting models, representational models and optimization systems. These three subtypes of model-driven DSS have many shared features, but because of differences in the underlying modeling technology additional or modified features associated with manipulating the specific model are sometimes provided. To add to the complexity, a specific model-driven DSS may also have multiple subsystems that use various models. In general, the initial model-driven DSS built in the 1970s were largely independent of centralized information systems (cf., Keen, 1981) so users had to supply all of the data used by the system. Today's model-driven DSS are more likely to be integrated with other DSS and Information Systems.

Accounting models use accounting definitions to calculate the consequences of planned actions. A DSS that helps prepare monthly or quarterly budget forecasts probably uses an accounting model. Accounting models assume certainty in a situation about input parameters.

Representational models try to capture the dynamic behavior of a system and estimate the consequences of actions where there is uncertainty. Simulation is the most commonly used process for studying dynamic systems. For example, a store manager may use simulation in a DSS with an inventory model to determine order quantities. Also, simulation is often used in one time special decision support studies.

Optimization systems help estimate the results for various decision alternatives given a set of constraints. Linear programming is the most widely used optimization technique. A typical DSS application of linear programming involves resource allocation. For example, a DSS with optimization might assist a saw mill operator in determining how to cut a log to minimize waste or help a production manager blend inputs to minimize costs and still meet product constraints. Optimization is also used in one time special decision support studies.

Model-driven DSS are used to assist in formulating alternatives, analyzing impacts of alternatives, and interpreting and selecting appropriate options. Tasks that have been supported with model-driven DSS include crew deployment, job scheduling, advertising allocation, forecasting product usage, cost estimation and pricing, tax planning and investment analysis.

A classic example of model-driven DSS is the system implemented at Gotaas-Larsen Shipping Corp. in the late 1970s (cf., Alter, 1980). The company used a DSS for preparing and revising a 15 month operation plan for its cargo ships. The model-driven subsystem supported cash flow and pro forma analyses on a per ship, per voyage, per division and company-wide basis. The DSS helped users simulate results. The computerized system aggregated plans for individual feasible voyages to help managers assess whether the overall plan would be effective. A data-driven subsystem provided variance reports and performance tracking.

Concannon and Tudor (1992) reported development of a visual interactive model-driven DSS that was used to support railroad track maintenance operations. The DSS helped monitor and prioritize track maintenance work including scheduling repair crews. The goal was to schedule track maintenance in a cost-effective manner with minimal disruption to commercial traffic. A data-driven subsystem helped managers compare actual maintenance performance to the plan.

Computerized quantitative models are often developed as part of a decision support special study and these applications are sometimes incorrectly identified as DSS. In most cases in a special study, the application user interface in not as sophisticated and feature laden as is one found in a DSS. Examples of one time special studies that use models include merger and acquisition analysis, lease versus purchase decisions, new venture analysis, capital budgeting, and equipment replacement decisions. Turban and Aronson (1998) explain how Siemens' Solar Industries used a computer simulation built using ProModel to evaluate alternative designs for a photocell fabrication "cleanroom" (p. 146) and Wesleyan University's development of a student aid planning model (p. 292). Both Siemen's Solar Industries and Wesleyan University used the results of a decision support special study that relied upon quantitative models, but neither organization built and used a model-driven DSS. When classifying computer applications, the criteria in Power (2004) can help avoid classifying model-based special studies with model-driven DSS.

The following are major features of model-driven DSS from a user's perspective:

1) Change a model parameter or classic "what if" analysis. Performing "what if" analysis involves varying a single model input parameter over a reasonable range. This is a major feature of model-driven DSS. For example, a slider may be used to adjust values in a range.

2) Context specific help and model definitions. Users of a model-driven DSS often have questions about the quantitative model, its assumptions and the relationships among variable. A good model-driven DSS provides online help. The system may include a graphical representation of the model.

3) Create and manage scenarios. A scenario is a specified combination of values assigned to one or more variable cells in a model. In Excel, scenarios can involve as many as 32 variables. The scenario summary examines relationships between scenarios. Some model-driven DSS have predefined scenarios while other systems make it easy for users to add and modify scenarios.

4) Extract specific historical data values from an external database. For example, a model-driven DSS for investment analysis may provide a capability to extract historical stock information.

5) Generate a sensitivity analysis. Users often want to determine the impact of systematic changes in the values of one or two variables over a reasonable range on the results of a model. This capability is called sensitivity analysis (see Power, 2006). In Excel, one and two variable data tables can provide sensitivity analysis. In a model-driven DSS, the sensitivity analyses are usually predefined and in addition to data tables, charts are often provided to visually display the sensitivity of results.

6) Output selection. Model-driven DSS usually have multiple formats for displaying outputs. For example, it may be possible to select a pie or a bar chart. Some DSS based upon simulation provide a visual animation as the process simulation is occurring.

7) Specify and seek goals. Goal seek is a capability for specifying the desired result of a model. In Excel, when using goal seek the value in a specific cell is varied until the formula that is dependent on that cell returns the desired result.

8) Store inputs, results and user actions.

9) Value elicitation and data input. Values are elicited from the user of a model-driven DSS (Power, 2003b). There are three primary approaches for eliciting values: 1) numerical, 2) graphical, and 3)

verbal.

The benefits of model-driven DSS include: 1) examining more alternatives; 2) gaining insights about how a process works; 3) making better and more effective decisions; 4) reducing clerical work; and 5) reducing costs and saving time, especially reduction in decision process cycle time.

Please note: A quantitative model is an abstraction of relationships in a complex situation and the results of using a specific model-driven DSS need to be carefully monitored for ongoing validity and usefulness. If the model is incomplete, inaccurate or misspecified, the results can actually negatively influence a decision makers judgment. Model-driven DSS should be periodically reviewed and when necessary revised.

As always your comments, questions and suggestions are welcomed.

References

Alter, S.L. Decision Support Systems: Current Practice and Continuing Challenge. Reading, MA: Addison-Wesley, 1980.

Bonczek, R. H., C.W. Holsapple, and A.B. Whinston. Foundations of Decision Support Systems, New York: Academic Press, 1981.

Concannon, K and S. Tudor, "Visual Interactive DSS Streamline Maintenance Operations," OR/MS Today, Dec. 1992.

Keen, P.G.W., "Value analysis: Justifying decision support systems," MIS Quarterly, 5:1, 1981, 1-16.

Mallach, E.G., Understanding Decision Support Systems and Expert Systems, Burr Ridge, IL, Irwin, 1994.

Power, D. J., Decision Support Systems: Concepts and Resources for Managers, Westport, CT: Greenwood/Quorum, 2002.

Power, D., How can simulation be used for decision support? DSS News, Vol. 4, No. 14, July 6, 2003 (a).

Power, D., What DSS interface design is "best" for eliciting values? DSS News, Vol. 4, No. 11, May 25, 2003 (b).

Power, D., What type of DSS is X? or What type of DSS is a revenue management application? DSS News, Vol. 5, No. 7, March 28, 2004.

Power, D. J., "How does sensitivity analysis differ from 'What If?' analysis?" DSS News, Vol. 7, No. 16, July 30, 2006.

Turban, E. and J. E. Aronson, Decision Support Systems and Intelligent Systems (5th edition), Upper Saddle River, NJ: Prentice Hall, 1998.

Author: Daniel Power
Last update: 2008-08-09 11:28