

Paradigm: workflows to scenarios

Is it time to stop thinking about “workflow” – a series of steps in a linear process – and think instead about “scenarios” – a sequence of tasks which are performed when they are required?

By Jean-Claude Dulac, executive vice president, and Emmanuel Gringarten, product manager, Paradigm

Today, geoscientists and engineers use a mix of different software applications to perform their work.

These various components can be part of a same integrated suite or come from different vendors.

Some of the functionalities will be very familiar while others are only used occasionally and can therefore be classified as unfamiliar.

When users are familiar with an application, they know how to navigate through menus or shortcuts. If they are experts at the use of the application, they will want to record and make use of their own macros to streamline repetitive tasks.

On the other hand, if they are not familiar with an application or are non-expert at the tasks at hand, they will have difficulty knowing where to find all the available options that may be necessary to successfully complete their work.

Workflow – linear sequence

To help tackle these two different ways of working, namely either being guided through or easily automate a series of actions, the software industry introduced the concept of workflows.

Two types of workflows are therefore needed to handle these two very distinct processes: workflows or macros created by users; and workflows created by the manufacturer or by expert users to help the “unfamiliar” user navigate through the application’s many options.

Generally speaking, the term workflow refers to a sequence of steps performed to accomplish a specific task.

No linear component?

However, if one considers for example the seismic interpretation task, it has for all practical purposes no linear component.

It uses a limited number of commands, repeated over and over for many hours and sometimes days.

In the past, there were specialized applications for doing interpretation only; menus in the user interface were therefore generally simple.

Nowadays, interpretation is often part of a bigger process and therefore includes many more options such as the validation

of the interpretation and steps that go beyond traditional interpretation, e.g. prospect sanction or geological model construction.

It is also often part of integrated multi-disciplinary “seismic to simulation” software application suite.

An interpretation “workflow” must therefore provide a focused environment by adjusting the graphical renderings and by grouping together the relevant options in such a way as to optimize the interpretation task.

Results of the interpretations and the decisions that led to these interpretations must be captured by the workflow.

A workflow therefore becomes a user interface component that physically groups options in a logical order and associates them to the steps that are effectively executed as well as to the data that is used, the decisions that are made, and the results that are obtained.

From workflows to scenarios

From an asset team point of view, geoscientists and engineers work on generating alternative exploration or development scenarios.

Each scenario may consist of a specific structural interpretation, a reservoir model, or a development strategy.

The scenario is the central concept around which practitioners organize their work and their decisions. The scenario must therefore also be a central concept around which the application organizes both input and output data for users.

| Name | Last Updated | Create Date |
|---|--------------|-------------|
| Root Scenario* | Today | Today |
| CASE 1 - F2 and F4 connected* | Today | Today |
| Structure & Stratigraphy (Geology_study)* | Today | Today |
| Flow Simulation Grids (FSG_study)* | Today | Today |
| Reservoir Data Analysis (RDA_study)* | Today | Today |
| LOW NTG* | Today | Today |
| Reservoir Properties (Property_study)* | Today | Today |
| Fluid Saturation Workflow (Fluid_Saturation_Study)* | Today | Today |
| LGR and Upscaler (Rescaling_study)* | Today | Today |
| HIGH NTG* | Today | Today |
| Reservoir Properties (Property_study_0)* | Today | Today |
| Fluid Saturation Workflow (Fluid_Saturation_Study_0)* | Today | Today |
| LGR and Upscaler (Rescaling_study_0)* | Today | Today |
| CASE 2 - F2 and F4 disconnected* | Today | Today |
| INFILL DRILL PLAN* | Today | Today |
| 4 WELL PLAN* | Today | Today |
| Drilling Planner (WellPlanning_Study)* | Today | Today |
| 6 WELL PLAN (current)* | Today | Today |
| Drilling Planner (WellPlanning_Study_0)* | Today | Today |

Figure 1: Instead of doing tasks in a specific sequence or “workflow”, you can keep track of which specific processes, or “scenarios”, your data has been put through and when it was done

Since a workflow is the way tasks are organized to generate interpretations and construct models, an instance of a workflow should be associated to a specific scenario. Furthermore, by attaching the results of these activities to the execution of the workflow, the results of the workflow are now linked to a scenario.

Under the assumption that every action is recorded as users advance through the workflow, activities and results are linked. All the results ensuing from the execution of a workflow (in the most generic sense) are then associated to a scenario.

Results can be objects (e.g. structural maps) or numbers (e.g. OOIP estimates) and the activities that created them are recorded. Given two scenarios (or two executions of the same workflow), users can investigate differences in output results and understand correlations with input parameters.

Workflow reports can be automatically generated detailing all the steps taken and containing all associated decisions and results as well as explanations of the methods and the parameters that were used.

A complete study generally corre-

sponds to a nesting of tasks resulting in a hierarchy of scenarios and workflows (Figure 1).

This can be seen to mimic a decision tree in which the probability of an end result is compounded from the probability specified at every node in the tree.

Made by users or software company?

As discussed above, there needs to be two types of workflows.

The first workflow type is created by the software manufacturer or an expert user, they are "built-in" workflows.

A second type is created by users and interactively constructed by concatenating application commands or activities associated with a given task into macros through a workflow builder (Figure 2) that operates either within a software application or at a higher level encompassing different applications.

Working across applications is a requirement for uncertainty propagation and optimization loops.

From a software design point of view, the question then becomes:

What level of complexity should a workflow builder allow for? Are users proficient enough to be programmers, even if only visual programmers? How can one easily check the validity of user-created workflows? What tools should be available for that purpose? Does the workflow builder need a sort of debugger?

User-defined workflows should be quite simple, that is, mainly sequential.

Any additional and more complex programming should be done with scripting languages such as Python, JavaScript, TCL, C++, or C# and should provide access to the application object states and windows.

Specialized "company-specific" workflows become application "plug-ins" available to users at the same level as built-in workflows.

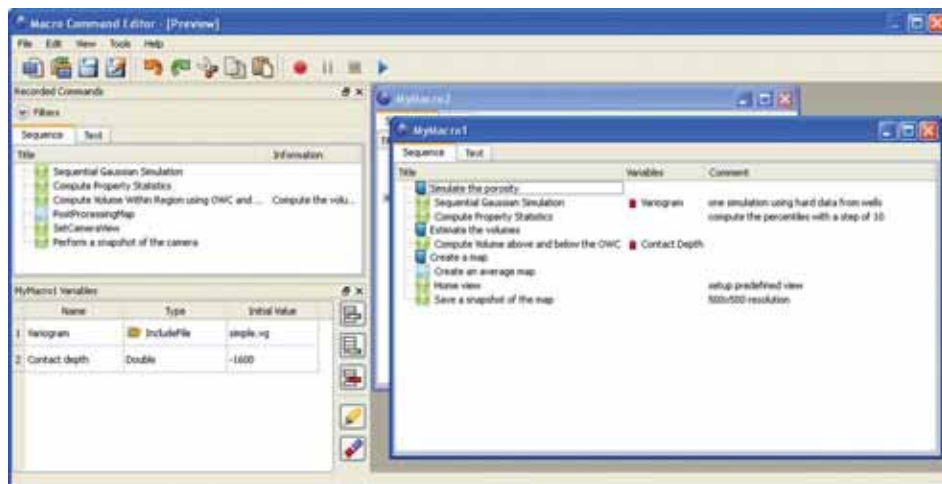


Figure 2: Users can construct their own workflows in the Macro Editor

Built-in workflows can serve different purposes. They can guide users through a series of activities or group sets of relevant commands that may otherwise appear at distant locations in the user interface. They may also act as a repository for processes that require many parameters such as the construction of a reservoir property model (Figure 3(a) and (b)).

In both cases, a workflow must remember all the choices made by the user, and enable a rapid replay of input data or modeling choices that were changed from one process to another. Furthermore, alternative scenarios can easily be created by simply copying the workflow and varying some parameters.

Since a single application platform rarely covers optimally all relevant technical aspects of asset management, it is necessary to construct and implement workflows across applications, specifically when dealing with complex optimization loops, real-time updates or uncertainty propagation.

In such a case, it is necessary to pilot the applications from an external workflow

engine. Typically, a Service Oriented Architecture needs to be implemented and each application becomes a service provider.

Data is passed from application to application as the workflow progresses. In our industry, initiatives such as WITSML, PRODML and RESQML are on track to provide the message data layer required in a SOA.

The service definition, however, is much more difficult to tackle. Many approaches exist in optimization systems in which a reservoir flow simulator cohabits with a reservoir modeling software package.

The messages passed between the applications are ad-hoc. For a more interpretative workflow, SOA architectures can be used to gather rather large chunks of functionalities, while "macros" and "plug-ins" bring additional technology and therefore additional workflow steps into existing applications.

Conclusion

Several goals can be accomplished using workflows: repeatability, ease-of-use, audit trails, and collaboration. They facilitate project and technology transfer and also provide training.

Built-in workflows and user-defined macros are two very important means of delivering efficient applications today.

Practical inter-application workflows are still in a distant future.

Linking the concepts of scenarios and workflows at the software level enables the grouping of both activities and results in better auditing, bearing in mind that "the methods used to obtain the results are as important as the results themselves."



Figure 3(a): Built-in workflow for building reservoir model architecture



Figure 3(b): Built-in workflow for capturing velocity model parameters