SPE 112268

A Framework for Design Space Exploration in Oilfield Asset Development

Cong Zhang[1], Amol Bakshi[1], Viktor Prasanna[1], Will Da Sie[2], and Birlie Bourgeois[2]

[1]University of Southern California, Los Angeles CA, USA
[2]Chevron Corporation, USA

This paper was prepared for presentation at the 2008 SPE Intelligent Energy Conference and Exhibition held in Amsterdam, The Netherlands, 25–27 February 2008.

Abstract

Design space exploration (DSE) is a common yet complex workflow in oilfield asset development. The "design" of an oilfield refers to a set of decisions about aspects ranging from well locations and number to facility sizing for optimum production. Evaluation of alternate designs – based extensively on reservoir simulations – corresponds to the evaluation of alternate development scenarios in face of uncertainty about subsurface structure and properties. The outputs of DSE influence many decisions in the development phase of an oilfield as well as operational decisions in a producing asset. In this work, we design and implement a generic framework to support DSE workflows in oilfield asset development. Our framework provides tools and services to allow rapid specification and evaluation of multiple design candidates using multiple realizations. The framework also supports hierarchical DSE workflows that allow users to first explore a large design space using proxy models and selectively refine the simulation quality of a smaller subset of designs via fine grained, detailed simulations. The usefulness of this framework is demonstrated through a case study that considers the design problem of selecting a drilling schedule for wells in an offshore oil and gas field.

1. Introduction

Design space exploration (DSE) refers to the general problem of selecting the values of a set of variables (input parameters) in order to optimize a certain function of those parameters. If the design problem involves many variables, it is typically impossible to exhaustively enumerate and then evaluate all design options in light of constraints on computational resources and the time to make the decision. For such cases, design space exploration can be modeled as a two-phase process: sampling (exploration) of the design space to identify a subset of design points, followed by an evaluation of each design point based on the utility function of interest to the decision maker. Commonly used techniques for the first phase include exhaustive exploration, random sampling, genetic algorithms, simulated annealing, etc. Evaluation of a design point can be abstracted as a function which can range from a simple arithmetic expression to a complex simulation. Because of the large number of design points to be evaluated for a non-trivial design space and the computing resources required for evaluating a given design point, DSE is typically a compute-intensive and data-intensive process.

DSE plays an important role in oilfield asset development. For instance, a reservoir development strategy deals with many *decision variables*, such as parameters in the field production system and physical properties of the geological reservoir. Some variables represent factors that can be controlled, and others represent uncertainty in available information at that stage. These variables are usually used as inputs to a reservoir simulator to generate a forecast of the production profile. To find the best development strategy, a simulation engineer generates design points from the decision variables and evaluates each design point by running a reservoir simulator followed by an objective function against the forecast of the production profile.

To improve the quality of asset development decisions, many evaluations of possible combinations of the decision variables are required. Designs are usually so complex that their effectiveness cannot be evaluated using an objective function expressed algebraically. Instead, numerical reservoir simulators and surface network simulators are typically employed (in standalone or coupled modes) to run computational models of the field. Also, the reservoir description is probabilistic due to uncertain knowledge of the reservoir geology, fluid properties, etc. Hence, it is to include multiple reservoir realizations in a DSE, thereby increasing the complexity of the workflow.

Most existing work on DSE has focused on techniques of exploring a design space for an optimal design. Techniques include genetic algorithms, simulated annealing, Monte-Carlo, etc. In [3], a hybrid Genetic Algorithm (GA) is developed to determine an optimal location of new wells. There is also some work on providing a framework for DSE workflow [1, 2, 8]. These work focuses on automating the data transfer in the optimization process and staging, formatting, and computation processes. It is assumed that a user generates the input for the DSE, perhaps through the use of portals that automatically configure and run DSE tools.

In this paper, we describe a framework for conducting DSE workflows in oilfield asset development. We concentrate more on the interactive, user-driven evaluation of strategies in a design space, and as much on techniques that automatically identify an 'optimal' design. The proposed framework allows various users - reservoir engineers, production engineers, and asset managers - to define and solve a class of DSE problems. To demonstrate the use of the framework, we study a well drilling scheduling design problem where the design space is defined as a set of alternate drilling schedules for new wells and each drilling schedule can potentially yield a different cumulative oil production over the period of interest. A schedule that maximizes cumulative oil production is identified based on the DSE results. We also consider the existence of multiple reservoir realizations representing the range of uncertainty about subsurface parameters.

This paper is organized as follows. In Section 2, we describe our design of the framework, including a discussion of the fundamental background of DSE, major components in the framework and a typical workflow that uses the framework. We then demonstrate the use of the framework by applying it to a illustrative use case in Section 3. We conclude in Section 4.

2. A Framework for Design Space Exploration

Our goal is to build a design space exploration framework specific for oilfield asset development. It is not our goal to present a new design optimization technique. Rather, we provide a framework upon which such optimization techniques can be better applied to design optimization problems in this domain. In this section, we will first discuss theoretical background of the framework. Then we will describe services and tools in the framework. Finally, a workflow example will be introduced to show how the framework is used.

2.1. Theory

A design space is defined as a collection of input parameters. Each point in the design space is called a design point or a design candidate. Each input parameter represents a *dimension* of the design space, and usually has an associated range of values. For example, the location of wells to be drilled can be used an input parameter in a DSE problem.

There are two aspects in a design space exploration problem: searching the design space and evaluating design points. Most DSE work and optimization algorithms are focused on how to search a design space for an optimal design point. For example, exhaustive exploration of a design space involves evaluating every possible combination of values of the input parameters and selecting the combination that yields the optimal value at the output. However, exhaustive DSE is not feasible for design spaces with many dimensions, especially when the evaluation of each design requires substantial computing resources. Other exploration techniques such as random sampling, genetic algorithms, simulated annealing, etc., are often used for many design problems. In these techniques, a design space is sampled and searched via a black-box optimization approach. New design points are generated based on information learned so far and by defining an appropriate neighbourhood function.

Our focus in this work is on evaluation of designs in a DSE problem. We argue that evaluation is as important as searching a design space, especially when: 1) each evaluation operation requires a lot of computing resources and human involvement, like numerical reservoir simulations; 2) a design optimization is under some time constraints; 3) large amounts of models are available for evaluation, for example, subsurface uncertainties result in multiple realizations for a reservoir. For these cases, it is usually unrealistic to apply all available models to evaluating design candidates and a subset of the models has to be identified and used.

One of the key objectives of our framework is to help users find an "optimal" subset of models in a model space in an interactive fashion, as opposed to automatically determining a design that is deemed 'optimal' by the system. In our framework, a model space is defined by the realizations and the levels of model granularity. The definition of optimality of selected models is subjective to the user who uses the framework, even though some objective functions, such as quality of a simulation service, can be used to tell if a model selection is optimal.

To achieve the objective, we provide a workflow in the framework. The foundation of the workflow is what we call *hierarchical DSE*. The foremost idea of hierarchical DSE is to explore a model space hierarchically with one design candidate. Exploration of the model space is an iterative process. Each iteration corresponds to a level of model granularity in the model space. The exploration starts at the highest level of model granularity when a user specifies an intitial selection of models. The selected models are "evaluated" using the reservoir simulators available for that level of granularity. Performance of the selected set of models is assessed using some objective functions, such as percentage of errors, quality of the model selection, etc. Performance values are used in the next iteration of model selection. The outcome of the exploration process is a subset of models which satisfies the time constraints and the users' requirements.

2.2. Components of the Framework

The services and tools provided by our framework are classified into four major categories: *user interface*, *integral subsystems*, *storage manager* and *simulation service*. The user interface allows a user to define a design space, manage a DSE session, and build utility functions for decision makings. The integral subsystems provide essential services such as data composition, DSE workflow and reservoir simulations. The storage manager provides services to read and write key data and metadata in a DSE process, e.g., simulation models, design specifications, DSE sessions, simulation results for each DSE session, and metadata in the system. Simulation service represents reservoir simulators and corresponding simulation models that are integrated into the framework.

Figure 1 shows a simplified diagram of the major components of the framework.

2.2.1. User interface

The main purpose of the user interface in the framework is to capture user inputs required by a DSE workflow. The user inputs related to a DSE workflow include design candidates and models selected specially for the workflow. Information on all available design candidates and simulation models can be accessed through the storage manager of the framework.

Utiltily builder is a module of the user interface that allows a user to quantify the influence of uncertainties in the reservoir and express the decision makers' risk attitude. The output of this module is an objective function which is formulated as the following: $F = \sum_i w_i \cdot f_i + r \cdot \sigma$, where f_i is the performance of the realization *i*, in terms of a performance metric, e.g., net present value, cumulative oil production, etc.; w_i is the weight assigned to realization *i*; *r* is the risk factor; σ is the standard deviation of outcomes of all realizations. Specifically, the weights for the realizations and the risk factor are what utility builder captures from the user.

2.2.2. Integral subsystems

DSE engine workflow is the central component of the framework. The workflow takes a design space specification as a key input, runs simulations dictated by the specification, and writes the simulation results into a central storage area of the framework. It is implemented using Windows Workflow Foundation (WWF) [10]. Figure 3 shows how the workflow is modeled in WWF. The core of the workflow is the evaluation of every selected design candidate with every selected simulation model. It is implemented with a nested loop, whose outside loop iterates selected design candidates and inner loop iterates selected simulation models.

Data composition in the framework is an operation which generates a working simulation model by modifying a base simulation model according to a design candidate. This operation is performed by reservoir simulation engineers in their daily work. It is mostly done manually and not a trivial task when dealing with large amounts of simulation models. However, it has not been sufficiently studied previously. In this framework, we provide a data composition service which can be easily integrated into a workflow or tool. For example, one of the activities in the DSE workflow engine in our framework is data composition, which can be seen in Figure 2. This activity essentially is a call to the data composition service from the framework.

2.2.3. Storage manager

Storage manager provides services to read and write the key data components in the framework. The key data include simulation models, session data, results for each DSE session and metadata in the system. Storage manager can be used to populate the framework with some initial data before the framework can be used for any DSE process. For example, the simulation models available for an asset should be checked into the framework initially. During the check-in process, the models could be copied into a private data storage area and the metadata (summary information) about the models is extracted and stored in a metadata catalog. Design strategies also need to be imported into the framework before they can be selected when specifying a design space. The storage manager can also be used to manage data produced as a result of each DSE session – e.g., the large amount of simulation results generated as each design point in the selected subspace is evaluated. These results need to be stored and managed properly so that they can be later retrieved and consumed by other tools for visualization, reporting, audit trail, knowledge management, etc.

2.3. A Typical DSE Workflow

Figure 3 shows a typical workflow that uses the framework in design optimization. The workflow consists of the activities for most DSE problems, including specifying a design space, running DSE, analyzing and visualizing DSE outputs. Among these activities, specifying those inputs can be conducted through design space designer and utility builder in the framework; running DSE is a call to DSE engine workflow. Since the activities after a DSE run are workflow specific and user specific, they might be carried out manually or with external systems outside of the framework.

3. Field Case

In this section, we present the application of our DSE framework to a well drilling scheduling problem in an off-shore oil and gas field. Simulation models from an existing oilfield were used for purposes of this case study.

Well drilling is a common process in oilfield asset development. There are two main challenges of drilling new wells: placement of the wells and scheduling of drilling the wells. Placement of wells is about where and how to drill new wells, while scheduling deals with an order in which new wells are drilled. Optimization of well placement has been investigated in many studies [4-7]. In this case study, we focus on the scheduling aspect. Specifically, we study how different well drilling schedules affect the cumulative production of the oil field and how we can use the DSE framework to facilitate the optimization of well drilling schedules.

3.1. Experimental Setup

In this work, we evaluate different drilling schedules for drilling 6 new wells in the oilfield, 3 producers and 3 water injectors. We use 20 unconditioned realizations of the two reservoirs. The objective function in this case is to maximize the cumulative oil production of the field for the next 20 years.

The evaluation is performed in the following steps. The first two steps make preparation for using the DSE workflow. After them, the designs required by the workflow are ready. The designs are stored in a format that is compatible with the DSE framework, and are loaded into the framework so that a user can start a DSE session.

Step 1: Enumerate drilling orders. All possible drilling orders are generated in this step. A well drilling order is different from a drilling schedule. The former specifies a sequence in which new wells are drilled, while the latter contains more information. For example, a drilling schedule specifies the time when a well is drilled and completed. Information such as what drilling rig is assigned to a well is also contained in a drilling schedule.

A customized graphical user interface is used to specify partial orders and partial rig assignments for drilling order candidates and is shown in Figure 4. The main purpose of the user interface is to reduce the number of possible drilling orders. It takes constraints such as partial orders and partial rig assignments as inputs. It outputs all drilling orders that satisfy those constraints. If the drilling orders are not constrained, the number of all possible drilling orders is $6! \times 2^6 = 46080$.

Step 2: Convert drilling orders into drilling schedules. In this step, drilling order candidates are converted into drilling schedules by a standalone application. In this application, the performance of two drilling rigs is modeled by a proxy. The proxy computes drilling time as a function of the length of a well being drilled. The downtime and the maintenance for the rigs are modeled by a workover schedule. A user can specify a workover schedule for each drilling rig before running the application. Besides a drilling order as generated in Step 1, the application takes as inputs the well placement data, e.g., surface location, target location, trajectory type, build angle, etc. Drilling schedules calculated by the application can be output in many formats and used for different purposes. For example, it can be used as Eclipse input data for well scheduling, or as Eclipse completion data. Most of all, it will be used as the input data for our DSE workflow.

Step 3: Create a DSE session. A DSE session is used to group the data components related to a DSE instance. The key data components contained by a DSE session include design space specification, DSE results and metadata about the session. The framework provides create, read, update, and delete (CRUD) basic operations for the DSE session entries. In creating a DSE session for our case, we 1) make a selection of the drilling schedules and simulation models. This is done through the design space builder in the framework; 2) specify a metric for evaluating a design performance and a risk factor, which is achieved by utility builder in the framework; 3) and configure the output of the DSE session. The outcome of this step is persisted into a central storage area for the framework.

Step 4: Evalute drilling schedules. To evaluate the drilling schedules, the DSE engine workflow in the framework is first used to run reservoir simulations. Outputs of the workflow are raw simulation data, i.e., forecast of production profile of this oilfield. These data are collected by the framework according to the configuration given by the users. The main purpose of the configuration is to filter the raw simulation data and extract data that the users are interested in. In this case, we configure the framework to collect production rates and cumulative production data at each time step. After the workflow is over, the next operation is an interactive process, where the user starts to apply various performance metrics and user-defined utility functions to the DSE outputs.

3.2. Results

For illustration purpose, we show the results of a DSE session. The DSE session is based on reservoir simulations at the highest level of model granularity in the framework, even though detailed reservoir simulation is readily supported by the framework. To evaluate the performance of each design candidate, we examine the cumulative oil production of the oilfield constrained by each of these designs (drilling strategies). Figure 5 shows a weighted average of cumulative oil production for eight of all design candidates. Each average is calculated among the 20 unconditioned realizations. Since these realizations are equiprobable, the weight for each realization is the same. The weights may change over time as particular models are history matched to actual production. Based on this figure, we can see that two designs (design #0 and design #1) result in lower cumulative production compared to the rest of the design candidates, while design #2 has the most production.

We then single out designs #0 and #2 for a detailed comparison. Figure 6 plots the cumulative oil production for the two design candidates for all 20 realizations. As can be observed, except for realization #14 and realization #15, design #0 is inferior to design #2 for all realizations. A conclusion can be drawn at this stage that based on high level simulations, design #2 is an overall superior strategy compared to design #0, under the uncertainties being modeled.

In an attempt to better understand a cause for the difference in performance between these two strategies, we examine the drilling schedules represented by the two design candidates. In design #0, two water injectors and one producer are open in the first year and the rest of the wells are open in the following two years. However, in design #2, three producers are open in the first year and the three water injectors are scheduled in the following two years.

Therefore, based on a high level analysis of the DSE results in this illustrative use case, we conclude that: 1) different drilling schedules lead to different production profiles and different cumulative oil production for an oilfield; 2) the order in which producers and injectors are drilled has an impact on the production of this field; and 3) for this particular field, the drilling schedule represented by design #2 appears to be a good strategy.

4. Concluding Remarks

This paper introduces a framework for design space exploration in oilfield asset development that allows for integration of tools and services required for performing DSE workflows. This framework supports multiple realizations that capture the range of uncertainty about the asset. The hierarchical design space exploration paradigm and workflow support helps the user to rapidly evaluate a large number of alternate development strategies for a large number of realizations through the use of high-level simulators based on proxy models, before identifying a smaller set of designs for further evaluation using successively fine grained simulations.

Our illustrative example uses reservoir models for a real offshore field to demonstrate the framework's potential for a systematic exploration of a broader set of drilling schedules to identify the optimal one. It allows a user to formally define a design space which includes all drilling schedules to be evaluated, realizations, and objective functions. The drilling schedules are prepared with the tools integrated into the framework. Through a DSE workflow service provided by the framework, a very large number of drilling schedules are used to drive numerical reservoir simulations. The DSE outputs, consisting primarily of 'raw' simulation results, are further aggregated and filtered using user-specified objective functions to assist in the identification of an optimal drilling schedule.

Future work includes applying this system to new design problems in oilfield asset development. By doing this, we expect to make the framework more flexible and scalable so that minimal effort is required for supporting any new design problem.

5. Acknowledgements

This research was funded by CiSoft – a joint USC-Chevron Center of Excellence for Research and Academic Training on Interactive Smart Oilfield Technologies (http://cisoft.usc.edu/). We are grateful to the management of CiSoft and Chevron for permission to present this work. We also acknowledge the contributions of Laurent Pianelo (Chevron) and Jing Zhao (USC) to the design and prototyping of the framework described in this paper.

6. References

[1] Haupt, T., Voruganti, A., Kalyanasundaram, A., & Zhuk, I.: *Grid-Based System for Product Design Optimization*. 2nd IEEE International Conference on e-Science and Grid Computing,, Amsterdam, the Netherlands, 2006

[2] Künzli, S., Thiele, L., Zitzler, E.: *Modular design space exploration framework for embedded systems*. IEE Proceedings - Computers and Digital Techniques 152, 183–192 (2005)

[3] Bittencourt, A.C., Horne, R.N.: *Reservoir development and design optimization*. In: SPE Annual Technical Conference and Exhibition, San Antonio, Texas (1997) SPE 38895.

[4] Guyaguler, B., Horne, R.N.: *Uncertainty assessment of well placement optimization*. In: SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana (2001) SPE 71625.

[5] Yeten, B., Durlofsky, L.J., Aziz, K.: *Optimization of nonconventional well type, location, and trajectory*. SPE Journal 8 (2003) 200-210 SPE 86880.

[6] Yeten, B., Durlofsky, L. J., and Aziz, K. *Optimization of nonconventional well type, location and trajectory*. SPE 77565. SPE Journal, v. 8, 200-210 (Sept.)

[7] Parashar M, Matossian V, Bangerth W, Klie H, Rutt B, Kurc T, Catalyurek U, Saltz J, Wheeler M. *Towards dynamic data-driven optimization of oil well placement*. In: Sunderam V et al (eds) Proceedings of the workshop on distributed data driven applications and systems. International conference on computational science 2005 (ICCS 2005), vol 3514–3516, Springer, Berlin Heidelberg New York, pp 656–663

[8] Parashar, M., Klie, H., Catalyurek, U., Kurc, T., Matossian, V., Saltz, J., Wheeler, M.: *Application of grid-enabled technologies for solving optimization problems in data-driven reservoir studies*. The International Journal of Grid Computing: Theory, Methods and Applications (FGCS) 21 (2005) 19-26.

[9] Cullick A.S., Heath D., Narayanan K., April J., and Kelly J., *Optimizing Multiple-Field Scheduling and Production Strategy with Reduced Risk*, SPE 84239, 2003

[10] Windows Workflow Foudation (WF), http://wf.netfx3.com/

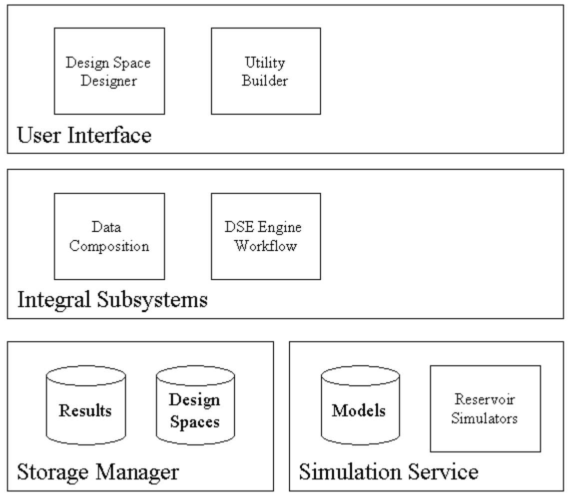

Figure 1: Architecture of the DSE framework

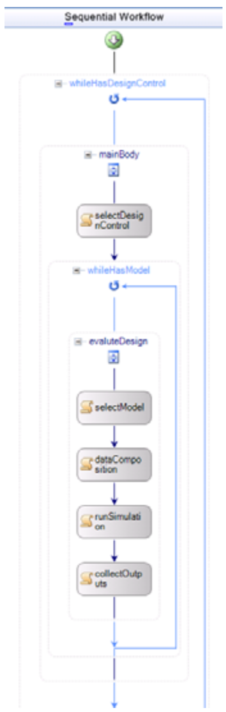

Figure 2: Core DSE engine workflow

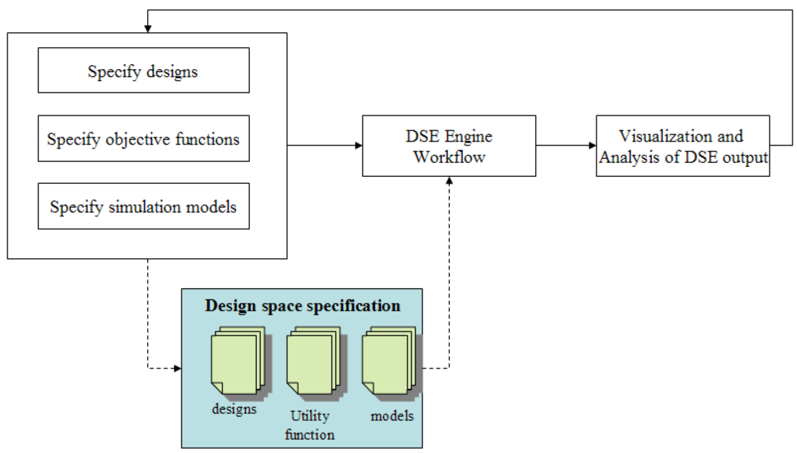

Figure 3: A typical workflow

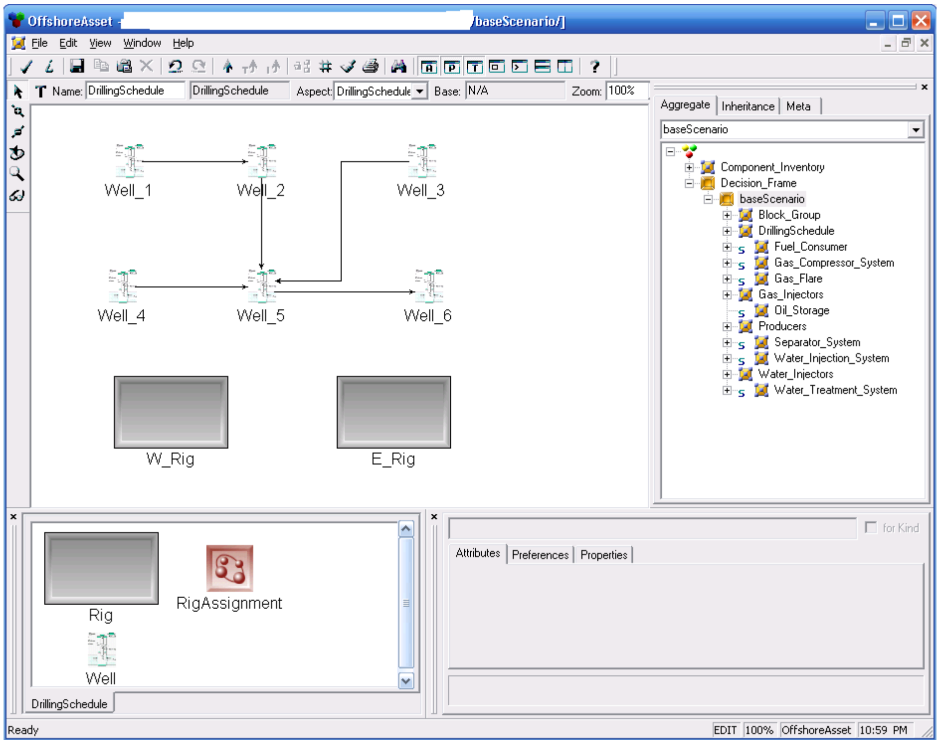

Figure 4: Constraing drilling order candidates

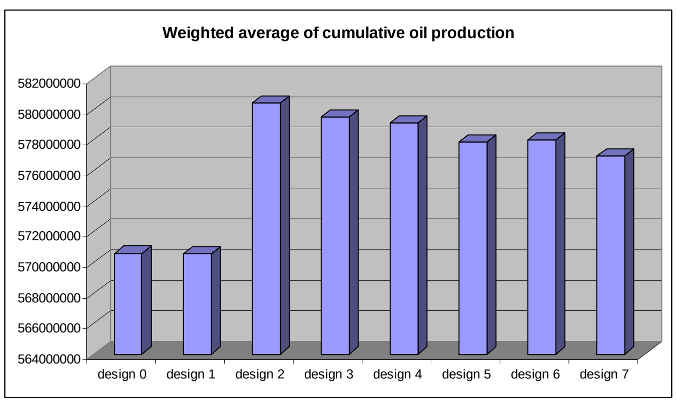

Figure 5: DSE results

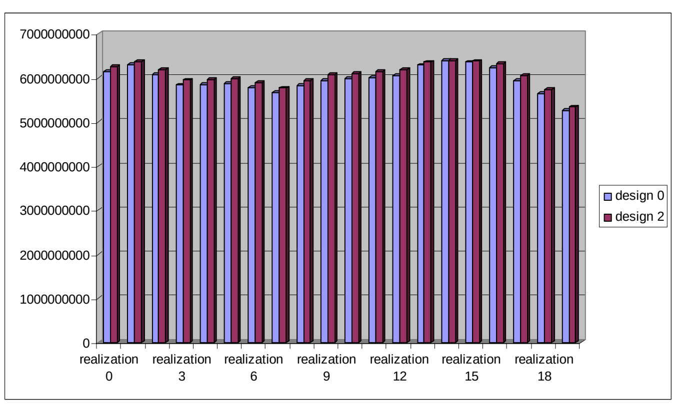

Figure 6: Comparing two design candidates across all realizations