SPE-132629

Intelligent model management and Visualization for smart oilfields

Charalampos Chelmis[1], Amol Bakshi[3*], Burcu Seren[2], Karthik Gomadam[3], Viktor K. Prasanna[3]

[1]Department of Computer Science, University of Southern California, CA, USA.
[2]Chevron Corporation, Houston, TX, USA.
[3]Ming Hsieh Department of Electrical Engineering, University of Southern California, CA, USA.

This paper was prepared for presentation at the SPE Western Regional Meeting held in Anaheim, California, USA, 27–29 May 2010.

Abstract

Simulation models are commonly used as an aid to decision making for oilfield development and operations. Such models represent information uncertainty and alternate operational strategies and form a design space, which is used by engineers to explore different “what-if” scenarios. As the composition of the engineering team changes over the lifetime of the oilfield and new modeling requirements emerge, it becomes important for engineers to be able to quickly “mine” from and understand the distribution of the models in the design space, and also to know if a particular scenario was already modeled in the past or if a new model needs to be created.

In this paper, we describe a technique to analyze arbitrarily large set of simulation models, identify similarities and differences between model parameters, and automatically cluster the models based on similarity in an n-dimensional design space. The major contribution of this work is a vector space based approach for automatic model clustering without human intervention. Building on this contribution, our system provides a smart browse, search and visualization capability over a legacy model catalog in a non-proprietary manner. The system also performs automatic analysis of available models in order to discover their underlying basic structure and, if possible, represents the models as variations of this basic structure. We demonstrate the application of our algorithm to a set of Integrated Production Modeling (IPM) models. However, our use of a standard, non-proprietary network model abstraction as an intermediate representation means that our analysis technique can be applied to models created using a variety of modeling and simulation tools.

The broader significance of this work is in the context of knowledge management for smart oilfields, specifically focused on extracting meaningful information from legacy simulation models, and making this information available and useful to the domain experts.

1. Introduction

Integration and optimization of asset development and operations from reservoir to wellbore to surface offers the potential for increased efficiency and recovery. An integrated solution requires consideration of all constraints; where engineers make use of the what-if scenarios to analyze all constraints from reservoir boundaries to bottom-hole to wellhead to surface. Simulation models are common ways to perform “what-if” scenarios that play an important role in oilfield development [Moitra2007] since their primary usage is to discover the impact of different parameters on the forecast [Chow2003]. To find the best development strategy, engineers create many different models containing parameter sets that are either completely disjoint or can be partially overlapping.

Model creation and maintenance presents three significant challenges:

1. In order to construct the models, engineers have to be experts in specific legacy applications, since different applications are used for the creation and analysis of different models.
2. As the volume of different models grows, it becomes difficult for engineers to manually compare models side by side.
3. When new engineers join the oilfield team, it becomes important for them to quickly familiarize themselves with the existing models.

* Author currently at Chevron Corp., Houston, TX, USA

In this paper we describe a technique to build cataloging and searching functionality that will allow domain experts to quickly analyze the information within the entire set of models, group and filter the models based on their parameters, and compare desired outputs for selected models. We also present a smart system that makes use of this technique to analyze arbitrarily large set of simulation models, identify similarities and differences between model parameters, and automatically cluster the models, as well as present all extracted information to domain experts in a meaningful and useful way. We demonstrate the use of the system using a set of IPM models.

This paper is organized as follows. In Section 2 we describe our technique, including a discussion about fundamental background technology and major components of our implemented system. We then demonstrate the use of our system by applying it to an illustrative use case in Section 3. We conclude in Section 4.

2. Intelligent Model Management

Our goal is to build cataloging and searching functionality that will allow engineers to quickly analyze the information within the entire set of models, group and filter the models based on their parameters, and compare desired outputs of selected models. In doing so, we address scalability concerns pertaining to (i) end user should be able to select any number of clustering options, (ii) the metadata extraction from the models should not be limited to a pre – specified parameter list but should scale if the model parameterization changes.

2.1 Theory

Petroleum Experts General Allocation Package (GAP) is a legacy tool, part of the IPM suite of tools [PETEX-IPM], which is often used for Complete Surface Production / Injection Network Modeling and Production Forecast. Production and injection systems are created and viewed by engineers using the GAP user interface. Properties of system equipment are available for viewing and editing through the GAP user interface as well. A very simple production network model created with GAP is presented in Figure 1.

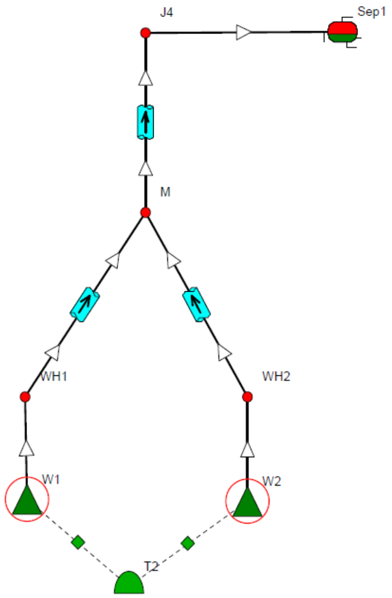


Figure 1. Simple GAP model

Engineers can construct different production/ injection models by adding and/ or removing equipment, or by changing the parameter values of existing equipment. For example, engineers can investigate the impact on the overall forecast prediction caused by the addition of a new well, W3, to the production system presented in Figure 1. Similarly engineers can investigate the potential impact of using different production schedules for wells W1 and W2.

As a common practice, engineers usually create a basic model of the oilfield and later explore different “what – if” scenarios by modifying the basic model. The basic model, in most cases, illustrates the overall oilfield production / injection systems, rather than smaller components. The models consist of different equipment components, the parameters of which must be manually specified. The modification of these parameters in a number of models makes the tracking of changes extremely difficult, if not impossible.

An oilfield IPM model catalog can be viewed as a set of variations of a basic initial model. The basic model can be treated as a template representing the minimum common / unchanged information across all available models. All other can be described as variations, called deltas, of this template. The challenge is to identify the one or more templates from the collection of models available in a model catalog and outline the parametric differences between the different models.

Let us consider a catalog containing three GAP models, two of which differ in pipe sizing, and two differ in separator pressure. The challenge is to:

1. Generate a template that represents what is common / unchanging among the 3 models;
2. Identify pipe sizing and separator pressure as the two parameters that differ;
3. Describe the first model as (template + pipe size value x + separator pressure y),
   Describe the second model as (template + pipe size value z + sep pressure y) and
   Describe the third model as (template + pipe size value z + sep pressure k).

Depending on the set of models and the similarity between them, more than one template may be produced. The goal is, therefore to discover the template(s) that offers maximum coverage across a catalog and minimize the total number of templates and extra information stored in deltas. Note that all this processing is to efficiently extract information from GAP models in order to address the challenge of model analysis. It is not our intention to use this information, templates and deltas, to regenerate models.

2.2 System Components

Our system has three major components: user interface, extracting module (parser), and storage system. The user interface allows users to browse the models, perform real time multi-dimensional model clustering and filtering, as well as visualize the model design space. The parser is responsible for the extraction of relevant metadata by parsing GAP models, analysis and grouping of the models. The storage system acts as the repository for the extracted information.

Figure 2 illustrates a high-level overview of our system components.

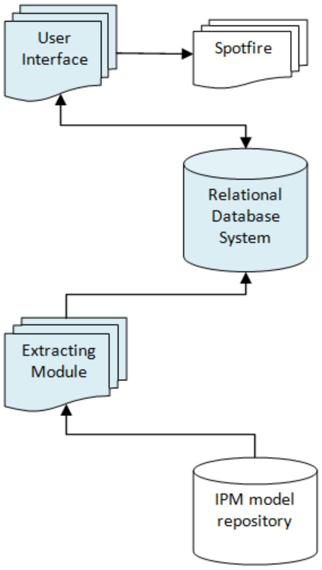


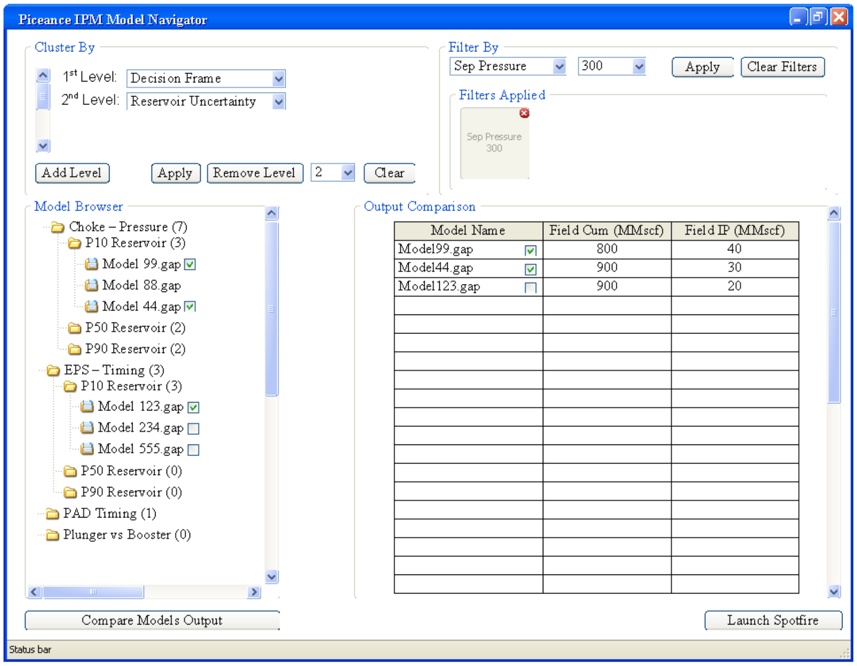
Figure 2. System Architecture

2.2.1 User Interface

The user interfaces allows users to perform smart and powerful model browsing, real time multi-dimensional model clustering and filtering, and visualize the model design space. Figure 3 shows a screenshot of the user interface.

Users can either select a single model or a group of models. In both cases models are parsed, metadata about them is extracted and analyzed, enabling the grouping of models. The parsed models are displayed in a tree hierarchy in the model browser. This view allows users to apply different clustering parameters and dynamically cluster the models in real-time.

When no filters are applied, all the available models are displayed in the model browser. Users can apply filters, such as a separator pressure of 300 PSI, to identify a set of models that are of interest. Similar to clustering options, multiple filters can be applied or removed dynamically. The model browser is updated each time a filter is set / unset to reflect the change.



1st Level: Decision Frame
2nd Level: Reservoir Uncertainty
2

Filter By: Sep Pressure 300

Filters Applied: Sep Pressure 300

Model Browser:

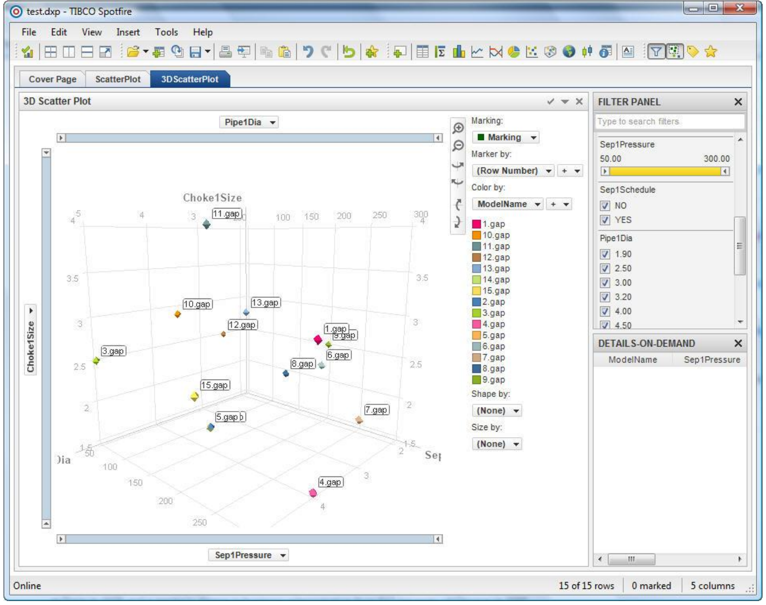
- Choke – Pressure (7)
  - P10 Reservoir (3)
    - Model 99.gap ☑
    - Model 88.gap
    - Model 44.gap ☑
  - P50 Reservoir (2)
  - P90 Reservoir (2)
- EPS – Timing (3)
  - P10 Reservoir (3)
    - Model 123.gap ☑
    - Model 234.gap □
    - Model 555.gap □
  - P50 Reservoir (0)
  - P90 Reservoir (0)
- PAD Timing (1)
- Plunger vs Booster (0)

Model Name		Field Cum (MMscf)	Field IP (MMscf)
Model99.gap	☑	800	40
Model44.gap	☑	900	30
Model123.gap	□	900	20

Figure 3. User Interface

To further investigate the output parameters of the models that are of interest, users can select and compare models in the output comparison region of the user interface. The application supports scalar visualization and plugs into the Tibco Spotfire visualization system to enable a richer comparison. [TibcoSpotfire].

Users can select the automatic clustering visualization option to visualize the model design space using the Tibco Spotfire visualization tool. An example of automatic clustering visualization of parsed models is shown in Figure 4 below.



Marking: Marking
Marker by: (Row Number)
Color by: ModelName
1.gap
10.gap
11.gap
12.gap
13.gap
14.gap
15.gap
2.gap
3.gap
4.gap
5.gap
6.gap
7.gap
8.gap
9.gap
Shape by: (None)
Size by: (None)

Sep1Pressure 50.00 300.00
Sep1Schedule ☑ NO ☑ YES
Pipe1Dia ☑ 1.90 ☑ 2.50 ☑ 3.00 ☑ 3.20 ☑ 4.00 ☑ 4.50

Figure 4. Automatic Clustering Visualization

2.2.2 Extracting Module (Parser)

In order to extract metadata about available models, we propose a two-phase technique. In the first phase, we parse the IPM models and generate an intermediate representation. The intermediate representation is independent of the underlying data representation or the parser that was employed. In the second phase we apply the clustering algorithms to the intermediate representation. The outcome of the clustering is not affected by the order in which the models are processed.

During the first phase, all indicated IPM models are parsed using the IPM OpenServer API and useful information about them is extracted. During the second phase, models are represented as vectors. This is done in steps described in Table 1, below:

Step 1.	Extracted information about models is scanned in a one model at a time basis. The purpose of this initial scanning process is to identify a unique list of parameters shared among models. Using each of these parameters to form a new axis, an n–dimensional space can be constructed in such a way that all parsed models can be represented in this space [Lerman-1999].
Step 2.	A weighted vector is created for each model, so that each model can be completely represented in the n-dimensional space by this vector.
Step 3.	Working on the vector representation of models [VS-Wikipedia], an upper right similarity matrix is computed. The cells of this matrix contain the similarity value between two models in the n – dimensional space. For instance, cell (i, j) contains the computed similarity value between models i and j. The greater the similarity value is, the greater the similarity between two models. Note that the matrix need only be upper right, since if the similarity value for models i and j is computed and i < j, then the similarity value that would be stored in cell (j, i) is equal to the value which is already stored in cell (i, j).
Step 4.	In this step, models are organized in clusters. Two models belong to the same cluster if their similarity value is greater than a predefined threshold. Each model may belong to one cluster only.
Step 5.	Finally, a template is created and stored for each cluster by identifying the minimal common / unchanged information between all cluster members. Each model can then be described by only storing those metadata that differ (delta) from its corresponding cluster template.

Table 1. Second Phase steps

To compute the similarity between two models, we calculate the Euclidean distance between their vector representations. This distance is computed by measuring the cosine of the angle between the two vectors (v_1, v_2) using the formula, $cos\theta = \frac{v_1.v_2}{||v_1||.||v_2||}$

Adopting the vector space approach makes it easier to compute the distances between models, along different parameters. The first step is to represent the documents, in this case the models, in a numeric representation that is amenable to clustering algorithms [Begelman-2006].

Many solutions have been proposed [Steinbach-2000] for the problem of document clustering. However, the problem of model clustering is unique because of the following reasons:

1. No term can be discarded. Traditionally, clustering techniques analyze the document corpus in order to remove any unnecessary terms, usually terms that do not differentiate documents by a high degree. However, in our case none of the extracted metadata can be discarded.
2. Latent Semantic Analysis [Deerwester1990] techniques are often used in document clustering approaches in order to uncover common patterns of word usage across a large number of documents. However, in our case the vector representation of models contain same or similar parameter-value pairs. Therefore, latent semantic analysis does not bring any significant value.
3. Clustering techniques are usually used in order to create hierarchical clusters of available documents. In our case however, the intention is to provide dynamic clustering options without any predefined order or hierarchy in the clustering choices.
4. The number of produced clusters is not known prior and must be determined in real-time, without preprocessing of the model corpora and human intervention. Clustering algorithms that have been proposed in the literature require a specific number of clusters to be produced before the start of the clustering process.

2.2.3 Storage system

The extracted information from the models is stored in a relational database. The selection of the database does not have an impact on the final result.

The extracting module manages the database content. When new models are parsed, the database is updated in order to reflect the current information obtained from the model, along with clustering information, produced templates, and deltas. This information is accessed by the user interface to realize the functionality described earlier in the paper.

Once the extracted information is stored in the database, access to the original set of IPM models is no longer required, unless additional models are created or existing models are updated and additional metadata is to be extracted. If the database resides on a local drive, it can be copied from one machine to another and the system can be seamlessly used by another set of users. Storing the information on a network drive eliminates the need of copying as the system can be simultaneously accessed by multiple users.

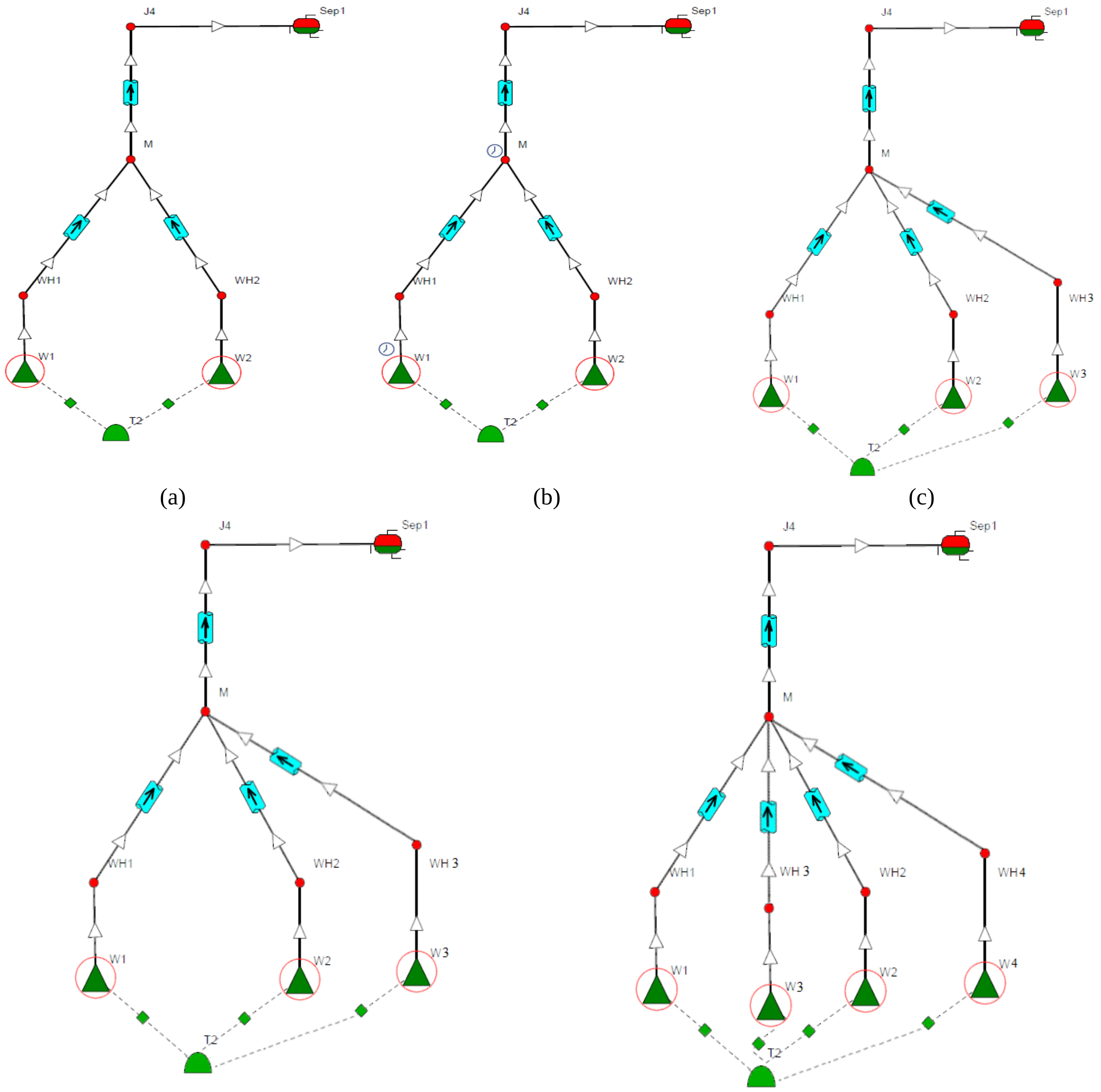
3. Use Case

In this section we present the application of our technique to an IPM model corpus consisting of five network production models. All five models are shown in Figure 6 below. We note here that these are synthetic models and were created by studying models existing in oilfields.

3.1 Experimental Setup

In this work, we evaluate our technique using five network production models. The objective of our evaluation is to demonstrate the accuracy of our clustering algorithm.

Model (a) is the basic model. Model (b), while being structurally same as model (a), has associated schedules with some of its equipment. A schedule is the number of operations to be performed in a future time and is represented by a small clock beside the equipment with which the schedule is associated with. Well W1 and manifold M have associated schedules in model (b). Model (c) is a new version of model (a) where a new well, W3, is added to investigate the change in performance of the production system, due to the drilling of the new well. Model (d) is the same as model (c) but with some parameter differences, such as the separator Sep1 pressure. Finally, model (e) is similar to the model (a) but has two more extra wells.



(a) (b) (c)

(d) (e)

Figure 6. (a) Basic Model, (b) Basic Model with Schedules, (c) Basic Model with an Extra Well, (d) Basic Model with an Extra Well and with Different Properties, (e) Basic Model with Extra Wells

3.2 Results

The outcome of the clustering algorithm is illustrated in Table 2.

Cluster 1:	Model (c), Model (d)
Cluster 2:	Model (a), Model (b)
Cluster 3:	Model(e)

Table 2: Results

As expected, models (a) and (b), which have no structural differences, are grouped together. Similarly, models (c) and (d) are grouped together as well. Model (e) however is considered to be dissimilar to the other groups and therefore forms a new group by itself. Users can provide their own clustering parameters to perform real-time clustering of models using the user interface. However, users can review the model clustering as proposed by our technique in order to get a very quick idea of the design space and to identify the models that have already been processed.

4. Concluding Remarks

This paper introduces an efficient approach to analyze arbitrarily large set of simulation models, identify similarities and differences between model parameters, and automatically cluster the models based on their similarity in an n-dimensional design space. The broader significance of this work is in the context of knowledge management for smart oilfields, specifically focused on extracting meaningful information from legacy simulation models, and making this information available and useful to the domain experts. Our technique provides the ability to perform multi-dimensional data comparison, discover relationships between model parameters and compare them to the actual underlying physical infrastructure.

5. Acknowledgments

We acknowledge the contributions of CiSoft, (Center for Interactive Smart Oilfield Technologies), a Center of Research Excellence and Academic Training. CiSoft is a joint venture between the University of Southern California, Los Angeles, CA and Chevron Corporation.

6. References

1. Petroleum Experts Integrated Production Modeling, http://www.petex.com/products/?id=71
2. C.V. Chow, J.C. Mothershead, D.L. Cox, Case History: Field Development Design Using A Risk-Based Integrated Production Model, Journal of Petroleum Technology, vol. 55, no 1, pp 61 – 67, January 2003.
3. S.K. Moitra, Subbash Chand, Santanu Barua, Deji Adenusi, Vikas Agrawal, A field-wide integrated production model and Asset management system for the mumbai high field, Offshore Technology Conference, Houston, Texas, USA Apr 30 – May 3, 2007.
4. Michael Steinbach, George Karypis, Vipin Kumar, A Comparison of Document Clustering Techniques, KDD Workshop on Text Mining, 2000.
5. Ruben Tous, Jaime Delgado, A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment, In Proceedings of the 17th International Conference in Database and Expert Systems Applications, DEXA 2006, Krakow, Poland, September 4-8, 2006.
6. Grigory Begelman, Philipp Keller, Frank Smadja, Automated Tag Clustering: Improving search and exploration in the tag space, Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland, 2006.
7. Jain A. K., Murty M. N., Flynn P. J., 1999. Data clustering: a review. *ACM Comput. Surv.* 31, 3 (Sep. 1999), 264-323.
8. Vector Space, http://en.wikipedia.org/wiki/Vector_space
9. Kristina Lerman, Document Clustering in Reduced Dimension Vector Space, http://www.isi.edu/~lerman/papers/Lerman99.pdf, 1999.
10. Tibco Spotfire, http://spotfire.tibco.com/

11 Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science, vol 41, pp 391 – 407, 1990.