The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-21827-9_45

Plan Recommendation for Well Engineering

Richard Thomson[2], Stewart Massie[1], Susan Craw[1], Hatem Ahriz[1], and Ian Mills[2]

[1] IDEAS Research Institute,
Robert Gordon University, Aberdeen, UK
{s.massie,s.craw,h.ahriz}@rgu.ac.uk
[2] Exceed Ltd, 12 Bon Accord Square, Aberdeen, UK

Abstract. Good project planning provides the basis for successful offshore well drilling projects. In this domain, planning occurs in two phases: an onshore phase develops a project plan; and an offshore phase implements the plan and tracks progress. The Performance Tracker applies a case-based reasoning approach to support the reuse of project plans. Cases comprise problem parts that store project initiation data, and solution parts that record the tasks and subtasks of actual plans. An initial evaluation shows that nearest neighbour retrieval identifies projects in which the retrieved tasks and subtasks are relevant for the new project. The Performance Tracker can be viewed as a recommender system in which recommendations are plans. Thus the data that is routinely captured as part of the performance tracking during offshore implementation is utilised as experiences.

Keywords: Case-Based Reasoning, Recommender Systems

1 Introduction

The oil and gas industry is one of the world's largest industries and is estimated to be worth $10400 trillion, based on current discovered oil reserves and the average price of oil. In the oil and gas industry, wells are holes produced by boring for the purpose of finding and producing hydrocarbons. Wells have various categorisations and are constructed either to obtain geological data prior to drilling (exploration well, test well and appraisal well), or to research possible oil fields (wildcat well), or to extract the raw materials from the ground (oil well, gas well, production well, aquifer producers and gas injectors).

This paper considers the reuse of project plans for subsea drilling. Offshore wells are constructed using rigs with various types of equipment used for drilling, casing the hole and extraction. The process of drilling a well can be split into 5 segments: planning where the tasks and subtasks required to construct the well are identified; boring the hole to reach the reservoir; preparing the hole for the extraction of the hydrocarbons by casing the hole with cement; extracting and refining the hydrocarbons; and lastly plugging the well when the reservoir is empty or the reservoir has stopped producing enough hydrocarbons to be seen as a viable use of resources.

Due to the contractual nature of employment within the oil industry, knowledge retention can be challenging and hiring individuals with the experience necessary is expensive, as a result the cost of retaining corporate memory is high. The capture and reuse of knowledge using a centralised system can help reduce this cost.

The rest of the paper is organised as follows. Section 2 explains the process of planning and monitoring a subsea well construction project. Section 3 describes how the Performance Tracker will use the data produced during the planning process to identify similarities between projects. The architecture of the Performance Tracker is described in Section 4. In Section 5 the effectiveness and performance of the approach is investigated. Related work is discussed in Section 6. Lastly, Section 7 reflects on the work done, highlighting our conclusions and planned future developments.

2 Problem Domain

Well construction projects often share some characteristics with related previous projects and, as a result, follow the same drilling process. It is therefore possible to reuse project plans of related projects in order to save time in the planning process. The planning process is split into two stages.

Onshore Planning Stage: The project plan is developed by the onshore team. At this stage a new well has been identified and approval for drilling has been received. A new project plan is developed after which, the plan is scrutinised during a “Drill Well on Paper” exercise where the type of rig is chosen, the project budget is calculated and potential causes of non-productive time are identified. The final project plan is then created, comprising the list of tasks and their associated subtasks, containing target times. This plan is then ready to be used during the second phase of the process.

Offshore Monitoring Stage: The plan is implemented by the offshore team where data is monitored and recorded. The project plan changes its primary function from a planning tool to a monitoring tool. The offshore project manager will input the operational drilling data. Where there is a discrepancy between the planned task times and the actual task times, it is classified as either invisible lost time which states inefficiency within the well drilling operation, or non-productive time which is time spent rectifying unforeseen problems during the operation e.g. tool failures. The project plan is then refined taking into account any lost time encountered. Once the project has been completed, the project plan is used to evaluate the project before being retained by the company.

3 Performance Tracker

During the onshore process it is common practice for the onshore team to employ the time consuming process of manually retrieving old project plans and tailoring these to suit the new project. Currently it is up to the onshore team members to recall previous project plans, based on their own past experiences. This can be problematic as individuals may forget about potentially suitable projects or may be unable to obtain plans located on a local machine. The project team may also have an incomplete understanding of the project resulting in a poor plan selection. Furthermore, an inexperienced team may lack past knowledge to effectively reuse project plans. The Performance Tracker addresses these issues by supporting the user during the onshore process with a CBR recommendation function for the retrieval and reuse of past project plans from a central project plan repository. This approach uses CBR to both reduce the time required for selecting suitable project plans and to provide a more informed set of potentially relevant plans.

3.1 Case Representation

A project case *c* is made up of a problem part *p* and a solution part *s*. Problem *p* contains the feature values of the project initiation data that will be used to identify the similarities between cases. The project initiation data stores the core project information required to begin the planning stage, consisting of a project description, the geographical location of the well and the planned drilling depth. Problem *p* is made up of five base features: a textual *description* of the project; the selected *rig*; the *well* to be drilled; the estimated drilling *depth* in feet; and the estimated *duration* in days. The *rig* and *well* are represented by a set of sub-features combining to give the overall representation in Figure 1.

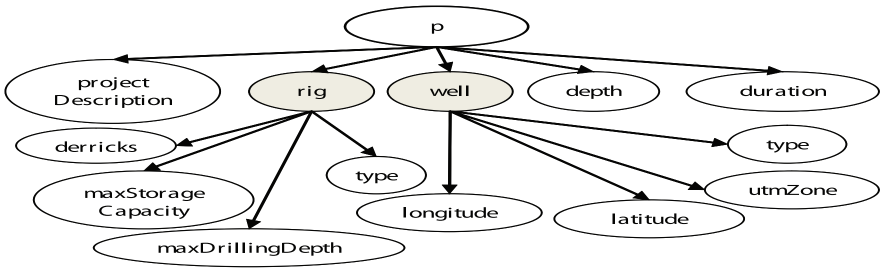

Fig. 1. Project problem structure.

Solution *s* contains tasks and subtasks required to complete the project (Figure 2). Each task is decomposed into a number of smaller subtasks that are used to plan and monitor a task in greater detail. The subtasks also allow any unproductive lost time to be captured and classified which provides useful knowledge for the refinement of future plans. A sample case is shown in Figure 3 where the left hand box contains the project problem features and the right hand contains a set of tasks making up the project solution.

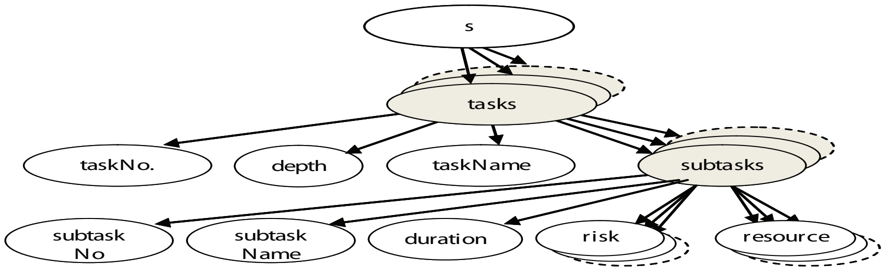

Fig. 2. Project solution structure.

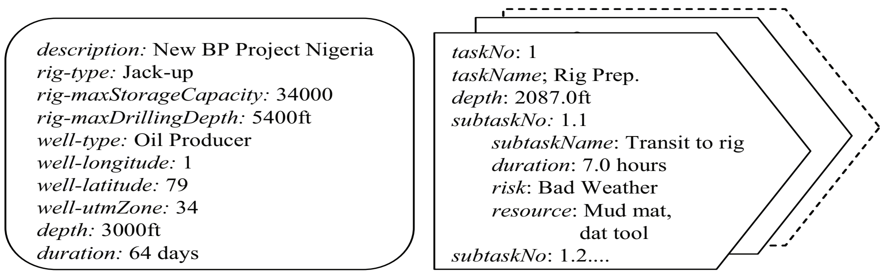

Fig. 3. Example of Project Initiation p and Solution s.

3.2 Similarity

Cases are selected by using k-Nearest Neighbour in which the case similarity score is determined by using a weighted feature average and the closest k cases are then recommended to the user.

Local Similarity The problem part of the case representation consists of numeric (*depth*, *duration*, *longitude*, *latitude*), symbolic (*rigType*, *wellType*) and textual features (*projectDescription*). The process for calculating the similarity for each feature type will now be outlined.

– *Numeric Similarity:* Similarity between numeric features is derived using Normalised Manhattan distance:

$$\text{sim(qv,cv)} = 1 - \frac{|cv - qv|}{r}$$

where qv is the numerical feature value of the query, cv is the numerical feature value of the case and r is the predefined feature range; e.g. the latitude feature will have values ranging from -90 to 90, hence range r will be 180.

– *Symbolic Similarity:* An oil rig may be suitable for various project types. Therefore a Boolean similarity of 1 for a match and 0 for no match will not be suitable. A similarity matrix is used to determine the similarity between different *rigType* and *wellType* values. The *rigType* matrix in Table 1 was developed in conjunction with a domain expert and shows the similarities between different rig types. This matrix serves as a look-up table to provide the similarity between *rigType* feature values. A similar process is applied to identifying the similarity between *wellType* values.

Table 1. *rigType* similarity matrix.

	Jackup	Semi Sub	Platform	Drillship
Jackup	1	0.8	0.7	0.6
Semi Sub.	0.8	1	0.9	0.8
Platform	0.7	0.9	1	0.5
Drillship	0.6	0.8	0.5	1

– *Text Similarity:* The *projectDescription* gives an overview of which company the project is for and any special conditions, such as "test deep water project" or "HSE recovery". The completed *projectDescription* is treated as a bag of words, as the presence of a word is of more importance than word position. The Jaccard Coefficient which assesses the overlap of words within the two word sets is used. *projectDescription* similarity is defined as:

$$\text{sim(qv,cv)} = \frac{|cv \cap qv|}{|cv \cup qv|}$$

Due to the industry specific nature of terms used within documents and the more personalised approach to shorthand (for example CC, circ and ccution are all used in place of circulation), the use of a generic lexicon such as WordNet would prove to be unsuitable. For this reason a domain specific lexicon has been created to analyse similarity and meaning behind these industry specific terms.

Global Similarity The global similarity *GSim* of a case is assessed by calculating the weighted average of the local feature similarities. The feature weights are set to give "more important" features greater influence. The global similarity function is shown below, where w_i is the weight of the i^{th} feature and $sim_i(q,c)$ is the local similarity of the i^{th} feature of the query q and of the case c.

$$\text{GSim(q,c)} = \frac{\sum_{i=0}^{n} w_i sim_i(q,c)}{\sum_{i=0}^{n} w_i}$$

The individual feature weights were set after consultation with domain experts and are shown in Figure 4. The features of the *well* and the *rig* are given higher importance. The depth will help determine the suitability of a *rig* for the project, as a *rig* will have a maximum depth that it can drill.

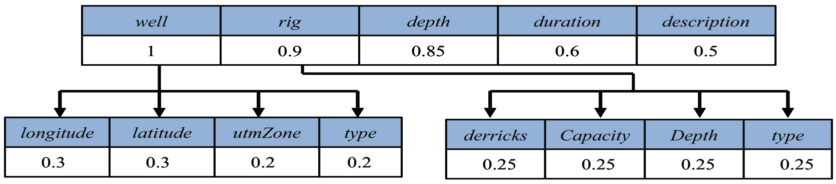

Fig. 4. Feature weights applied.

4 Architecture

The Performance Tracker architecture shown in Figure 5 consists of 2 main components, the onshore project plan recommendation tool and the offshore monitoring tool, both of which reside within a central server. The onshore project plan recommendation tool uses the JColibri2 framework in order to apply the modelling rules to the cases [1]. JColibri2 proved to be a suitable framework because it contains modules for both pre-processing text and calculating the similarity of the three feature types used by the Performance Tracker.

New project initiation data is sent from the web portal to the Case Modeller where modelling rules are applied prior to the connection with the case base. The Similarity Assessor applies the similarity metrics to identify the k most similar cases in the case base. The Solution Extraction Module can now take the solution part of the k cases and displays these as recommendations in the Web Portal. Once the user has adapted the project plan of the selected case, it is stored as a new project plan within the database. This ensures that the project plans in the database can be refined during the offshore monitoring process where task times are constantly being adapted to correspond with the live project data.

In the initial design the case base was built from the database for each query, however, retrieval times were unacceptably slow. In order to alleviate this problem the case base is now stored in server memory providing quick access for the onshore plan recommendation tool when a query is made. In order to ensure that both new projects created by the onshore process and refined projects generated by the offshore monitoring are consistent a weekly synchronisation process was developed, adding any newly created projects as new cases whilst updating current cases. The project case base stored in memory is used as the data source for case similarity matching and case retrieval.

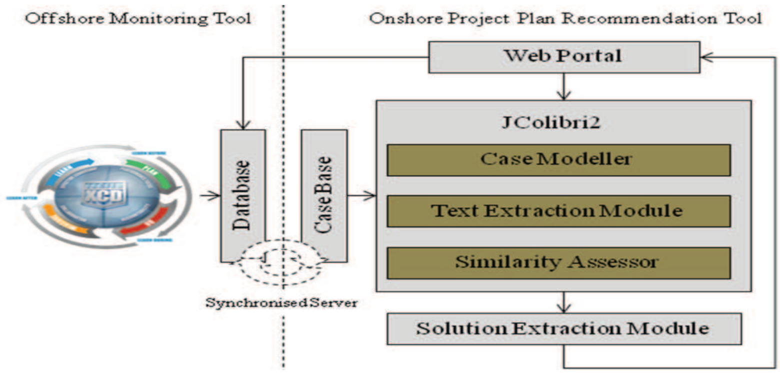

Fig. 5. Performance tracker architecture.

5 Evaluation

Testing is split into two parts. The first set of experiments investigates the quality of recommendations and describes the results of three recommendation tasks made using the Performance Tracker. The second set of experiments measures efficiency of the Performance Tracker in terms of retrieval time. The case base used for testing comprises 200 cases for 200 wells, drilled by 87 rigs in 9 locations. The drilling depths range from 958 feet to 23,060 feet.

5.1 Recommendation Quality

Three cases are extracted from the case base in turn, providing the project initiation data for three queries and leaving a case base of 199 cases. After each query a comparison of the problem and solution similarities is made between the query and the top 5 recommended cases. The quality of the recommended solutions is measured by the overlap between the tasks of each retrieved case and those of the query.

Query 1: The first query was based on an oil well construction project off the Nigerian coast. Table 2 shows that the 5 most similar cases retrieved were all based on the same well type. All of the project plans returned were very similar to those in the query, as shown by the high overlap values, but would require adaptations to the depth and task times in order to be suitable. The top ranked case contained an almost identical project plan to the extracted case with only one difference. The plan for both cases was the same; however, the retrieved project was refined to include a new task to address a lost time issue.

Query 2: The second query contained incomplete project initiation data,which is common in this domain, for an oil well project. The *projectDescription* and *well-type* attributes had missing data and, as a result, the similarity of the retrieved cases was low. The overlap of the retrieved plans fluctuated greatly, however 3 of the recommendations have an overlap of over 75%. It is evident

Table 2. Results for query 1.

Rank	Well Type	Initiation Data Sim (%)	Plan Overlap (%)
1	Oil Well	83%	97%
2	Oil Well	81%	84%
3	Oil Well	81%	80%
4	Oil Well	78%	81%
5	Oil Well	74%	80%

from Table 3 that the type of well being constructed greatly impacts the similarity value. Although the 2nd most similar retrieved case had a lower Project Initiation similarity value, the project problem was very similar in respect of *depth*, *well* and the *duration*. The low ranking was due to the dissimilarity between the *rig* and the *projectDescription* attributes. When a *projectDescription* was added to the problem, the second ranked case was promoted as the most similar case.

Table 3. Results for query 2.

Rank	Well Type	Initiation Data Sim (%)	Plan Overlap (%)
1	Oil Well	71%	79%
2	Oil Well	63%	81%
3	Wildcat	63%	44%
4	Gas Well	62%	77%
5	Test Well	53%	39%

Query 3: The third query used initiation data for the JADA oil field test well, containing a sizeable *projectDescription* with the phrases "test well", "JADA field" and "oil and gas producing well" which provides a larger vocabulary for evaluating similarity within the text matching module. Table 4 again illustrates that the type of well being constructed has a large impact on the project plan overlap. The top ranked case used the same rig to drill an oil well that was closely located to the query and this resulted in a high similarity of project initiation data. However, tasks only overlapped during the rig set-up and abandonment stages. The case ranked 2^{nd} contained the project plan for the test well of an oil field adjacent to the JADA field it is therefore not surprising that this case had a higher plan overlap value than the top ranked case.

5.2 Retrieval Speed

In this experiment we compare the Performance Tracker using weekly synchronisation, as described in Section 4, with a Baseline system that performs a retrieval based on the standard JColibri2 cycle with the case base being built for each query from the database. Twenty identical queries were run on the synchronised

Table 4. Results for query 3.

Rank	Well Type	Initiation Data Sim (%)	Plan Overlap (%)
1	Oil Well	87%	34%
2	Test Well	85%	94%
3	Oil Well	81%	34%
4	Oil Well	80%	37%
5	Test Well	74%	86%

and baseline systems to see the effect that the synchronisation process has on the retrieval times for a query.

The baseline system had an average retrieval time of 1 minute 34 seconds with times ranging between 1 minute 27 seconds and 1 minute 45 seconds. The synchronised system had an average retrieval time of 5.7 seconds. The fastest retrieval time was 2.15 seconds and the slowest retrieval time was 1 minute 34 seconds. The slowest retrieval time was the first retrieval as the case base had to initially be built from the database. Subsequent retrievals did not require the case base to be rebuilt and the retrieval times were much shorter. This experiment confirms that the design architecture adopted greatly improves retrieval times while the synchronisation process ensures the availability of revisions made during implementation.

6 Related Work

Case-based planning is not a new concept with applications ranging from holiday planners [2] to planning for logistics as used in CaPER [3]. However, the Performance Tracker has more in common with CBR recommender systems and applies the single shot, proposal type of recommendation described by Smyth [5]. This approach uses a specific user problem, and based on the user criteria, a set of cases are returned. Other similar systems such as Cobot [6] use a natural language conversational approach to extend a query by asking questions until conditions are met to provide a recommendation. The process in obtaining the user criteria is different but the end result of recommendation is the same.

Research has shown that a CBR methodology can be used effectively for reusing past experiences within a drilling environment, particularly within the context of lost time reduction. Skalle et al [7] identified the usefulness of CBR when reducing lost time by analysing one problem area, stuck drill strings. Although the research focused on one area, and primarily on research, it is very important as $250 million per annum is wasted on this form of downtime alone. Drill Edge [4] uses CBR to identify possible reasons for a lost time problem during the offshore monitoring stage of a project, and then advises users on how the project could be refined to solve the problem. Drill Edge builds on the studies carried out for Creek and TrollCreek knowledge intensive CBR frameworks [8]. TrollCreek was developed to identify lost time based on data from the “Drilling Club” [9, 10].

7 Conclusions

We have presented an approach to apply similarity matching to offshore well drilling project data to effectively recommend past project plans. We show that, by choosing appropriate project initiation features and feature weightings, it is possible to retrieve a set of suitable project plans that can be manually adapted for a new well construction project. Furthermore, the process of project plan refinement during the offshore monitoring stage has demonstrated that the project plan is constantly being updated and becoming a more robust solution, throughout the life of a project.

We also show that by storing the case base within server memory, rather than constantly extracting data from the database, retrieval times can be reduced. The weekly synchronisation with the database will enable the case base to remain up-to-date, as more refinements and projects are added to the database.

Drawbacks came when a user was required to manually adapt a small number of task depths of retrieved project plans. To limit the amount of adaptation required, the addition of a sea bed depth feature and a true vertical depth feature has been made. It was also shown that the type of well being drilled had a large impact on the suitability of a project plan and this may be emphasised in retrieval.

Acknowledgments. This partnership received financial support from the Knowledge Transfer Partnerships programme which is funded by the Technology Strategy Board along with other UK government funding organisations.

References

1. jColibri CBR Framework, `http://gaia.fdi.ucm.es/projects/jcolibri/`
2. S. Stewart, C. Vogt: A Case-Based Approach to Understanding Vacation Planning. Leisure Sciences, vol 21(2), pp 79–95 (1999)
3. High Performance Case-Based Planning, `www.cs.umd.edu/projects/plus/Caper/`
4. DrillEdge, `www.verdandetechnology.com/products-a-services.html`
5. B. Smyth: Case-Based Recommendation. The Adaptive Web, LNCS 4321, pp 342–376, Springer (2007)
6. S. Sahay, A. Ram: Conversational Framework for Web Search and Recommendations. ICCBR Workshop Proceedings, pp 161–170, Springer (2010)
7. P. Skalle, A. Aamodt, J. Sveen: Case-Based Reasoning, a method for gaining experience and giving advice on how to avoid and how to free stuck drill strings. IADC Middle East Drilling Conference, Dubai (1998)
8. A. Aamodt: Knowledge-Intensive Case-Based Reasoning in CREEK. 6th Int. Conference on Case-Based Reasoning, Workshop Proceedings, pp 62–71 (2005)
9. A. M. Islam, P. Skalle: Review of wellbore instability cases in drilling through case base resoning (CBR) method. Proceedings of the International Conference on Mechanical Engineering (2007)
10. P. Skalle, A. Aamodt: Knowledge-Based decision support in oil well drilling. Intelligent Information Processing II IFIP International Conference on Intelligent Information Processing, vol 163:443–455 (2005)