

A Real-Time Decision Support System for High Cost Oil-Well Drilling Operations

Odd Erik Gundersen, Frode Sørmo

Verdande Technology
Trondheim, Norway

Agnar Aamodt

Department of Comp. and Info. Science
NTNU, Trondheim, Norway

Pål Skalle

Department of Petroleum Technology
NTNU, Trondheim, Norway

Abstract

In this paper we present DrillEdge - a commercial and award winning software system that monitors oil-well drilling operations in order to reduce non-productive time (NPT). DrillEdge utilizes case-based reasoning with temporal representations on streaming real-time data, pattern matching and agent systems to predict problems and give advice on how to mitigate the problems. The methods utilized, the architecture, the GUI and development cost in addition to two case studies are documented.

Introduction

DrillEdge is a software system that provides real-time decision support when drilling oil wells. Decisions are supported through analyzing real-time data streams of parameters measured both on the surface and downhole when drilling. The real-time analysis identifies symptoms of problems, which are combined to provide best practices for how to handle the current situation. Verdande Technology has developed DrillEdge to reduce the cost and decrease the probability of failures in oil well drilling. Currently, DrillEdge continuously monitors around 30 oil well drilling operations in parallel for several customers and has been deployed commercially for two years. Verdande Technology's customers include Baker Hughes, Petroleum Development Oman and Shell, among others. In March 2011, Verdande Technology was awarded the Meritous Award for Engineering Excellence for its DrillEdge software platform by E&P Magazine.

More and more oil well drilling operators monitor a large part of their drilling operations in real-time from off-site Real-Time Operation Centers (RTOCs), which are located close to their experienced subject matter experts of various kinds. In this way the operators seek to provide remote support and ensure that knowledge can be transferred between shifts, operations and regions. Typically, less experienced personnel are located at the drilling rig and more experienced personnel are stationed in the RTOCs. The experiences gained from handling a problem at one rig can be utilized by the personnel in the RTOC when handling another problem later on at a different rig probably located in another region. Another advantage of RTOCs is cost reduction

as subject matter experts are centralized. Booth has documented the advantages and history of RTOCs (Booth 2011).

Monitoring a situation is to become aware of the situation, which is a main concern for decision makers. Situation awareness is investigated in depth by Endsley, and three levels of situation awareness are identified (Endsley 1995). First the elements of the situation are perceived, then the situation must be comprehended, and finally the future state of the situation can be projected. It is only after the decision maker is aware of the situation that a decision can be made regarding how to handle it. This of course applies to oil well drilling operations too. Experience with similar situations is valuable when making decisions. By recalling similar problems, different options for actions and risk assessment may be identified (Crichton, Lauche, and Flin 2005).

Reusing past experience, i.e., being reminded of similar situations and making use of decision steps made earlier, has turned out to be an efficient way for human beings to handle new situations. Case-based reasoning (CBR) is a computational method that is based on this principle. CBR compares the current problem with a set of concrete, past episodes called cases. The most similar cases are retrieved and used to solve the current problem (Aamodt and Plaza 1994). In DrillEdge, a case is a concrete drilling situation comprised of a collection of symptoms that lead to a problem with an advice for how to handle a similar situation. Cases are captured from actual historic drilling data and contain the operators' best practices for how to handle the situations.

The rest of the paper is organized as follows. In the next two sections we explain some core terms of oil well drilling and explain the type of problem addressed by the DrillEdge system. This is followed by three sections in which the rationale for our method choices are described, also describing the two main method types - Pattern recognition and CBR in more detail. Followed by a subsequent chapter on related work, the DrillEdge system architecture is described. Information about the system development process and costs are provided, and a status description of the system's deployment status is given. Finally two case studies of DrillEdge in real operation are summarized.

The Problem

The main task of drilling engineers situated remotely in a RTOC is to monitor and understand the situation on the oil

well drilling rigs in order to be able to support the rig crews if a problematic situation occurs. There are several problems related to that task.

Perceiving elements of the drilling situations is done through monitoring the real-time parameter graphs. For some drilling engineers, this means staring at graphs for 12 hours straight. This is of course a tiring and boring task, which might result in important symptoms and trends being overlooked. Therefore all elements of the situation might not be perceived.

In order to comprehend the current situation, all relevant information that describe the situation should be available. Even if the perception of elements is performed perfectly, as two or more shifts perform the monitoring task, important information might be lost because of rotating the shifts. Symptoms of problems that might have been deemed unimportant in the past might be crucial in understanding what is happening currently. If these seemingly unimportant symptoms were observed by another shift, but were not communicated, making the right decision in the current situation is hard.

Another problem is the huge amount of data provided by different sources. Understanding a problem is to build a mental model of it based on the relevant information. Often the problem for a drilling engineer is not that the right data is not available, but it is combining all the information or finding the right clue (Booth 2011). Some of the information is provided in real-time, like some physical models that are updated accordingly, while others need manual computation. Other information is stored in the drilling plan or daily drilling reports. Combining all the relevant information to get the full view constitutes a high workload and is not practically possible.

Being able to project the future status of the situations requires not only principle knowledge, but also experience. One of the reasons for moving expert knowledge from the rigs to RTOCs was to co-locate the experience. However, there is still unsatisfactory experience transfer in the oil well drilling industry, which easily leads to a lack of experience when it would be needed. The two main reasons for this is the age gap and the connection between personnel demands and the oil price. Because the oil price governs the personnel demands, lots of personnel are laid off when the oil price is low. The age gap describes the current distribution of personnel experience. Recently, the oil price has been high and the demand for personnel is high. Also, many are close to retirement age, and there are few people in between. Therefore the industry expects a gap in the experience needed.

Lack of experience can be mitigated by looking up best practices or incident reports that are relevant to the current situation. However, as searching for the right information in a data base or report often results in too many or too few relevant documents, this is rarely done.

Motivation for using AI technology

In the past decade, the focus in the oil & gas industry has been on collecting drilling data and making it available remotely in real time. This focus has seen the development of standards for transferring data as well as web-service based

standards for accessing databases. This work means that there is now a common platform to plug into data streams. However, applications that plug into this infrastructure to do useful things with the data have been limited, and initially focused on visualization tools to support manual interpretation of data in centralized real-time operation centers. The idea behind RTOCs is that if a problem develops, deep expertise from across the organization can be brought in to assist, rather than having to depend solely on people at the rig site. However, in order to bring expertise in on a well, the rig or the RTOC must first recognize that a problem is developing, and manually monitoring sensor data from a rig is a highly skilled and work-intensive task. It is difficult for even an experienced person to monitor more than a few operations at a time manually. This means that unless such monitoring can be partially or fully automated, the value of RTOCs cannot be fully realized.

The most obvious way to automate rig monitoring is to set more or less complex alarms that will notify human operators if some value or combination of values crosses a threshold. This type of technology exists, but is of very limited use because most symptoms are much more complex than what can be captured by such an alarm. Traditionally, the oil & gas industry has relied on physical models to generate expected data for many sensor values, such as down-hole pressure. These physical models are essential tools in planning a well, and much research in the industry has been directed towards making real-time versions of these models that are continuously updated and tuned with data from the well. This is a challenge because it involves such issues as inverse modeling of fluids, taking single point measurements of pressure and flow and working out the fluid dynamics of the whole system. Further complications arise as the physical properties of the formation are not completely understood in many cases, and differences in equipment cause differences in sensor responses. Still, drilling specialists are able to interpret the data to understand what is going on in the operation. In this situation, we are interested in investigating a data-driven AI approach to modeling the heuristics used by domain experts, rather than the use of full physical models. This approach has proved highly successful.

Our initial approach to the problem was to use instance-based reasoning and other machine learning approaches to predict problems directly. This very quickly proved difficult. First, there are relatively few examples of serious incidents. An incident such as stuck pipe, where the drill string gets stuck and part of the well has to be re-drilled can happen a few times a year. It is therefore necessary to have a system that is able to learn from very few examples, ideally a single case. The data dimensionality and dynamics increase the difficulty of the task. Although there may not be more than 10-20 parameters to monitor, it is not possible to detect symptoms of these problems without taking the dynamics into account, often looking at trends and the frequency of occurrence of events over the last 12-24 hours. Learning such dynamics directly from sensor data would be extremely difficult even with an abundance of cases, but to do so with only a few examples is clearly not feasible. The approach we have taken is to use a two-stage approach. First, pattern

matching agents are used to identify symptoms in the data, by forming abstractions over the sensor data, then CBR is used to identify if this set of symptoms has caused problems in similar situations in the past.

The pattern matching agents making up the first step of this process are designed so that each agent seeks to identify one type of symptom that the domain experts have identified as important in diagnosing a particular problem. For instance, one type of symptom that can be predictive of stuck pipe is the pack off (plugging of the wellbore around the drill string) symptom. The process of creating these agents is a classical knowledge acquisition process seen in expert systems in general. The domain experts are typically able to provide some pointers in what to look for, and can easily identify examples, but the expert is typically not conscious of all the heuristics he or she uses in identifying the pattern. Thus, the creation of agents that reliably identify symptoms is an iterative process where the agent is tested on as much data as possible, and the results studied by the knowledge acquisition engineer and the domain expert together in order to improve it for the next iteration. The abstraction provided by the pattern matching agents provides an abstraction where machine learning can be performed even with relatively few instances. Our main reason for choosing case-based reasoning during this stage, is that DrillEdge is a decision support system used by domain experts, and we need to be able not only to provide an answer, but also explanatory support for that answer. In DrillEdge we have even chosen not to make the prediction of the system explicit. DrillEdge simply displays all the most similar cases (if any is sufficiently similar), and sorts the cases by the prediction they imply. This allows us to present DrillEdge as an automated search tool that will bring relevant cases to the attention of the user whenever a situation develops, rather than a classical style expert system that provides a prediction or a diagnosis. This has helped in presenting DrillEdge as a tool rather than a threat to the expertise of the user.

Symptom Recognition

Several methods of symptom recognition are being and have been tested. So far we have settled on two stable and reliable methods: *Rudimentary Mathematical Modeling* and *Deviation from Expected Trend*.

Rudimentary mathematical modeling is able to describe the following type of symptoms: Formation Hardness, Soft Formation and Hard Stringers. Advanced models provide good understanding of the underlying physical processes, but they can be computationally very expensive and highly dependent on data quality, and, on the assumptions. Therefore, in many occasions, it is better to use simple models by just focusing on selected effects. A simple or a rudimentary model, as opposed to an advanced model, is characterized by selecting only the most obvious influencing parameters and by ignoring unimportant effects. Such models do not explain or model every aspect of the process, but are limited to those parameters that are important for the specific deviation from the normal background level. A simple model of hardness is exemplified below:

$$ROP = C_1 \cdot DR \cdot WOB^{C_2} \cdot RPM^{C_3} \quad (1)$$

Here, ROP is Rate of Penetration, DR is Drilling Resistance, WOB is Weight on Bit, and RPM Rotations per Minute. Testing this equation for roller-cone bits has shown that best results were found when C2 and C3 were equal to 1.5 and 1 respectively. C1 is taking into account all effects not explicitly included in the Equation above, like bit type, bit characteristics, hydraulics, pore pressure etc. These effects are assumed constant after drilling has started.

$$FH = C_1 \cdot DR = ROP / (WOB^{1.5} \cdot RPM) \quad (2)$$

If formation hardness, FH, is above a certain threshold value recorded over a drilled distance of less than 2 meters, then Hard Stringers and Soft Formation are triggered and marked in the real-time drilling data.

We have selected the problem of interpreting repeatedly mechanical resistance during the tripping operation (removing the drill pipe from the hole to replace a dull drill bit) to introduce Deviation from Expected Trend, illustrated by the two symptoms Overpull and Took Weight. Whenever the resistance of moving the drill string axially becomes too large (more than 15-20 tons), it is necessary to stop and clean (ream) the hole in accordance with Best Practice. Neglecting to do so may turn the situation into worse failures like stuck pipe or lost circulation. To be able to detect the event Overpull when tripping out on the basis of this method we need first to establish a normal HKL (hook load) value. Overpull is seen as a deviation above a certain threshold value. When tripping-in the corresponding deviation will be recognized as Took Weight.

Case-Based Reasoning

While the heuristic mathematical models identify the symptoms observed while drilling, the CBR engine compares the current drilling situation with past drilling situations stored in the case base in order to find out whether the current situation develops into a problematic one. Thus the CBR engine is diagnosing the current situation and this encompasses finding out which type of problem is being observed. The past drilling situations stored in the case base are selected because they represent problems experienced in the past that the operator wants to avoid. However, the role of a case is not only to classify the current situation, but also to provide experience and best practices for how to handle the situation. Also, the outcome of the past situation is known, and it indicates what might happen in the current situation if the problem develops and corrective action is not taken.

Symptoms of drilling problems typically happen around the drill bit, and the drill bit can be placed anywhere in the well. As the drill bit is not at the bottom of the well at all times, but often is pulled up from the bottom when cleaning the hole or completely out of the hole when a change of tools is required, the position where a symptom is observed is essential. Some sections of the well (i.e. vertical parts) might be particularly problematic, and in such sections the symptoms typically cluster together. Also, the exact time a

symptom has been observed is important, as it is essential for the operator to know whether the rig is experiencing problems right now. So, both the location in the well and the time when a symptom was observed is important.

Some problematic situations are characterized by symptoms occurring in the same section where previous symptoms were observed, while other situations are characterized by symptoms occurring repeatedly over the last couple of hours. In some situations the operators should be warned that they are pulling through a section in which problems were experienced earlier.

However, symptoms of problems are not the only features that characterize a problematic oil well drilling situation. Some problematic situations are characterized by the design of the bottom hole assembly that is used, such as which drill bit, the number of and the placement of stabilizers and whether a mud motor is used or not. Others, again, are related to a specific formation or its composition. Also the type of drilling fluid that is used and its composition might be relevant, as some drilling fluids react with the formation, which can cause swelling of the well making it thinner. The problems experienced in vertical wells might be quite different from the ones experienced in one with a horizontal trajectory. Considering the fact that paths of wells are getting more and more complex, some wells have trajectories that curve more than ninety degrees from the vertical (going upwards rather than downwards), the trajectory of the well is important too. Hence, the context in which the symptoms occur is an important factor when differentiating between problematic situations, especially when trying to identify possible root causes.

Cases have two parts; one that is used by the computer to compare the two cases, the case description, while the other is an experience transfer from one human to another, the case solution. All the above mentioned features are parts of the case description. The problem solution part is a textual description intended for humans, and it has four subparts. A problem description part that describes the operation they were performing in the specific situation, a symptoms part that describes the symptoms they were experiencing, a response action that describes the actual actions they took in this situation to rectify the problem, and a recommended action, based on a post-analysis, that describes what should have been done. The problem description can easily contain pointers and links to best practices or incident reports that are stored in different software systems. Cases are captured and written by drilling experts that are trained in recognizing problematic situations with symptoms that can be identified by DrillEdge.

Cases are represented in tree structures and stored as XML files. The root node of the case contains two main sections, the problem description and the problem solution. Sections might contain other sections or leaf nodes. For example, the problem description contains the formation, drilling fluid, bottom hole assembly, well geometry and symptoms. The formation section contains the formation name and the formation composition (lithology), while the drilling fluid section contains properties describing the drilling fluid that is used, such as mud weight and whether

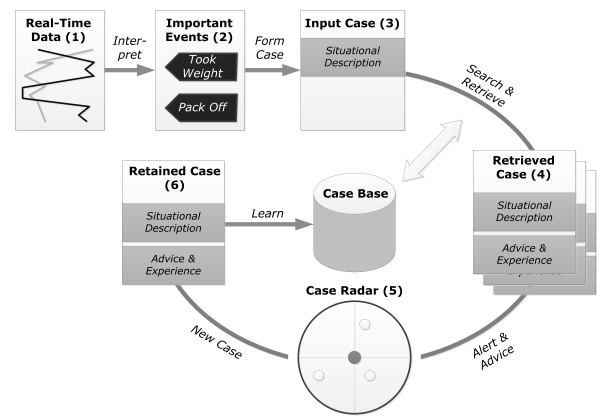


Figure 1: The DrillEdge CBR cycle.

it is oil-based or water-based. The bottom hole assembly describes the design of the bottom hole assembly. Important features are how long it is, which drill bit is used, the number of stabilizers and their positions. The section well geometry represents target depth of the current section and the depth of where the section started. The sequence section is a special kind of section that contains events, and events are representations used for symptoms. Events have different types corresponding to the type of symptoms they represent. The symptoms section contains two sequence sections. One represents the distribution of events over a given depth around the drill bit, while the other represents the events distributed over a limited time period. The end of the time sequence is the time the drill bit was located at a given depth, and the start of the time sequence represents the time the current situation started. The ranges above and below the drill bit in the depth sequence indicate the section of the well that is relevant for the situation in the opinion of the expert that built the case.

The degree of similarity between two cases is found by comparing the root nodes in the case trees. The similarities of root nodes are aggregated into section similarities, and section similarities are combined recursively until the similarity of the root nodes are found. The similarity of the root nodes is the resulting similarity of the case comparison. Root nodes can be of different types, like integers, doubles, enumerations and sequences. For each type of node a set of similarity measures can be configured for comparison. For example, all numeric features are not compared using the same type of similarity measure. Both standard similarity measures (Richter 2008) and custom made similarity measures are used to compare features.

The CBR process continuously compares the current situation with cases stored in the case base. Figure 1 illustrates the reasoning process. For each time step, the real-time data parameters are *interpreted*, and if symptoms of problems are identified, events are fired. The current situation is represented by both important events and contextual information. Events are stored in the case as depth and time sequences sorted on the distance from the drill bit and distance from

the current time respectively. The CBR system *searches for and retrieves* cases from the case base and compares them to the current case. The comparison result is sorted on similarity. All past cases with a degree of similarity above a given threshold are visualized on a GUI element, the radar, to *alert and advice* the user of past historic cases that are similar to the current situation. By investigating the similar situations the user is advised on what others did and what should have been done in similar situations in the past. A *new case* is made by the drilling engineer if the current situation is not covered by any cases stored in the case base or if other advice applies to this situation. New cases are typically quality assured through peer review by a group of domain experts. The system *learns* when new cases are added to the case base.

Related Work

Several researchers and companies have tried out case-based reasoning methods for oil drilling assistance. Very few systems have reached deployment, however, and no other system than DrillEdge, to our knowledge, links on-line data streams to past cases for real-time decision support in an ongoing drilling process. Partly based on a previous study (Shokouhi, Aamodt, and Skalle 2010), we discuss related work where CBR methods are applied to an ongoing oil drilling operation as well as applications in related drilling domains, such as well planning, reservoir engineering, and petroleum geology.

An application of CBR in planning of a drilling operation was described by the Australian research organization CSIRO (Kravis and Irrgang 2005). The system, Genesis, can use multiple cases at varying levels of generalization.

Mendes et al. (Mendes, Guilherme, and Morooka 2001) developed an application of CBR in offshore well design. A genetic algorithm (GA) was used to determine the proper trajectory of the well, starting out from a set of solutions that form the initial population. The cases retrieved via CBR constitute the initial population.

Perry et al. (Perry et al. 2004) developed a case-based system for drilling performance optimization. Project documents, well summary documents, and technical lesson documents are three levels of documents in the knowledge base hierarchy.

Popa et al. (Popa et al. 2008) applied CBR to determine the optimum cleaning technique for filter failures. A small subset of historical cases was taken from the database to evaluate the proposed solution with the actual results. According to the similarity assessment, 80% of the cases were correctly assigned to the successful cleaning method.

Bhushan and Hopkinson (Bhushan and Hopkinson 2002) developed a CBR system to search for reservoir analogs in the planning of new fields. A knowledge sharing tool, called the Smart Reservoir Prospector (SRP), was developed.

A CBR framework was developed by Schlumberger to assess the applicability of seven lift methods for land, platform, and subsea wells (Sinha, Yan, and Jalali 2003).

Abel et al. developed a CBR system to support the interpretation and classification of new rock samples (Abel,

Reategui, and Castilho 1996). To provide petrographic analyses, the system achieves its reasoning power through the set of previous cases combined with other domain knowledge. Later the system was introduced into a real corporate environment (Abel et al. 2005).

Research related to the DrillEdge system, but not part of it, has been done as joint work between Verdande Technology, Statoil, and our university groups. One line of research has been to study the effects of including an explicit model of general domain knowledge with the cases, as done in the earlier Creek system (Aamodt 2004). Shokouhi et al. utilized a newly developed version of Creek to integrate case-based and model-based reasoning (MBR) for oil drilling (Shokouhi et al. 2009). Abdollahi et al. studied the problem of uncontrolled release of formation fluids into the well through the lifecycle of a well. It was shown that pre-defined rules could successfully be integrated with CBR to obtain causes of well leakages (Abdollahi et al. 2008).

DrillEdge

DrillEdge has a client server architecture in which the server is a distributed system running on large computing clusters like the Amazon Elastic Computing Cloud (ECC) while the clients run on regular desktop machines.

Architecture

As depicted in the leftmost part of figure 2, the process view of the architecture can be illustrated as a layered architecture with four layers. The lowest level is *data acquisition* in which data are acquired from the data sources. Data sources can be both manually updated when starting to monitor an operation and updated automatically in real-time. The second lowest level is *data interpretation* where patterns are recognized and symptoms are flagged by software agents. The symptoms are fed into the *case-based reasoning* engine, which recognizes broader patterns, not focusing on a smaller set of parameters, but the complete situation. Finally, at the topmost level the data and findings are visualized for the users.

The middle part of figure 2 depicts a client-server view of the architecture. Each machine in the cluster is indicated by a gray, squared box which runs services that are drawn as white boxes with soft corners. Different kinds of services are provided by the server cluster. The main service is the operation service, and it is responsible for the three lowest levels of the processes view which are related to one real-time drilling operation. In the illustration two machines run a total of 4 operations designated with an O and a subscript for its ID. Other services are the license service, which ensures that the operator is not running more operations than it has licenses for and is designated L, and the management service, designated M, which manages the setup and configuration of operations. Clients, on the right, communicate with services through a front end server, which the only task is to ensure that the services get the right messages. All communication between the server cluster and the clients is done using the HTTP protocol.

The rightmost part of figure 2 shows the architecture of the operation service. The operation service includes a

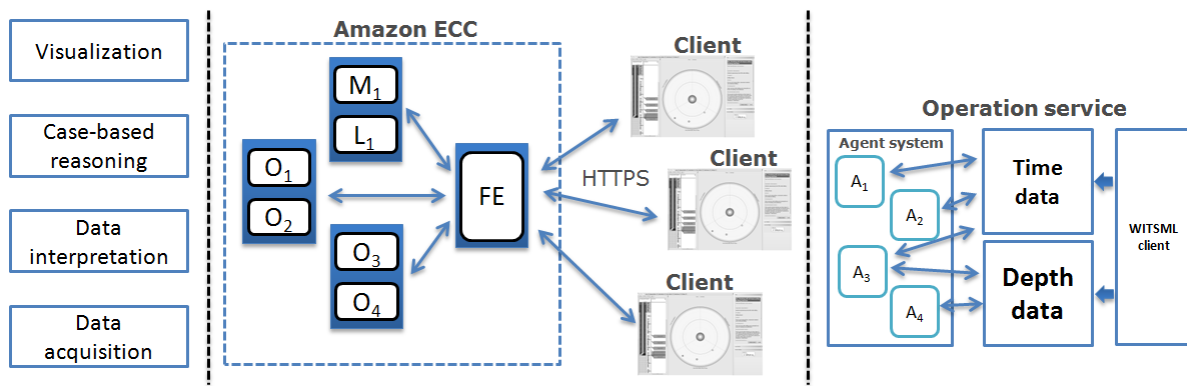


Figure 2: The DrillEdge architecture explained: Process view to the left, the client-server architecture in the middle and the operation service to the right.

WITSML (Wellsite Information Transfer Standard Markup Language) client that requests data from a WITSML server and is responsible for storing the received data in the appropriate data structure. Some data are stored in a depth table while other are stored in a time table. The agents fetch data from the data structures to perform their calculations. Some of the agents compute new values each time step, for example trend agents, while others only store data when they find new symptoms. The CBR engine is also implemented as an agent. It is however dependent on the calculations from all the other agents, so it will not execute until all other agents have done their calculations for a given time step.

Client

The intention behind the client is that it should be easy to use and self-explanatory. Users can choose between several different tabs that visualize data in different manners, like parameters plotted against time or depth, the static properties that the operation is configured with, or an overview. The two main tabs are the time tab and the overview tab. Figure 3 shows screen captures of the two tabs with the overview tab on the left and the time tab on the right. The screen shots are included in order to illustrate the structure of the screens and not their detailed contents.

The overview provides three main types of information: the depth view, the radar and the case solutions. The *depth view* visualizes the drill bit in the hole, and it shows the current depth and the formation layers and their lithology. Both rotation of the drill bit and mud flow are animated to make it easier for the users to understand the current activity. Also, events are plotted according to depth so if the drill bit is pulled through a section of the well that was problematic to drill, then this can be seen from the depth view. The most important GUI element on the overview screen is the radar, and this is emphasized by the amount of screen real estate it occupies. Past cases are plotted on *the radar* according to how similar they are to the current situation. The more similar a case is to the current situation the closer to the center of the radar the case is located; if a case is 100% similar, it will be in the center. The radar threshold is 50% so all cases

that are less similar that the threshold will not be plotted on the radar. This ensures that only relevant cases are brought to the attention of the user. The radar is divided into different sectors according to the problem area the cases in the case base are representing, such as Lost Circulation and Mechanically Stuck Pipe. When clicking on a case on the radar, the *case solution* will appear on the right describing the past situation, the problem and recommended actions.

The time tab provides a list of *available parameters* that can be dragged over to the *parameter graph* columns so that the values will be drawn as graphs with time flowing downwards. Events are drawn in the *event column* so that users can see exactly when symptoms appeared. As event names use common terminology for the users, they can easily look at the corresponding parameters to see whether they agree with the symptoms. *Case graphs* can be drawn for each case in the case base, and they show the similarity degree of the cases at every time step. Thus the case graphs convey the history of how the current situation has developed seen through the lens of the past cases stored in the case base.

Development and Cost

The development of DrillEdge was based on research performed at the Norwegian University of Science and Technology (NTNU), more specifically within the AI and Drilling Engineering research groups (Skalle, Sveen, and Aamodt 2000; Aamodt 2004). In 2006, the Norwegian Research Council's Petromaks program in conjunction with Statoil, the largest Norwegian oil company, provided funding for a two year project to develop a pilot system for studying whether case-based reasoning and related technologies could be used to detect and predict drilling problems. This project funded the two first years of commercial development, providing approximately \$3 million to fund 14 man-years of development over these two years. At the end of this period, in 2008, a prototype of DrillEdge was finished, although another 6 man-years of development were required before the first customer version was released in 2009, placing the total development cost (excluding marketing, but including all overhead) at around \$4.5 million. Since then,

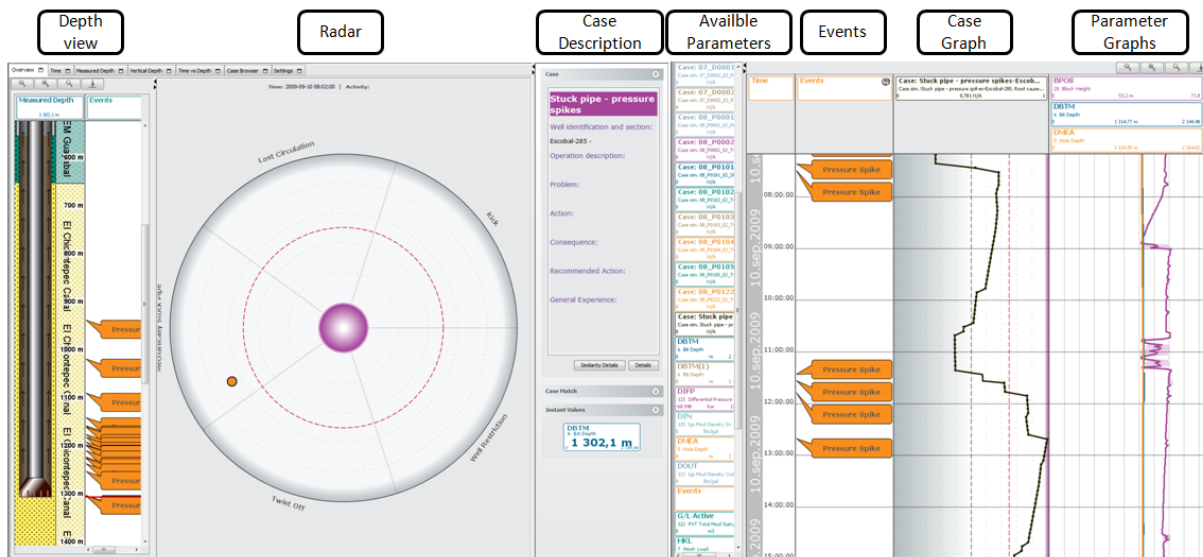


Figure 3: Two different screens from the DrillEdge client is shown above. The overview screen is on the left hand side and the time view is shown to the right.

development has continued both to broaden the types of drilling problems that can be detected, but also to develop features such as administrator tools, email alerts, support for cloud deployments and so on. Today, the DrillEdge development team consists of 12 full-time developers and testers.

Commercial Deployment

Drilling for oil and gas is very expensive, especially in off-shore locations. In the North Sea, a single 6000 meter well can cost \$100 million to drill. The industry also recognizes that there are significant savings possible here – depending on the well complexity, the average NPT can be as much as 30%. Some of this NPT is unavoidable; e.g., waiting out especially bad weather, but DrillEdge is currently able to address problems causing about half of this downtime. At the time of writing, there are 1190 offshore rigs in the world, about 70% of which is operational at any time (RIGZONE 2012). These rigs have an average day rate (cost for the operator company to rent the rig) of \$144,000 per day per rig. Provided that the NPT can be reduced by 5% across the off-shore rig fleet, this translates to a cost saving of over \$2.1B per year. In addition, there are over 3000 rotary land rigs in operation in the world, a number that is increasing rapidly due to the drilling for shale gas in the United States (there are about 2000 rotary land rigs in the US alone, an increase of about 300 over the last year) (Hughes 2012). These rigs have lower NPT and day rates than offshore rigs, but are interested in real-time drilling data to improve efficiency and consistency of drilling.

DrillEdge is in use by Shell in their Real-Time Operating Centers in the US, where it has been used on many of the most challenging wells drilled by Shell across the world in the last six months. It is also installed as a core part of the real-time monitoring infrastructure in the national

oil company of Oman, PDO. At the time of this writing, DrillEdge monitors over 20 rigs in Oman alone. In September 2011, Verdande Technology signed a partnership with Baker Hughes, a major oil service company, which means they will use DrillEdge as a key component in their real-time monitoring services as sold to operators, starting in 2012. In addition, Verdande Technology also executed successful pilot installations for more than 15 oil operators during 2011.

Case Studies

DrillEdge has been used successfully for analyzing data for several customers both in pilot tests and commercial deployment. Two of the case studies that have been made are summarized below. In each of them DrillEdge analyzed several sets of historical data. The tests were performed blindly; Verdande Technology was not informed beforehand of which or when the operator had experienced problems. Nor was any information about the outcome given.

Case Study: Stuck Pipe

A major service company wanted to minimize the risk of stuck pipe events for their clients in Latin America and around the world. DrillEdge was employed in a post-well historical analysis to determine if the losses and ballooning events could have been recognized in advance.

A project was launched to evaluate the potential of DrillEdge technology to recognize the stuck pipe problems. Time and depth-based data from several prior land drilling operations in Latin America were employed in a rigorous testing routine. Overpull events, erratic torque and pack offs were thought to be key contributors to the stuck pipe scenarios. Cases were built in DrillEdge and these cases were associated with other data, such as BHA, trajectory, mud properties and the formation. Well drilling data

was then streamed while DrillEdge technology monitored it for precursor events. Tests were highly successful with the DrillEdge technology consistently predicting stuck pipe incidents 6 to 8 hours prior to the events occurrence. While drilling, this window was deemed sufficient to proactively address a stuck pipe risk. The success of this project set the stage for further testing in live field trials from RTOCs.

Case Study: Lost Circulation

A major operator encountered costly NPT due to mud losses and subsequent hole ballooning. DrillEdge was employed in a post-well historical analysis to determine if the losses and ballooning events could have been recognized in advance.

Using a combination of information including pit volumes, well and drill string geometry, ROP and lag times, the DrillEdge system was able to recognize changes in volume that were not accounted for by the drilling process. In a historical analysis of a well with known problems related to mud losses and hole ballooning, DrillEdge first saw indications of losses three days before the customer experienced total losses. During this time, a series of loss events was detected, combined with hole ballooning at connections. Had DrillEdge been employed on this well, these risk associated with these problems could have potentially been better assessed, leading to remedial actions based on company best practices that could have reduced costly NPT.

Conclusion

In this paper, we have presented DrillEdge, a software system for supporting decisions in real-time when performing oil-well drilling operations. DrillEdge has been developed since late 2006 and has been commercially deployed for over two years on more than 200 wells. We show how AI methods such as case-based reasoning, pattern matching and agent systems have helped large oil-well drilling operators to prevent doing the same mistakes over and over again.

References

Aamodt, A., and Plaza, P. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1):39–59.

Aamodt, A. 2004. Knowledge-Intensive Case-Based Reasoning in Creek. In Funk, P., and González-Calero, P., eds., *ECCBR*, volume 3155 of *Lecture Notes in Computer Science*, 1–15. Springer.

Abdollahi, J.; Carlsen, I.M.; Randhol, P.; Tenold, E.; Haga, H.; and Jakobsen, T. 2008. A Case-Based Approach to Understand the Complexity of Causal Connections Related to Well Integrity Problems. In *SPE 111129-MS, IADC/SPE Drilling Conference, Orlando*. SPE.

Abel, M.; Silvaa, L.; Campbell, J.; and De Rosc, L. 2005. Knowledge acquisition and interpretation problem-solving methods for visual expertise: Study of petroleum-reservoir evaluation. *Journal of Petroleum Science and Engineering* 47(1-2):51–69.

Abel, M.; Reategui, E. B.; and Castilho, J. M. 1996. Using Case-Based Reasoning In A System That Supports

Petrographic Analysis. In *Artificial Intelligence in the Petroleum Industry: Symbolic and Computational Applications II, chapter 7*, 159–172. Technip.

Bhushan, V., and Hopkinson, S. 2002. A Novel Approach to Identify Reservoir Analogues. In *European Petroleum Conference, Aberdeen*.

Booth, J. 2011. Real-Time Drilling Operations Centers: A History of Functionality and Organizational Purpose—The Second Generation. *SPE Drill & Compl* 26(2):295–302.

Crichton, M. T.; Lauche, K.; and Flin, R. 2005. Incident Command Skills in the Management of an Oil Industry Drilling Incident: A Case Study. *Journal of Contingencies and Crisis Management, Vol. 13, No. 3*, pp. 116-128, September 2005.

Endsley, M. R. 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37:32–64(33).

Hughes, B. 2012. "Baker Hughes Investor Relations: Overview and FAQ".

Kravis, S., and Irrgang, R. 2005. A Case Based System for Oil and Gas Well Design with Risk Assessment. *Applied Intelligence* 23(1):39–53.

Mendes, J.; Guilherme, I.; and Morooka, C. 2001. Case-based system: indexing and retrieval with fuzzy hypercube. In *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*. IFIP.

Perry, P.; Curry, D.; Kerridge, J.; Lawton, J.; Bowden, D.; and Flett, A. 2004. A Case Based Knowledge Repository for Drilling Optimization. In *SPE 87994-MS, IADC/SPE Asia Pacific Drilling Technology Conference and Exhibition*. SPE.

Popa, A.; Popa, C.; Malamma, M.; and Hicks, J. 2008. Case-Based Reasoning Approach for Well Failure Diagnostics and Planning. In *SPE 114229-MS, SPE Western Regional and Pacific Section AAPG Joint Meeting*. SPE.

Richter, M. 2008. Similarity. In Perner, P., ed., *Case-Based Reasoning on Images and Signals*, volume 73 of *Studies in Computational Intelligence*. Springer. 25–90.

RIGZONE. 2012. Offshore rig day rates.

Shokouhi, S.; Aamodt, A.; and Skalle, P. 2010. Applications of CBR in oil well drilling. In *Proceedings of IIP2010 - 6th International Conference on Intelligent Information Processing*, 102–111. IFIP.

Shokouhi, S.; Aamodt, A.; Skalle, P.; and Sørmo, F. 2009. Comparing Two Types of Knowledge-Intensive CBR for Optimized Oil Well Drilling. In *Proceedings of the 4th Indian International Conference on Artificial Intelligence (IICAI-09)*, 722–737.

Sinha, S.; Yan, M.; and Jalali, Y. 2003. Completion Architecture: A Methodology for Integrated Well Planning. In *SPE 85315-MS, SPE/IADC Middle East Drilling Technology Conference and Exhibition*. SPE.

Skalle, P.; Sveen, J.; and Aamodt, A. 2000. Improved Efficiency of Oil Well Drilling through Case-Based Reasoning. In *PRICAI*, 712–722.