

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

Odd Erik Gundersen, Frode Sørmo, Agnar Aamodt, Pål Skalle

■ *In this paper we present DrillEdge — a commercial and award-winning software system that monitors oil well drilling operations in order to reduce nonproductive 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. We document the methods utilized, the architecture, the GUI, and the development cost in addition to two case studies.*

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 (PDO), 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 (2011) has documented the advantages and history of RTOCs.

Monitoring a situation is to become aware of it. This is the main concern for decision makers. Endsley (1995) investigated situation awareness in depth and identified three levels of situation awareness. 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. These three levels apply to oil well drilling operations. 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, that is, 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) (Kolodner 1992) is a computational method (Watson 1999) that is based on this principle. One of its roots stems from Roger C. Schank's work on representation of meaning for generation of natural language (Schank 1980). While working with natural language processing, Schank started considering inference as being the most important part and proposed scripts as a method to connect events together by the remembrance that they have been connected before, which again led to the study of dynamic memory (Schank 1983) for situation understanding, problem solving, and learning.

The leap from memory structures and retrieval algorithms that allow understanding to happen to the use of cases in other sorts of reasoning is a short one. [...] Further observations showed that, indeed, cases were useful in problem solving (Kolodner [1993], page 102).

In CBR, new problems are solved by reusing the solutions of the most similar past problems stored in a case base. Furthermore, newly solved problems are stored in the case base and thus incremental and sustained learning is supported intrinsically by the reasoning method. The reasoning process can be viewed as a cyclic, four-step process often referred to as the CBR cycle (Aamodt and Plaza 1994).

First, the current problem is compared to the past cases in the case base, and the most similar cases are retrieved. Second, the solutions of the most similar cases are reused to solve the current problem. Third, the suggested solution is revised

and possibly updated, and finally the current problem with its revised solution is retained in the case base. A case in DrillEdge is a concrete drilling situation that comprises a collection of symptoms that previously lead to a problem and a recommendation 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.

DrillEdge is not alone in being a commercially deployed CBR system, as CBR systems have a long tradition in this regard (Watson 1997, Cheetham and Watson 2005). General Electric has deployed several CBR systems internally, and one of them is a software tool for determining color formulas that match requested colors (Cheetham 2004). CBR even helped users with specially compiled TV guides that suit their particular viewing preferences (Cotter and Smyth 2000).

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. After a section on related work, the DrillEdge system architecture is described. Information about the system development process and costs are provided, and a description of the system's deployment status is given. Finally, we present two case studies from actual deployment.

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. Problems that might occur that can be detected by remote observation of drilling parameters are when the drilling pipe gets stuck in the ground because of gravel pack around it, the drill string twists, off and drilling mud is lost in the formation. There are several problems related to the task of monitoring the drilling situation remotely.

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.

To comprehend the current situation, all relevant information must 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 to combine all the information or to find the right clue even when the data is available (Booth 2011). Some of the information is provided in real time, such as some physical models that are updated accordingly, while other information needs to be computed manually. 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 theoretical knowledge but also experience. One of the reasons for moving expert knowledge from the rigs to RTOCs was to colocate 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 are the age gap and the connection between personnel demands and the oil price. Because the oil price governs the personnel demands, a significant number 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 and 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. As a result, 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 the pressure down in the hole with the drill bit. These physical models are essential tools in planning a well, and much research in the industry has been directed toward 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 redrilled, 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 to 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 to 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 whether 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 on 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 are 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 to provide not only 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 formation 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 is rotations per minute. Testing this equation for roller-cone bits has shown that best results were found when C_2 and C_3 were equal to 1.5 and 1 respectively. C_1 is taking into account all effects not explicitly included in the previous equation, like bit type, bit characteristics, hydraulics, pore pressure, and so on. 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 to introduce deviation from expected trend, illustrated by the two symptoms overpull and tool weight. Tripping out is to remove the drill pipe from the hole to replace a dull drill bit, for example, while tripping in is to run the drill bit back to the bottom of the hole. Overpull is when the weight on the hook carrying the whole drill string suddenly gets higher when tripping out, and a tool weight is when the weight of the whole drill string suddenly gets lower when tripping in. 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.

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 (that is, 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 are 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 com-

plex, some wells have trajectories that curve more than 90 degrees from the vertical (going upward rather than downward), 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: The case description and the case solution. The case description is used by the computer to compare the two cases while the case solution is an experience transfer from one human to another. 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 postanalysis, 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 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

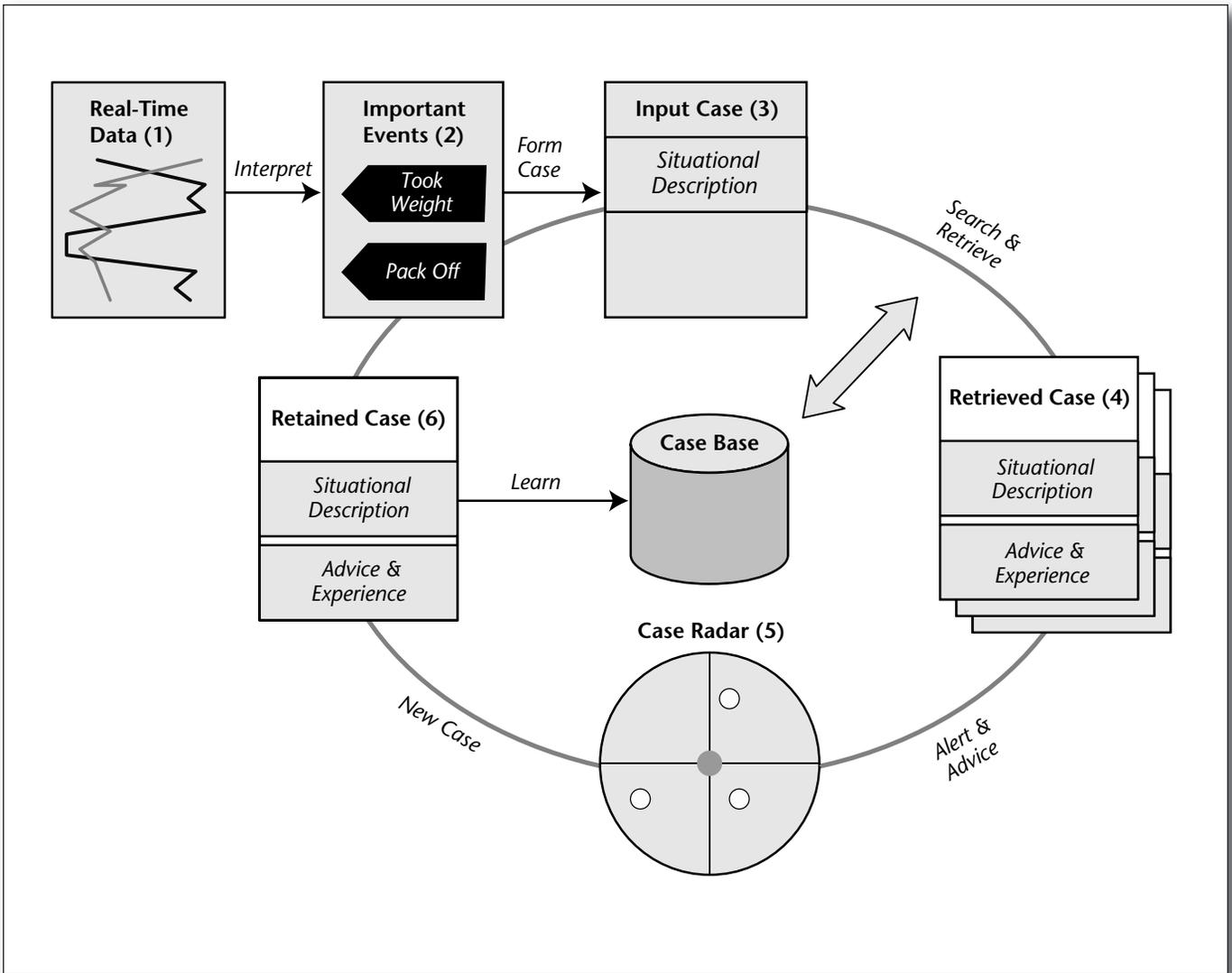


Figure 1. The DrillEdge CBR Cycle.

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 is 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, not all numeric features are compared using the same type of similarity measure. Both standard

similarity measures (Richter 2008) and custom-made similarity measures, which are domain specific, 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 advise 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 created 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 online 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, 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 through CBR constitute the initial population. 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. (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 percent of the cases were correctly assigned to the successful cleaning method. 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, Silvaa, Campbell, and De Rosc 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. (2009) utilized a newly developed version of Creek to integrate case-based and model-based reasoning (MBR) for oil drilling. 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 predefined 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 that runs services, which 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 four operations desig-

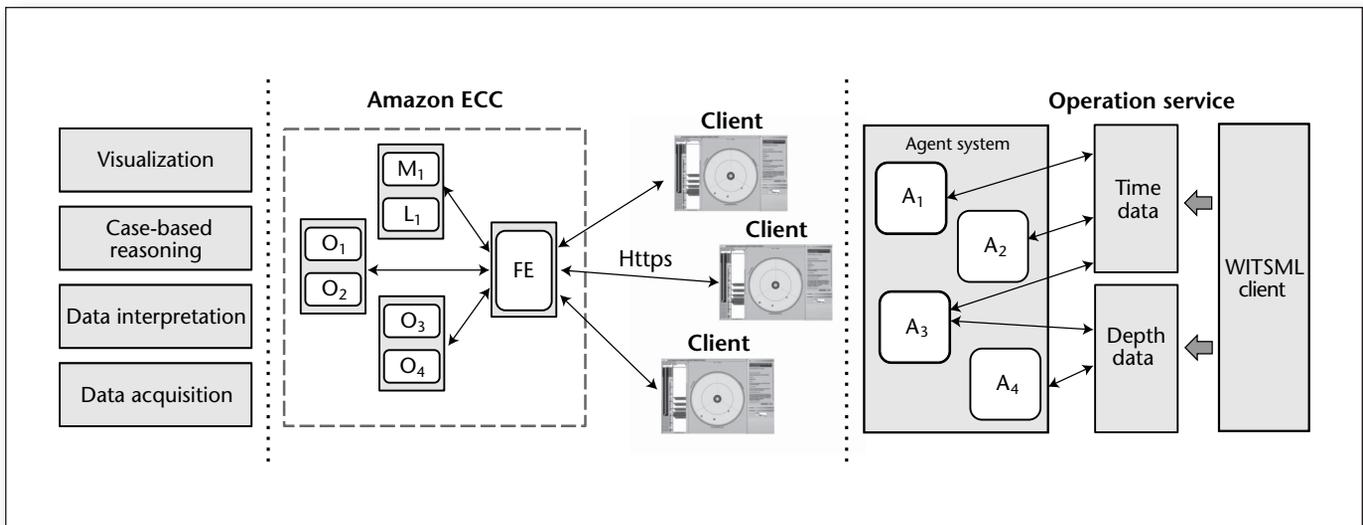


Figure 2. The DrillEdge Architecture Explained

Process view to the left, the client-server architecture in the middle, and the operation service to the right.

nated 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, whose only task is to ensure that the services get the right messages. All communication between the server cluster and the clients is done using HTTP.

The rightmost part of figure 2 shows the architecture of the operation service. The operation service includes a 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 others 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 store data only 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, 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 percent similar, it will be in the center. All cases that are less similar than a threshold will not be plotted on the radar. A threshold of 50 percent has shown to ensure 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

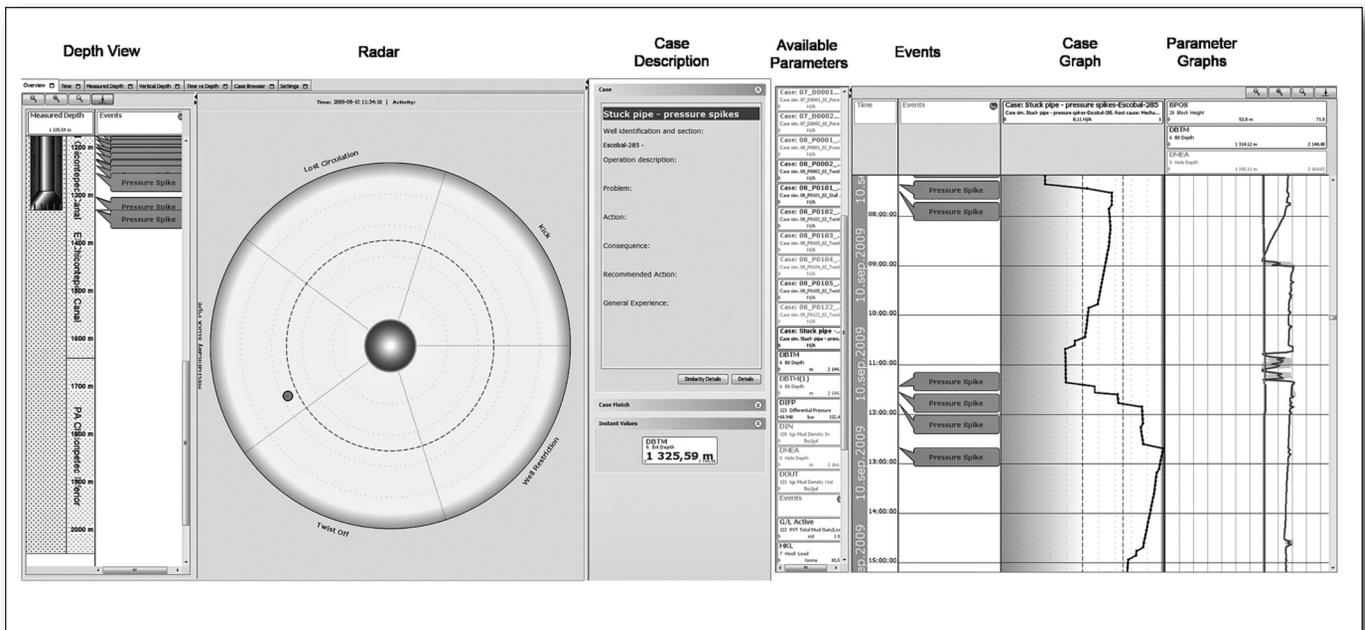


Figure 3. Two Different Screens from the DrillEdge Client.

The overview screen is on the left side and the time view is shown to the right.

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, 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 offshore 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 nonproductive time (NPT) can be as much as 30 percent. Some of this NPT is unavoidable, for example, 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 percent of which are operational at any time.¹ These rigs have an average day rate (cost for the operator company to rent the rig) of \$144,000 per rig. Provided that the NPT can be reduced by 5 percent across the offshore rig fleet, this translates to a cost saving of over \$2.1 billion per year. In addition, there are more than 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 United States alone, an increase of about 300 over the last year) (Baker Hughes, inc. 2012). These rigs have lower NPT and day rates than offshore rigs, but we are interested in real-time drilling data to improve efficiency and consistency of drilling.

DrillEdge is in use by Shell in its Real-Time Operating Centers in the United States, 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 more than 20 rigs in Oman alone. In September 2011, Verdande Technology signed a partnership with Baker Hughes, a major oil service company, which means Baker Hughes will use DrillEdge as a key component in its 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 postwell 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 their 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 postwell historical analysis to determine whether 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, the 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.

Challenges

We face mainly three challenges in our work. These are revising the symptom recognition agents, the time used to find and capture a proper case, and the real-time demands on the similarity comparison.

The symptom-recognition agents recognize pattern signatures for the different symptoms by analyzing the change of several parameters in the WITSML data streams. Several factors affect the symptom signatures, such as the depth, the hole diameter, and the tools used. Hence, when entering new markets the conditions can change significantly, and thus the symptom-recognition agents need to be revised. Our goal is to avoid regional customization of agents, which makes the agent development considerably more difficult.

Building good cases that capture significant situations and provide proper advices require more work than first anticipated. Finding the data containing problematic situations and quality assurance of the data in regard to data frequency and missing parameters are very time consuming, as companies have not really used the stored data before. After finding and assuring the quality of the data, experts analyze it and assure that it contains symptoms supporting the claimed problems. Then, the symptom analysis is performed before experts can start identifying where the cases should be captured. Captured cases need to be quality assured to check how early and whether they actually would create relevant alarms for the problems. Building one case requires around 40 man-hours on average by a drilling expert trained in identifying proper situations.

Providing decision support in real time puts constraints on the case comparison. We have found that sequence matching is the bottleneck, and we are currently studying methods for speeding up the sequence comparison (Gundersen 2012).

Conclusion and Future Work

In this article, 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 more than 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.

Verdande Technology is currently expanding into new industries and offers solutions in finance and health care in addition to oil well drilling. In contrast to the oil well drilling domain, in which five minutes reaction time is sufficient, financial services are very focused on low latency, and this requires further improvement of the case-based reasoning engine.

Note

1. See the RIGZONE 2012 offshore rig day rates (www.rigzone.com).

References

- Aamodt, A. 2004. Knowledge-Intensive Case-Based Reasoning in Creek. In *Advances in Case-Based Reasoning: The 7th European Conference*, volume 3155, Lecture Notes in Computer Science, ed. P. Funk and P. González-Calero, 1–15. Berlin: Springer.
- Aamodt, A., and Plaza, P. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1): 39–59.
- 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. SPE 111129-MS. Paper presented at the IADC/SPE Drilling Conference, Orlando, FL, 4–6 March.
- 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. Paris: Technip.
- Baker Hughes, Inc. 2012. Baker Hughes Investor Relations: Overview and FAQ. Houston, TX: Baker Hughes Incorporated.
- Bhushan, V., and Hopkinson, S. 2002. A Novel Approach to Identify Reservoir Analogues. Paper presented at the European Petroleum Conference, Aberdeen, Scotland, UK, 29–31 October.
- Booth, J. 2011. Real-Time Drilling Operations Centers: A History of Functionality and Organizational Purpose — The Second Generation. *SPE Drilling & Completion* 26(2): 295–302.
- Cheetham, W. 2004. Tenth Anniversary of the Plastics Color Formulation Tool. In *Proceedings of the 16th Conference on Innovative Applications of Artificial Intelligence*, 770–776. Menlo Park, CA: AAAI Press.
- Cheetham, W., and Watson, I. D. 2005. Fielded Applications of Case-Based Reasoning. *Knowledge Engineering Review* 20(3): 321–323.
- Cotter, P., and Smyth, B. 2000. PTV: Intelligent Personalised TV Guides. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 957–964. Menlo Park, CA: AAAI Press.
- 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* 13(3): 116–128.
- 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).
- Gundersen, O. E. 2012. Toward Measuring the Similarity of Complex Event Sequences in Real-Time. Paper presented at the 20th International Conference on Case-Based Reasoning, Lyon, France, 3–6 September.
- Kolodner, J. L. 1992. An Introduction to Case-Based Reasoning. *Artificial Intelligence Review* 6(1): 3–34.
- Kolodner, J. L. 1993. *Case-Based Reasoning*. San Francisco: Morgan Kaufmann Publishers.
- 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 *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.
- Perry, P.; Curry, D.; Kerridge, J.; Lawton, J.; Bowden, D.;



IAAI-13

Bellevue, Washington USA
July 16–18 2013

and Flett, A. 2004. A Case Based Knowledge Repository for Drilling Optimization. Paper presented at the Ninth Biennial International Association of Drilling Contractors (IADC) and the Society of Petroleum Engineers (SPE) Asia Pacific Drilling Technology Conference, SPE 87994-MS. Tianjin, China, 9–11 July.

Popa, A.; Popa, C.; Malamma, M.; and Hicks, J. 2008. Case-Based Reasoning Approach for Well Failure Diagnostics and Planning. Paper presented at the SPE Western Regional and Pacific Section AAPG Joint Meeting SPE 114229-MS, Edmonton, AB, Canada, 29 March–2 April.

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

Schank, R. C. 1983. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. New York: Cambridge University Press.

Schank, R. C. 1980. Language and Memory. *Cognitive Science* 4(3): 243 – 284.

Schank, R. C. 1983. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. New York: Cambridge University Press.

Shokouhi, S.; Aamodt, A.; and Skalle, P. 2010. Applications of CBR in Oil Well Drilling. In *Intelligent Information*

Processing V, Proceedings of the 6th International Conference on Intelligent Information Processing, 102–111. Berlin: Springer.

Shokouhi, S.; Aamodt, A.; Skalle, P.; and Sørmo, F. 2009. Comparing Two Types of Knowledge-Intensive CBR for Optimized Oil Well Drilling. Paper presented at the 4th Indian International Conference on Artificial Intelligence, Tumkur, India, 16–18 December.

Sinha, S.; Yan, M.; and Jalali, Y. 2003. Completion Architecture: A Methodology for Integrated Well Planning. Paper presented at the SPE/IADC Middle East Drilling Technology Conference and Exhibition, SPE 85315-MS. Abu Dhabi, United Arab Emirates, 20–22 October.

Skalle, P.; Sveen, J.; and Aamodt, A. 2000. Improved Efficiency of Oil Well Drilling Through Case-Based Reasoning. In *Topics in Artificial Intelligence, 6th Pacific Rim International Conference on Artificial Intelligence*, Lecture Notes in Computer Science 1886, 712–722. Berlin: Springer.

Watson, I. 1999. Case-Based Reasoning Is a Methodology Not a Technology. *Knowledge-Based Systems* 12(56): 303–308.

Watson, I. D. 1997. *Applying Case-Based Reasoning — Techniques for the Enterprise Systems*. San Francisco: Morgan Kaufmann.

Odd Erik Gundersen is the research and development director at Verdande Technology and a research scientist at the Norwegian University of Science and Technology (NTNU). He has published papers on case-based reasoning, intelligent agents, and physically based animation of fire.

Frode Sørmo is the chief technology officer at and a cofounder of Verdande Technology. He received his Ph.D. in 2007 in computer science and artificial intelligence from NTNU.

Agnar Aamodt is a professor of computer science and artificial intelligence at the Norwegian University of Science and Technology and a cofounder of Verdande Technology. He received his Ph.D. in 1991 in computer science and artificial intelligence from the Norwegian University of Science and Technology. His main research interest is in knowledge-based decision support, combing case-based and other reasoning methods. He has spent longer periods in the AI labs at the University of Texas, Austin (1987–88), Free University of Brussels, VUB (1991–92), and at the Institut d'Investigació en Intel·ligència Artificial (IIIA), Barcelona (2001–02). He is currently a visiting professor at the University of Auckland.

Pål Skalle is an associate professor at the Norwegian University of Science and Technology and a cofounder of Verdande Technology. He received his Ph.D. in 1983 in drilling engineering from NTNU. He has served both as technical and associate editor for the journal *SPED&C*, and was in 2005 honored by SPE as an outstanding technical editor and in 2008 as a peer apart. Skalle's present research activities and fields of interest cover topics like knowledge engineering, drilling fluid engineering, evaluation of real-time drilling data, and pressure control during drilling.