SPE 112131

Use of Agent Structures for Event Detection: Identification of Wells Watering Out on Troll

Randi-Helene Halmøy, SPE, and Frédéric Verhelst, SPE, Epsis AS; Marta Dueñas Díez, StatoilHydro; Martin Halvorsen, StatoilHydro; and Jan-Erik Nordtvedt, SPE, Epsis AS

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

Abstract

The industry-wide focus on *Intelligent Energy* in recent years has lead to more and better data being available, both onshore and offshore. Regrettably, processing tools have not seen a proportional improvement, and therefore engineers experience a data overload. Fairly simple alarms along with manual routine checks using independent, "offline" tools are dominating the daily tasks in production optimisation, and more intelligence and flexibility is thus needed. In this paper we suggest an agent structure allowing online, real-time event detection.

In this work producing oil wells watering out on Troll were used as an example. The detection algorithm was designed to search for significant drops in the wellhead pressure. By detection of such trends at an early stage, a successful correction and restabilisation is more likely.

Introduction

In recent years development in computer technology has lead to a considerable increase in availability, quality and quantity of data. More and more data is made available both onshore and offshore, and the acquisition frequency and storage possibilities have been improved considerably. This development has offered a new range of possibilities for improving operations, which is widely known as IO (*Integrated Operations), e-operation* or *i-field.* However, the handling capacity in the organisations has not been able to follow the vast improvement in data availability. A variety of analysis and visualisation programs exist, but data retrieval, preparation and configuration may be time-consuming and complex. This leads to unreleased potential, especially related to *real-time* and *online* applications. The amounts of data available and the possibilities this offers largely "overpowers" the capacities of the current tools and methods used, thus leaving what can be called a *capacity gap* between data availability and data handling.

To keep up with the amounts of data, *data-driven workflows* can be a useful tool, with a central element being automatic event-detection. By *data-driven workflows* we understand workflows where tasks, analyses and adjustments to be performed are triggered by events detected in the real-time data stream, as opposed to being part of predefined, scheduled routines. In certain contexts and areas this is partly done today, but many alarms and detections used today are not "smart" enough, resulting in both false detections and missed events.

Although the equipment in the industry is becoming more automated, with less manual operations related to daily operation, there will still be tasks that may be too risky, too complex, or simply too expensive to automate, either due to the nature of the task or the state/properties of available equipment. The solution(s) presented here are mainly intended as a tool to help filling this gap, by assisting production/process monitoring and by highlighting irregularities. Nevertheless, a natural expansion of the system would involve more complex algorithms simulating more "intelligence", and could also include automation of tasks based on detections.

For the type of monitoring envisioned here, we have investigated the use of *agents.* An agent-based software solution is designed as an automated system composed of multiple agents. These would be small programs integrated with the existing

system, meaning they would mainly run in the background and thus be invisible to the users. Each agent would perform specific and limited tasks, where the main gain from the system appears when the information gathered by different agents is compared and seen in context with each other. By dividing the analysis into several parts in this way, the information gathered by one agent can easily be used in different contexts and for various purposes.

Agents

Smart agents or *Intelligent agents* are probably most known for their use on the internet for searching and tracking. However, the application of *agents* spans over much more than this. As examples, we could mention that agents and automated intelligence are part of extensive NASA research projects [1], and in 2004 the first fully agent-controlled unmanned aerial vehicle (UAV)-flight took place in Australia [2]. A study showing their applicability for the oil industry have been undertaken by E.Landre et al. [3]

There is no unified definition of *an "agent*", as different groups within the agent research and development community prefer different definitions of an *agent.* Two examples follow:

Definition by Russel & Norvig [4]:
"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors"

Definition by Franklin and Graesser [5]:
*"An **autonomous agent** is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future"*

Of these two definitions, the *Russel and Norvig*-definition is more open and spans over a wider area of applications than the one by *Franklin and Graesser*. The latter definition can be extended by characterizing the agents by certain properties: *1.Reactive, 2.Autonomous, 3.Goal-oriented, 4.Continuous over time, 5.Communicating, 6.Learning, 7.Mobile, 8.Flexible,* and 9.*Character*. Based on the definition above, all agents should possess properties 1-4. In addition, agents may be divided into subgroups depending on the additional properties they possess.

The grouping of agents into subgroups based on their properties has also been suggested by other authors, but with slightly different groups. One of these is Nwana [6], who suggests that cooperation, learning and autonomy are the three basic properties, and the combination of these defines the type of agent. For example, a smart agent would need to possess all three properties, while collaboration agents would need only cooperation and autonomy.

The Franklin and Graesser-definition corresponds fairly well to the use of the term in the context of this work. The aim of our agents is to monitor and cross check trends to alert a user of parameters needing attention. In this way, the agent will indirectly affect what it senses in the future. We may also want to make the agents more intelligent by introducing a certain degree of learning capabilities at a later stage, but this is not included at the present stage.
The agents in our system would therefore initially possess the properties 1- 4, 5 and 8 in the list above, thus making them *flexible communicating agents.*

Different analysis agents will study different aspects of the system. The specific analysis algorithms could be collected from a library, so that the same algorithm can be used by different agents, and one agent could also use different algorithms in different situations. The agents could be organised in a small hierarchy, where a superior agent makes the decisions on whether or not an alarm should be raised.

Agents of the same level should also be able to communicate with each other, which makes them able to request information from other agents and possibly trigger actions from each other directly when needed. In a large system, a separate prioritizing agent could be designed, where the criticality of the situations may be assessed, and the agent keeps an operator updated with a priority list, most critical tasks first. This list should be dynamic, with continuously updated content and priority order if necessary.

Application and example

a) The Troll Oil Field

In this paper, we will apply agents to monitor the well status on the Troll Oil Field. The physical process that is assessed in doing so is discussed later in this section.

The Troll Field is located in the North Sea about 80 km off the coast at 315 – 340 m water depth. Troll is a giant gas field [7] with a thin oil layer. The commercial success of the Troll Oil Field was secured through actively utilizing horizontal drilling and geosteering [8]. Troll has about four billion barrels of oil in place, and consists of two main structures, Troll East and West; most of the oil is located in thin oil zones (4 – 27 m) in the Troll West oil province. The Troll Oil Field is one of the largest subsea developments in the world. The oil from Troll is produced through close to 120 horizontal wells that are producing into 27 clusters with more than 200 km flow lines at the seabed. Figure 1 and Figure 2 show the Troll B platform and the corresponding subsea layout.

Oil production from Troll is highly dynamic with rate and time dependent GOR [9]. To optimize production from Troll, several monitoring and control issues are addressed on a daily basis; see Hauge and Horn [9]. In this paper we address well monitoring utilizing an agent structure to enhance the overall Troll monitoring capabilities.

Further details on Troll and oil production from the Troll field may be found in Mikkelsen et al. [8] and in Hauge and Horn [9].

Figure 1: The concrete floating production platform, Troll B.

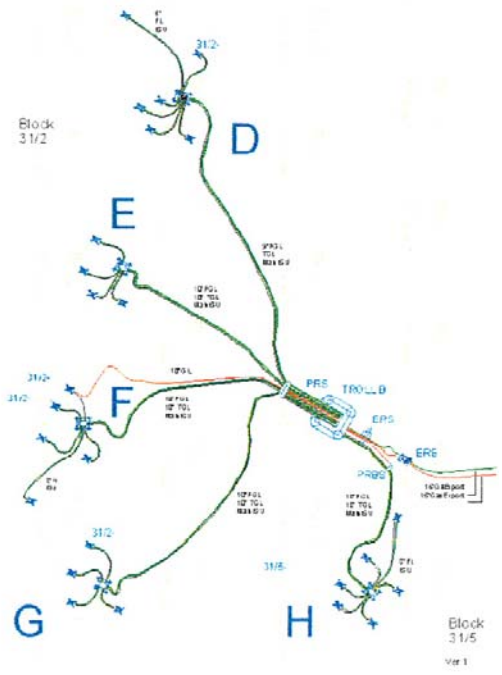

Figure 2: Troll B subsea layout [9]

b) Wells watering out – conceptual description of example problem

In the following description [10] [11] a two-phase system with oil and water is assumed for simplicity and clarity. The same principle would apply to a system with gas present, but the force balance would be slightly different, and the pressure drop would most probably be considerably accelerated compared to a pure water-oil-case.

To allow transportation of fluids in the well, a pressure difference ΔP between the wellhead pressure (P_s) and the pressure at the bottom of the well (P_{wf}) (Figure 3) is necessary. A certain pressure difference will have a *lifting capacity* in terms of fluids that can be transported, in our example oil containing water. The density difference between oil and water results in the water sinking towards the bottom, while a drag force due to friction and dependent on production rate will counteract this trend.

Water accumulation may lead to both a heavier column and a lower pressure difference available for lifting the oil, resulting in a decrease in production rate, reduced lift capacity and further water accumulation. This will eventually lead to a full stop in

production, i.e. the well is “watered out”. This effect can be observed as an increasing drop in the upstream wellhead pressure. As illustrated in Figure 4, the pressure drop starts slowly, but the gradient increases and the drop rate is accelerated with time, leaving the last bit of the drop to happen very fast. If the drop is noticed and action taken early enough, only small adjustments to the choke opening would be needed to correct the pressure and restabilize the flow.

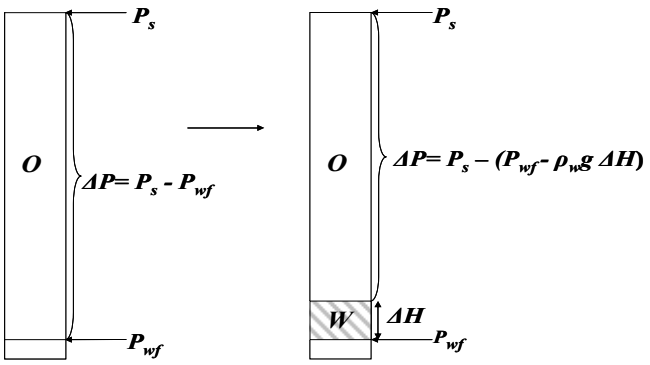

Figure 3: Schematic of principle of wells watering out

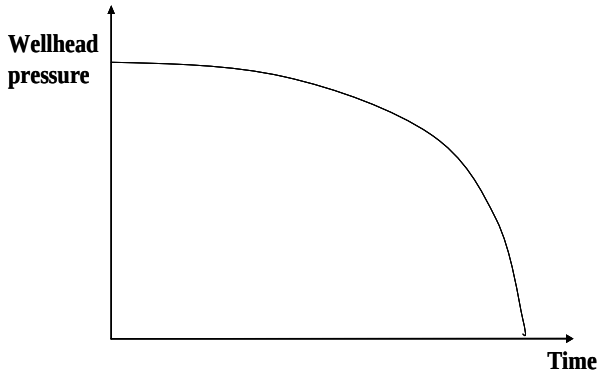

Figure 4: Sketch of pressure development for a well watering out

Pressure drop detection was chosen as an example to illustrate how even a simple agent structure could yield highly beneficial results.

c) Solution outline and agent structure

Many of the alarms and detection algorithms commonly used today are fairly simple. Basic thresholding directly on measurements or on gradients are frequently used, but these are not ideal methods, leading to both false alarms and missed events. It is our opinion that with more complex and better adapted detection methods, events and irregularities may be detected at an earlier stage, thus giving the operators a possibility to be proactive, react in time and prevent incidents.

Certain existing applications may offer this type of analysis, but in many cases this still requires the applications to be run independently (stand-alone applications), and with numerous applications for different purposes it may become difficult to keep track of alarms.

To detect the periods with pressure drops presenting the described properties, a simple algorithm was designed, and only two measured parameters were used.

The first parameter used is the upstream wellhead pressure, measured at the point indicated in *Figure 5*. The second parameter was the control position of the subsea production choke. The latter was included to allow for expected alterations to pressure due to corrections effectuated on the choke position.

The algorithm was designed to answer the following three questions:
1. *Is the general trend during a certain, predefined period decreasing?*
2. *Is the pressure drop gradient larger than a certain predefined limit?*
3. *Is the current pressure change simply the result of a choke change?*

Only two parameters were used for the analysis: the upstream wellhead pressure, and the choke position. For the demonstration case, each of these parameter analyses is represented by one agent.

Questions 1 and 2 above are answered by analyses performed by a “PressureCheck Agent”, while question 3 is answered by a “ChokeCheck Agent”.

The PressureCheck Agent

In the pressure check, the agent is scanning the input data looking for periods of minimum length of continuous pressure drop, or larger pressure drops over shorter periods. No specific filtering is used, but an intrinsic method in the database handling tool used for data collection returns interpolated values of the collected parameters when equal point spacing is demanded. This data treatment is necessary as the data acquisition frequency depends on system settings and parameter stability and is variable.

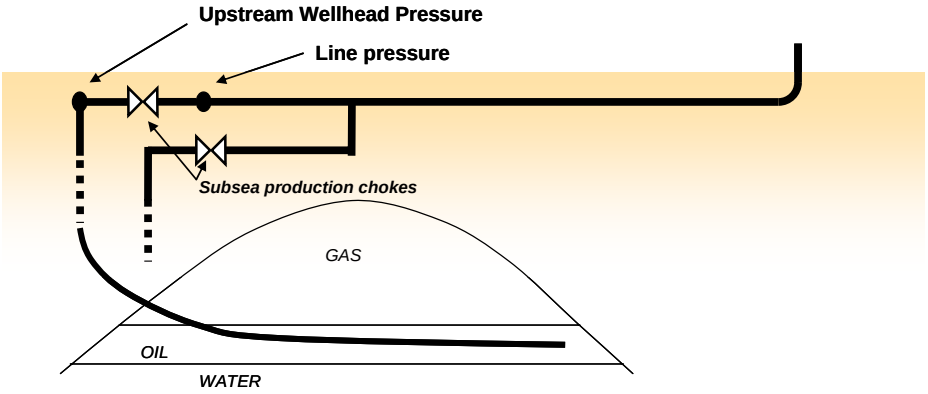

Figure 5: Illustration of lines and locations of pressure gauges

Check 1: Is the general trend decreasing?
To determine if the general trend is decreasing, the continuity of the sign of the gradient is analysed. The gradient is here taken to be the pressure difference between points with a certain timespacing, and a default value for this was taken to be 15mins or points. This is illustrated in Figure 6.

In the output from the pressure development analysis, a "1" means that the pressure is dropping, while a "0" means that the pressure is increasing or remains unchanged. The direct output from this step, without any further counting or analysis, is shown in Figure 9.

The method for determining the continuity of the drops is illustrated in Figure 7.
From the first "1" indicating falling pressure, the variable *Ones* counts the number of successive points detected as pressure drops. The default period for the continuity check was set to 150 mins, meaning that in principle, 150 consecutive ones would be needed to trigger the alarm. However, a certain noise level and local variations will always be present in the system. To account for this and make the system more robust we introduced a *buffer*, defining the number of errors allowed before the continuation count is reset. This means that even if a continued row of ones is split by zeros, the *Ones*-variable will keep counting until the number of consecutive zeros reaches the *buffer* limit. Only then will the number of ones be reset to 0. The default value used for this buffer was 30 mins/points. As soon as *Ones* passes the set limit (150 in our case) the alarm output state is changed to "on".

The output from this test is shown in red in Figure 10.

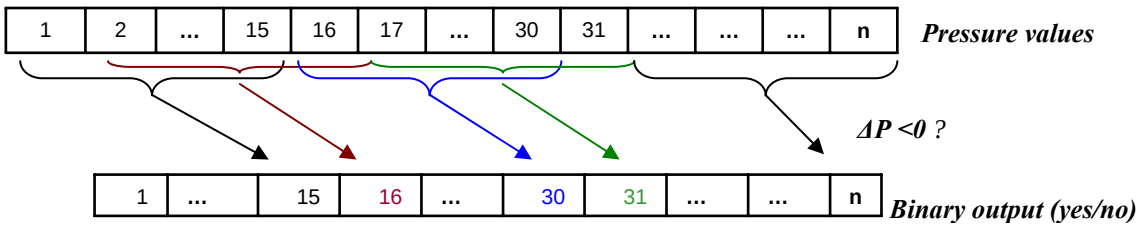

Figure 6: Illustration of drop continuity analysis. The numbers represent time point/point numbers.

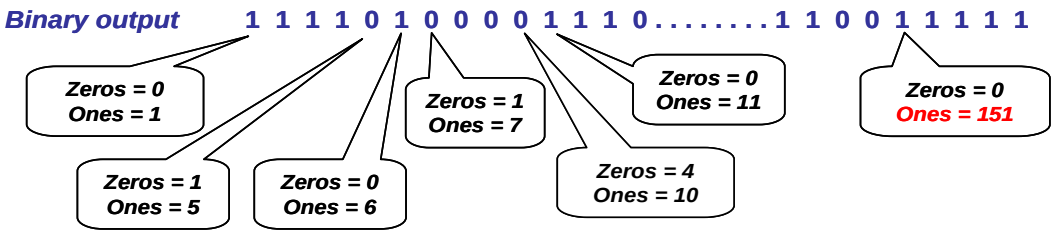

Figure 7: Illustration of counting method for drop continuity test

Check 2: Large drop gradient?
If the pressure drops fairly slowly, the default time period for the continuation checks works fairly well.

However, depending on the mechanism triggering the water accumulation, the gradient may sometimes be larger already from the beginning. In such cases a verification period of 150 mins may be far too long, and a different trigger mechanism on the alarm is needed. To ensure this, the total pressure drop during the last 30 mins is continuously checked, and if this value is larger than a given value, the alarm is triggered. The default value used here was 3 bars, but as the fluctuation levels may vary between wells, this value will more likely need adjustment to different wells than the time period definitions.
The results from this part of the analysis are shown in green in Figure 10.

Analyses 1 and 2 are performed by the same agent (*PressureAgent*), resulting in a single output from the pressure analysis – *pressureAlarm output.* This is shown in Figure 11.

The ChokeCheck Agent
Check 3: Expected pressure change?
This question was addressed using a very simple, but effective check. Any pressure development occurring during a fixed period after changes to the HSV control position is ignored. In this part, a change period is given an output value of "0", while an output of "1" indicates a choke stable period, in which no pressure changes due to choke settings are expected. An example of this is shown in Figure 12.
The default value used for this period was 180mins.
Another possibility concerning the buffer length could be to make it dependent on the size of the control position change. Such a change would probably not be very complex to implement, and would add both flexibility and precision to the system.

This analysis is performed by a second agent, giving a *chokeAlarm output*.

Final Output
To obtain the final output state of the drop detection system, the alarm outputs from the two agents are compared by a *CrossCheck Agent*, and the *finalAlarm* output periods are the periods for which both the *pressureAlar*m and the *chokeAlarm* were in an *"On"*-state. The structure of this system is illustrated in Figure 8.

Figure 13 and Figure 14 show the final output for two examples, the first one being the case used for the illustrations in the previous part.

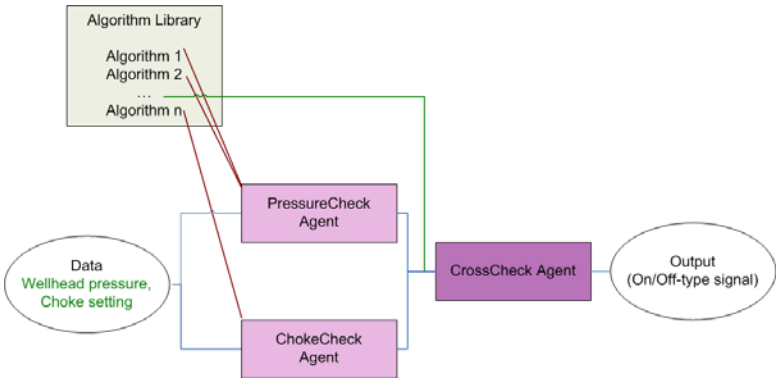

Figure 8: Illustration of structure of example case

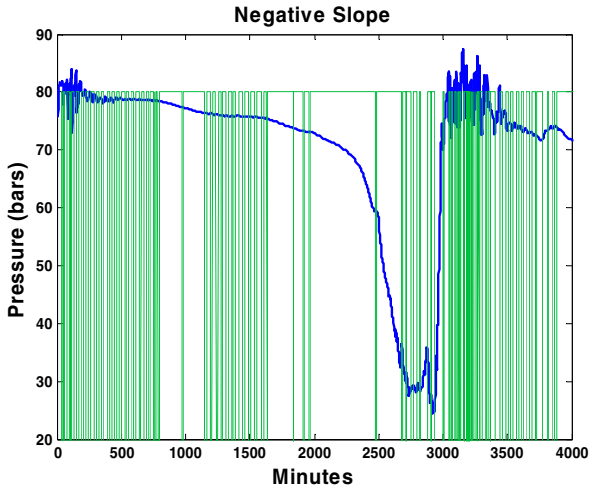

Figure 9: Illustration of results if only negative slope was considered

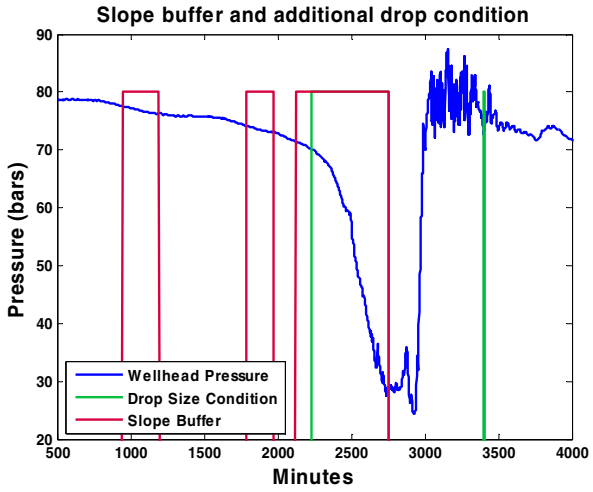

Figure 10: Result from pressure analysis after including continuity check and buffer, as well as the additional condition on drop size

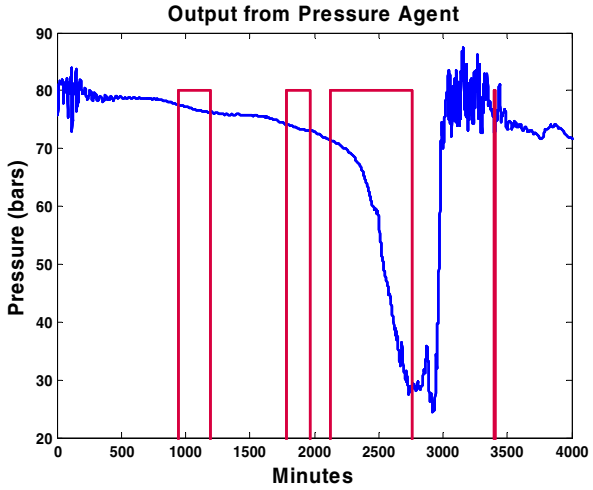

Figure 11: Final output from the pressure analysis agent

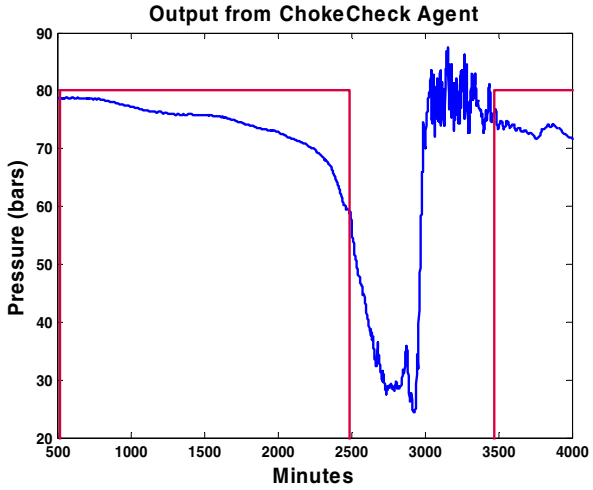

Figure 12: Output from the choke change analysis

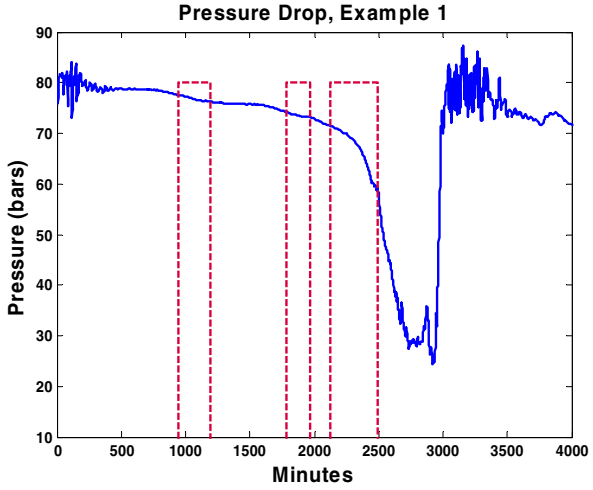

Figure 13: Pressure drop and corresponding drop detection signal (same example as used in algorithm illustrations)

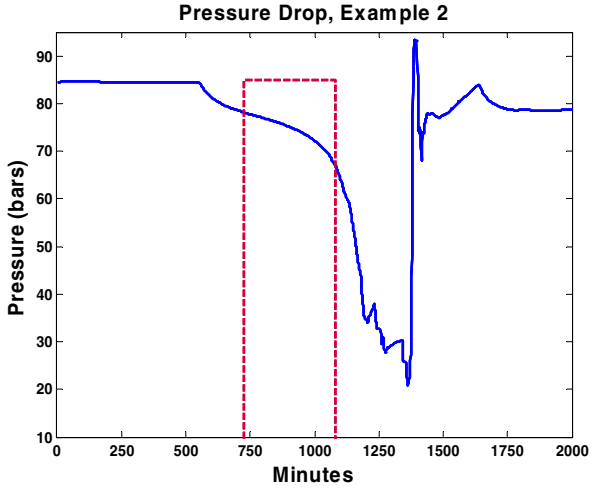

Figure 14: Example 2 of a pressure drop and the corresponding drop detection signal.

Results and Discussion

The algorithms described in the previous part were designed and used in an agent structure to enable an automated detection of pressure drops of the type experienced when wells are watering out. The agents would then trigger alarms drawing attention when pressure developments satisfying predefined conditions occur. In our case, an alarm would be triggered if the pressure trend is generally decreasing over a period of 150mins, or if the pressure drop is of 3 bars or more over a period of 30 mins. During a period of 180mins after changes to the choke position, pressure alarms are ignored by the system.

Detected events
The dotted, red line in *Figure 13* shows the output from the drop detection system. As the pressure seems to stabilize in periods, the drop is detected in blocs and not as one continuous decrease. However, this is not a problem, as the objective is to *draw the attention* to wells needing closer observation. In this specific case, a production engineer would most probably discard the first decrease period as non-critical without making any changes. The second time the decrease is indicated, the engineer would be more aware of the situation and most probably make changes, and the third time action would definitely be taken.
Other examples can be seen in *Figure 14* and *Figure 15.* In both these cases it is likely that an early alarm provided by this system would lead to changes being made, and the events here could have been avoided.

In total, continuous data for 4 wells, over a period of 5 months, was used. The acquisition frequency of the data spans from 1 to 10 minutes intervals, depending on acquisition settings and parameter change rates, while the agents are designed to operate with data with 1 minute frequency. The needed interpolation is currently performed by the data collection tool, and not by a data preparation agent as part of our system.

The “Actual drops detected” are potentially critical drops where only time will show whether the drop continues or restabilizes naturally. Interaction by choke adjustment may be needed, and thus an engineer should be informed about the development to allow manual inspection of data.
“Shut-downs detected” are drops that remained unnoticed until the pressure drop was already too low, and production had to be significantly reduced or stopped to restabilise the system.

A summary of the results are shown in Table 1. 77 out of 91 alarms, or ***85%,*** were actual drops of which detection is desirable. The majority of the false alarms were due to relatively large and fairly regular fluctuations where the pressure drop per 30 mins was larger than the pre-set limit. However, as these pressure variations present no risk to the system, one could consider to set a higher gradient threshold value or to disable the gradient evaluation function for wells with large pressure variations.

Unfortunately, some drops were still unnoticed by the program. “Missed drops (undetected)” are drops that should ideally have been detected, but that for various reasons were missed. “Missed shut-downs (undetected)” are critical drops which ended up leading to a shut-down, but that were not picked up in time by the agent.
The undetected drops were mainly due to the fact that a fairly regular and periodic variation seemed to be superposed to the general trend, thus resulting in local extremities making the trend less obvious. Advanced filtering could probably solve this problem, but at present this option has not been further explored. However, even when these missed drops are counted, 77 of 89, or ***87%,*** of the drops are detected.

In our test-data, 5 drops resulted in situations where one or several wells had to be shut in or production severely reduced to allow restabilisation. Of these, only one of these was not detected in time for restabilisation to be secured, which means that 4 out of 5 cases were successfully detected. In the case that failed, the pressure is falling very quickly from a stable situation. It had already dropped too far down by the time the alarm is raised, and major intervention is needed to reestablish production.

Table 1: Results from tests based on data from 4 wells during five months.
***This drop was detected, but probably too late for successful correction**

Well	*Alarms*	*Actual drops detected*	*Shut-downs detected*	*Missed drops (undetected)*	*Missed shut-downs (undetected)*
1	*17*	*9*	*2*	*4*	*0*
2	*53*	*48*	*2*	*2*	*0*
3	*5*	*5*	*0*	*3*	*1**
4	*16*	*15*	*0*	*3*	*0*
Total	*91*	*77*	*4*	*12*	*1*

Value estimation

Figure 16 shows an illustration of the principle behind an estimation of the additional production achievable by early detection, while *Figure 15* shows the corresponding pressure development. The data represent the production from a cluster where one well is watering out, but where several wells are producing on the same line. The involved well has a fairly low production rate, and the impact of a reduced production from this well itself is small. The main problem is then the effects the situation has on the adjacent wells, as it becomes necessary to reduce production on larger wells to reestablish production once the pressure has fallen too far. In the estimation we assumed that the production would be reestablished after the first dip, resulting in the red line on the plot. As can be seen, there is a delay between the alarm and the point of production reestablishment.
The system needs time to react to changes and adapt to the new flow conditions, and thus a certain delay is to be expected.

In this particular case, a total of 400Sm^3 in additional production was estimated, which is equal to around 0.5% of the monthly production from this cluster. It should be noted that the delay would most probably be smaller than what was used here, leaving a potential for even higher gain.

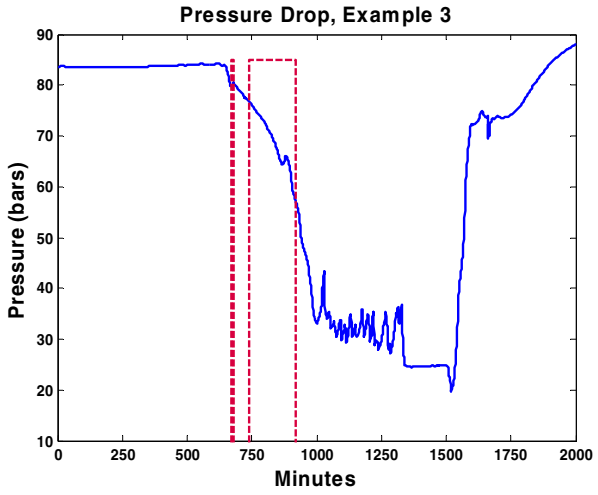

Figure 15: 3rd example of pressure drop and the corresponding drop detection signal (Same time-scale as Figure 16)

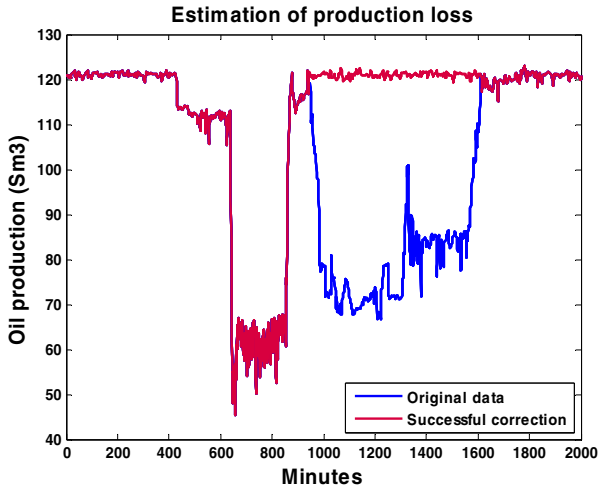

Figure 16: Example of estimation of value potential. The red line illustrates a rough estimation of a potential production development if a successful correction had been made. (Same time-scale as Figure 15)

Conclusion

In this work we have described an agent structure for event detection by trend recognition.
In the example problem of wells watering out, the results indicate considerable benefits of such a system. The system could easily be expanded to include more analyses, and the agents can make use of each other's results to adapt their analyses to the current circumstances.

Using a fairly simple algorithm, the aim was to detect pressure drops that were consistent over time, and/or dropped with a certain gradient. More than 85% of the pressure drops occurring were detected. At least 85% of the alarms obtained were actual pressure drops, but this number will most probably be even higher if an advanced filter is applied. 4 out of 5 critical pressure drops leading to shut-ins could most probably have been prevented with actions taken immediately after alarms from the suggested system. In one of the example cases, the production loss that could have been avoided was estimated to around 400Sm3 or 0.5% of the monthly oil production from that specific cluster of wells.

We have shown that events can be detected based on the trends observed directly, and in this case without specific knowledge about and application of physical models describing the phenomena. However, it is our belief that the flexibility and structure of an agent system is very well suited for including first-principle models and/or links to more demanding modelling applications, for further to combine different types of information and analyses.

- Alarms drawing attention when needed may be an alternative to manual routines for trend monitoring
- Considerable production losses may be avoided by early detection of pressure drops
- Crosschecking between pressure development and choke setting changes allows a reduction in false alarms
- This type of trend recognition/detection may be achieved using simple algorithms, without a deep physical understanding and extensive modeling
- Trend-detection is also an important enabler for event-driven workflows

Acknowledgements

The authors would like to thank StatoilHydro for their participation in this project, and in particular Espen Halvorsen, Discipline Advisor Production Steering and Jan Hauge, Department Manager Troll Production Technology, for their help and support. We are also grateful to the Troll license holders: Petoro, Shell, Total, ConocoPhillips and StatoilHydro, for allowing this work to be published. Any opinion expressed are those of the authors and do not necessarily represent the view of the license partners. In addition, we acknowledge the support from the Research Council of Norway for this project (168120/S30 Petromaks).

References

[1] NASA SISM: *Intelligent Systems Project: Automated Reasoning*, http://is.arc.nasa.gov/AR/index.htm

[2] Press release, 2004: *First Flight – True UAV Autonomy At Last.* http://www.agent-software.com.au/shared/resources/pressReleases/Avatar-JACK-F040706Ausb.pdf

[3] Landre, E., Ølmheim, J., Wærsland, G.O., Rønneberg, H.: *Software agents – An Emergent Software Technology That Enables Us To Build More Dynamic, Adaptable, and Robust Systems.* SPE 103354, presented at the SPE Annual Technical Conference and Exhibition, San Antonia, Texas, 24-27 Sept 2006.

[4] Russel, S., Norvig, P., 1995: *Artifical Intelligence: a Modern Approach, p33.* Prentice-Hall, Englewood Cliffs, NJ.

[5] Franklin, S., Graesser,.A, 1997: *Is it an agent, or just a program?* In *Intelligent Agents, III* (eds. J. P. Müller, M. Wooldridge and N.R. Jennings), LNAI Volume 1193, pp.21-36. Springer, Berlin.

[6] Nwana, H.S., 1996**:** *Software agents: An overview*. Knowledge Engineering Review, *11 (3).: 205-244*

[7] Erskine, S., Schmidt, H.S, Nygaard, O., Nordtvedt, J-E.: *Reservoir Management of Gas Fields Using Permanent Sensors forPore Pressure Monitoring: The Troll Case.* SPE 71518, presented at the 2001 SPE Annual Technical Conference and Exhibition held in New Orleans, Louisiana, 30 Sept-3 Oct 2001.

[8] Mikkelsen, J.Kr., Norheim, T., Sagatun. S.I.: *The Troll Story.* OTC 17108, presented at the 2005 Offshore Technology Conference held in Houston, TX, 2-5 May 2005.

[9] Hauge, J. Horn, T.: *The Challenge of Operating and Maintaining 115 Subsea Wells on the Troll Field.* OTC 17111, presented at the 2005 Offshore Technology Conference held in Houston, TX, 2-5 May 2005.

[10] Lea, J.F, Nickens, H.V, 2004: *Solving Gas-Well Liquid-Loading Problems.* SPE 72092, Journal of Petroleum Technology, Vol.56, Number 4, p.30-36.

[11] Personal communication with Dr. Arild Bøe