SPE 112078

Improving Production By Use of Autonomous Systems

Jørn Ølmheim, Einar Landre, and Eileen A. Quale, SPE, StatoilHydro

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

Several companies have reported from 2% to 5% increase in oil production, and in some cases up to 10%, by a closer collaboration between onshore-based production optimization groups and offshore personnel. Several aspects have become apparent:

- shorten reaction time to implementation of optimization measures enables staff to take full and immediate advantage of opportunities
- the work is manpower-intensive during the collaboration sessions, but ensures continuity and easy access to technical competence
- engineering resources are already scarce, and there is a need to relieve staff of minor, routine tasks to enable attention to be focused on realtime issues requiring human expertise
- increased level of instrumentation complicates the work because it becomes increasingly difficult to utilize the available information to make better decisions

Other industries in information rich and highly dynamic domains have over years utilized the capabilities provided by autonomous systems and their dominant implementation platform, software agents to improve the decision making process. Over the last few years in collaboration with an autonomous software platform vendor, StatoilHydro have tried to apply the experience from other domains on some of our core business processes, concentrating on improving the volume of produced hydrocarbons and the regularity of the production process.

The work so far shows that in certain circumstances, autonomous systems can contribute to more effective production. In particular they enable more frequent adjustments in response to the actual well conditions. The main difference between the autonomous systems and more traditional automation is the ability to keep the human in control.

In our latest attempt we have applied the concepts of variable and delegated autonomy from the domain of unmanned flight on our production process. Compared with more traditional automation, the concepts of delegated and variable autonomy introduce negotiation between different sub-systems and a much more adaptable human-machine interaction.

Introduction

The trend within the oil industry is towards increased instrumentation and the use of "smart" well technology for new developments. This has contributed to an increased information flow into the control room. The control room operators face new challenges with respect to their ability to utilise this information. From other industries, it is documented that increased information flow can contribute to information overload and lack of shared situation awareness between key stakeholders.

In the control room of a large installation with 100 wells, where each well produces 15 different signals, 1500 different signals need to be processed at any given time. Humans are not able to process this flood of data rapidly and efficiently.

One of the objectives of Integrated Operations is better utilization of technical experts, independent of location. For this vision to become reality, shared situation awareness between control room operators and remote experts is a prerequisite. Achieving the required level of situation awareness will be considerably assisted by an increased level of computerized support that is able to analyze complex data streams and transform the raw data input into events and derived recommendations for further analysis and action.

The problems of information overload and lack of shared situation awareness are not unique to StatoilHydro or to the oil industry. In fact they have been addressed in other domains such as military command and control, space exploration, unmanned flight, and industrial planning and resource scheduling by use of autonomous systems in the form of multi-agent systems. Examples of such applications include space shuttle ground processing [6], engine power management for unmanned vehicles [2] and decision-making capabilities in unmanned air vehicles [7, 8].

Over the last few years, StatoilHydro, (formerly Statoil), has investigated how autonomous systems might be used to improve core business processes. The approach has been to develop a set of demonstrators exploring the potential of multi-agent systems to meet the challenges in upstream hydrocarbon production.

The primary objective of this paper is to present the experience with autonomy and software agents in StatoilHydro, and to outline what is seen as the next generation of systems for the purpose of improving hydrocarbon production.

Hydrocarbon production practices and challenges

Planning of production is based on an agreed production target or profile, in accordance with the field's drainage and injection strategy, or on a delivery contract obligation. The individual well's contribution at any given time will depend upon the required production rate and other factors such as limitations on wellhead or downhole pressure, sand production potential, equipment restraints, coning/cusping conditions, velocity restrictions etc. In addition to the individual wells, total production to a facility will be dictated by the available handling capacity for the different fluid phases and export requirements, as well as necessary maintenance work. A facility's production profile will generally be characterized as "well-limited"; where well potential is the limiting factor, or "process-limited", where the production facilities are the bottleneck.

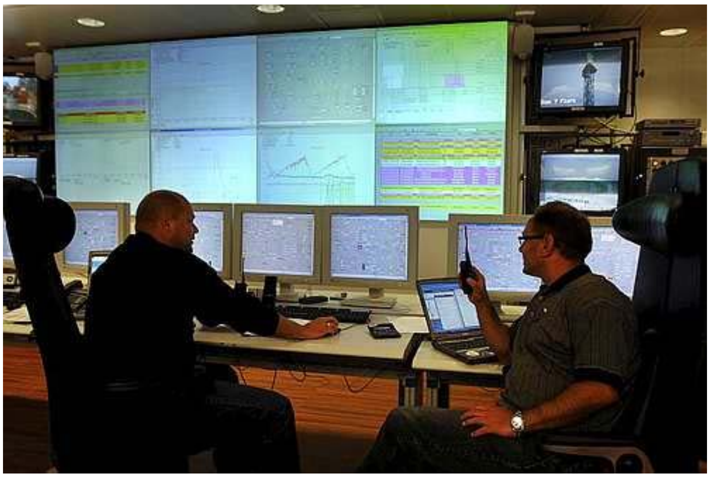

Figure 1: The control room at Åsgard B

Optimization of production will thus be based on addressing limiting factors and adjusting the feed stream to a level and composition which makes best use of the available facilities as well as minimizing lost or deferred production due to maintenance. A given premise is that the operations are carried out in accordance with good HSE practice.

While "optimize" and "maximize" are often used interchangeably in description of production operations, these terms are not synonymous. In perhaps the majority of cases, optimization will be based on maximizing (oil) production, but this may not be the best long-term solution with respect to optimization of recovery or reservoir management. Feed stream mix requirements may also limit production to a level below the maximum volumetric rate which could be achievable. These factors must be taken into account in the production planning.

Traditional production optimization is directed towards achieving the production target by adjusting well contributions and process parameters. Typically, this is carried out by observation/analysis of producing parameters by production and process engineers, and subsequent adjustment of production set points to improve total output. Control room operators may also make adjustments based on the performance of the production facility and possible current equipment problems. Figure 1 shows the control room on the Åsgard B platform, and gives an idea of the level of information being directed at the operators at any one time.

This assessment and adjustment exercise will be carried out on a regular basis, and now more commonly, in "real-time" through use of collaboration rooms involving onshore and offshore staff.

Automation and the role of well instrumentation

With the advent of new systems and an increased number of sensors and analysis tools, the engineers face new challenges with respect to processing of information. While the tools take care of the "number-crunching", the engineers are still required to assess the results and to recommend a best course of action. In certain cases, an automated system could help to decrease the load by automatic implementation of actions and adjustments in accordance with well-defined and pre-approved scenarios.

The involved level of applied automation of the adjustment process will vary according to the logic programmed into the production control system and will be proportional to the level of confidence the engineers have in system response to adjustments. Well behaviour and response to a given adjustment will not be 100% predictable and are always subject to variation and error. In most cases for example, a well's production capacity and response are not measured directly, but derived from other measurements. Despite the use of "expert" analysis and control systems, it is generally true that the expertise is provided by the engineers and most system changes with respect to well management will be approved by the engineer before implementation.

For many oilfield operations, instrumentation for well control and monitoring, such as choke, safety valve and measurement of wellhead pressure/temperature conditions is standard in all completions. Additional well instrumentation for data acquisition will contribute to a better analysis of well conditions and a correspondingly better basis for prediction of well behaviour. Instrumentation in this context means sensors such as downhole gauges, downhole valves, sand detectors and multiphase flow measurement. This is equipment which is not always present in a well completion, but which will enhance understanding and exploitation of the well's potential. The presence of this instrumentation will not in itself indicate a higher level of automation, but in certain cases gives the potential for fine-tuning of the production stream. Aside from the challenges in installation of downhole sensors, their usefulness will depend upon reliability both in function and in quality of the data delivered. Where these are in doubt, an automated system based on their input will not function optimally and may be seriously in error.

Optimization of production between wells implies further that the well's contribution at given conditions is known, and within a reasonable margin of error. Modern multiphase meters give a continuous and direct measurement of rate, subject to the meter's reliability and accuracy. However, in many cases, the main method of determining well performance is by periodic welltesting through a test separator. This will give an estimate of production under given conditions, but the estimate will become inaccurate over time, until fresh data is acquired by new testing.

These common uncertainties in system behaviour prediction are a major barrier to moving from automation to autonomy.

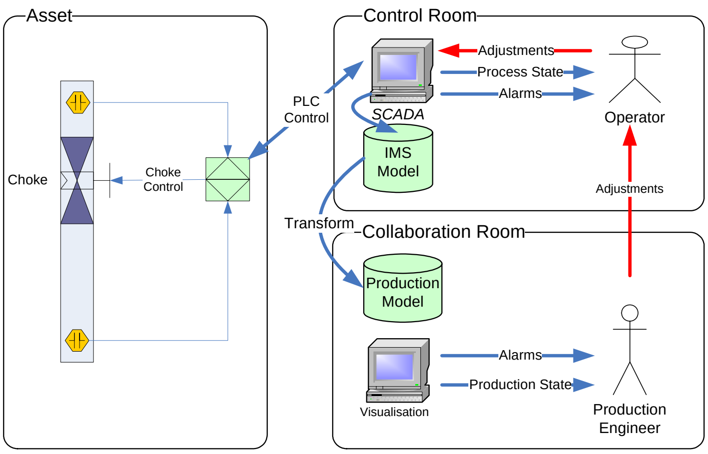

Figure 2: Traditional production control model

Figure 2 illustrates a production control flow and the typical communication between onshore production engineers and the offshore control room operator. The production engineer plans and monitors the asset with respect to meeting production goals for the asset, while the operator's responsibility is to implement the production plan and monitor the process state and the general performance on a continuous basis.

There are three major problems related to this setup:

- Large amount of information to be processed by the involved parties.
- Differing situation awareness. The parties have different perspectives of the situation with respect to priorities and criticality.
- No mutually-agreed timeframe for action. Prompt implementation of optimization measures will give best results.

The establishment of collaborations rooms is a big step towards achieving a common platform and understanding of the current situation and action required.

Improved Computer Support

Improved sensor technology and high speed computer networks have transformed our core production assets into high volume data sources that can easily overload the human operators, presenting new challenges in effective use of the additional data in the decision-making process. By considering the experience gathered in other domains it seems that improved computer support in the form of autonomous systems can help address some of the identified problems.

In Figure 3, the production visualization system from Figure 2 is extended into a production control system. This is an autonomous system capable of making decisions and directly controlling the production process. The design of this process control system is a non-trivial task, and autonomous systems are ideally suited for the implementation.

To understand the potential of this system it is important to understand what an autonomous system is and particularly the difference between automation and autonomy [2]. What might seem at first sight to be subtle differences could have a significant impact on the way human operators interact with industrial control and decision support systems, including the design of the human system interface.

Automation [1] is defined as:

- An activity carried out by prior arrangement when certain conditions are fulfilled, without the need for a decision.

Autonomy [1] is defined as:

- Having the ability to make decisions and act on them as a free and independent agent. Having the freedom to determine one's own actions.

It is the ability or authority to make decisions that distinguishes an autonomous system from an automated system. A practical example of this difference is illustrated by an aircraft. The autopilot automates the process of keeping the aircraft on a predefined course, while the aircraft's autonomy is represented by the pilot and the authority delegated to the pilot. Applying this approach on our domain implies for example that the choke can be opened and closed to achieve a stable upstream pressure. In this case the produced fluid can be gas, water and oil, the automated control loop does not make any distinction.

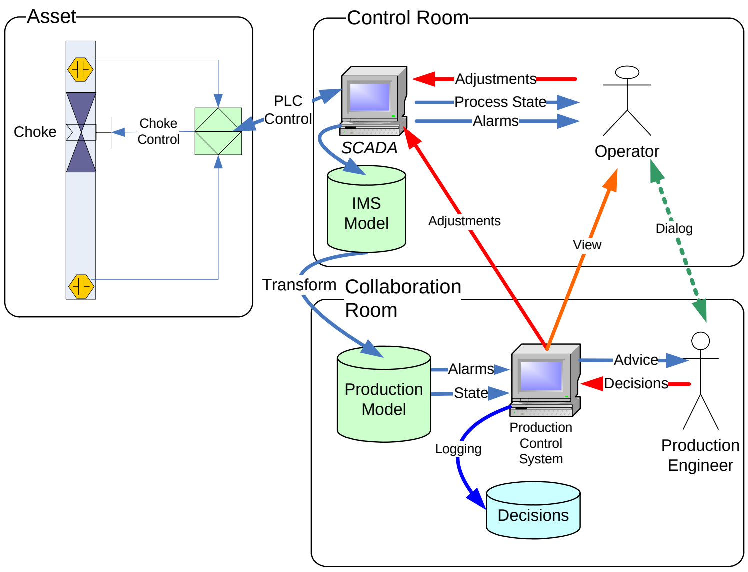

Figure 3: Proposed production control model

The oil and gas industry has been a very slow adopter of large scale automation in the production area. The reasons are many and include too few and erroneous measurements and inadequate or unreliable process models. In addition to these technical reasons, the industry is known to be conservative, and sceptical to the viability of automation. This is especially true where the safety of personnel may be perceived to be put at risk. This conservatism is something that must be taken into account when introducing extended computer support. Good communication of the systems' capabilities and limitations will be necessary to establish trust in both automation, and the further step of autonomy.

Variable and delegated autonomy

When designing and building support systems it is critical to understand which functions/actions the computer is authorized to perform on its own, and which require human approval, as well as the form of the actual approval. In the aerospace industry this is called variable autonomy [1].

Variable autonomy means that a computer system can be authorized to make some decisions and perform certain actions independently, while other decisions require human approval. An example of variable autonomy is defined by the PACT (Pilot Authority and Control Tasks) taxonomy for unmanned flight. In [1] the PACT levels have been modified and aligned with Sheridan & Verplank's levels of automation. This is illustrated in Table 1.

PACT Locus of authority	PACT Level	Sheridan & Verplank Levels of HMI
Computer monitored by pilot	5b	Computer does everything autonomously
	5a	Computer chooses action, performs it and informs human
Computer backed up by pilot	4b	Computer chooses action and performs it unless human disapproves
	4a	Computer chooses action and performs it if human approves
Pilot backed up by computer	3	Computer suggests options and proposes one of them
Pilot assisted by computer	2	Computer suggests options to human
Pilot assisted by computer only when requested	1	Human asks computer to suggest options
Pilot	0	Whole task done by human except for actual operation (autopilot)

Table 1: Taxonomy for variable autonomy

The beauty of the modified PACT levels is that they correlate a defined level of autonomy with what that level means with respect to the human-machine interface. In addition, it provides a framework for classification of decisions with respect to level of autonomy.

Applied to the domain of hydrocarbon production, variable autonomy can be illustrated by the following scenario: A production well might be given the authority to adjust its choke within a limit of 10% while adjustments in the range of 11% to 40% are accepted if the human operator does not disapprove (4b), whereas larger adjustments require human approval (4a).

This model of variable autonomy can be extended into what is called delegated autonomy [2]. The model of delegated autonomy is basically the model found in human organizations. A manager can delegate authority and responsibility to subordinates, while at the same time being accountable for the decisions made by the subordinate.

In our domain delegated autonomy can be illustrated in the following way: two autonomous components exist; a production well and a field manager. The field manager is given a production objective for a specific period of time. The field manager requests a contribution from the well. The well can then respond: “I am unable to comply with this request without lowering bottomhole pressure below the minimum recommendation. I can fulfil 80% of your request within the recommended conditions”. The field manager can in this situation accept the offer, and try to get the remaining 20% from some other sources, or decide to overrule the well and let it run at 100% for a specific period of time. If the field manager overrules the well, the responsibility for the consequences lies with the field manager.

The key property of systems with delegated and variable autonomy is that there are some intricate patterns of collaboration and complex decision-making that must be captured and implemented in software. Such implementation has proved non-trivial, and there is one class of computer systems that has proved itself very useful in this area, namely multi-agent systems.

Multi-agent systems

Multi-agent systems are a class of computer systems capable of independent autonomous action in order to meet their objectives. The software agents are capable of deciding for themselves what to do in any given situation. They are known as rational agents because they make good decisions about what to do, meaning that they are predictable in their behaviour. They can act reactively, i.e. respond to changing environment or proactively, i.e. act in anticipation of future goals [3]. For a more detailed description of software agents and multi-agent systems, see [3, 9].

JACK is a multi-agent platform from the Australian company AOS [4]. JACK is based on the BDI [3] (Belief, Desire and Intention) model of artificial intelligence and is one of the most mature multi-agent platforms available on the market. It is used by a wide range of corporations including BAE Systems and Rolls Royce in addition to the British, Australian and American military.

JACK provides a set of concepts that make the development of decision-making software simpler, compared to more traditional general purpose programming languages such as Java, C++ or ADA. These concepts are:

- Agent – a named entity that responds to events, pursues goals and executes plans.
- Capability – a grouping mechanism for related functionality described in terms of “capable of doing”.
- Plan – a function, or more precisely, a piece of logic performed as a response to an event. A plan is allowed to fail and if so, the agent might decide to perform an alternative plan.
- Event – represents something that triggers the agent to perform some work. In the case of JACK, events are bearers of goals i.e. what the agent should try to achieve.
- Beliefset – a data structure used by the agent to store facts about its environment.

The concepts of JACK are natural constructs provided to help the developer with the design of the system. Charles Patchett, BAE Systems, stated that they needed the constructs of JACK to be able to implement decision making capabilities for the UAV (Unmanned Aerial Vehicle) after failing to build it in ADA (a general purpose programming language).

In addition to the programming constructs, JACK provides powerful reasoning capabilities in the form of plan selection, event logic and beliefsets. These reasoning capabilities combined with the run-time capabilities of JACK, make it possible to create truly autonomous systems. JACK is designed so that agents run independently of each other, and uninterrupted by the rest of the system. The agents collaborate by passing messages called events. The JACK platform is inherently distributed, and each agent has its own communication portal to which messages are routed. This makes it possible for each agent to be distributed on a separate machine.

Experiences with multi-agent systems in StatoilHydro

StatoilHydro has a relatively long history with multi-agent systems. The initial application was for implementing smarter process support for a trading system portfolio, evaluating rule-engines and agent platforms. Based on experience from other industries, multi-agent systems were introduced to attempt to achieve similar results. The first exercise with agents was a workflow prototype planned to be included in the trading portfolio. The second system was a jetty planning system for an oil terminal.

The experiences from these two trials were assessed for applicability to other core business domains in StatoilHydro. This potential was outlined in [9]. This paper also introduced an agent model that became the architectural baseline for the later systems. The Heidrun agent demonstrator project was based on this architecture, combined with the experience from the jetty planner, and was the first attempt at a multi-agent system for production optimization.

Each of these systems will be described in the following sections, to analyze the lessons learned from each.

Workflow prototype

The workflow prototype was built as part of a development program tasked with replacing a trading system portfolio. It had been identified that smarter process support was required, and software agents were considered to hold the key to providing this. One example of the problems to be solved was contract document generation for a new oil trade (Deal). The process to be supported can be described as follows:

- The trader has agreed on a sale or purchase of oil with a customer and entered it into the trading system
- Business rule: A contract document must be generated, approved and sent to the customer within 24 hours
- Steps:
    1. Generate contract document
    2. Approve document

The basic architecture of the workflow solution is shown in Figure 4. The system consists of four different agents:

- Process Controller
- User
- InfrastructureBroker
- TaskBroker

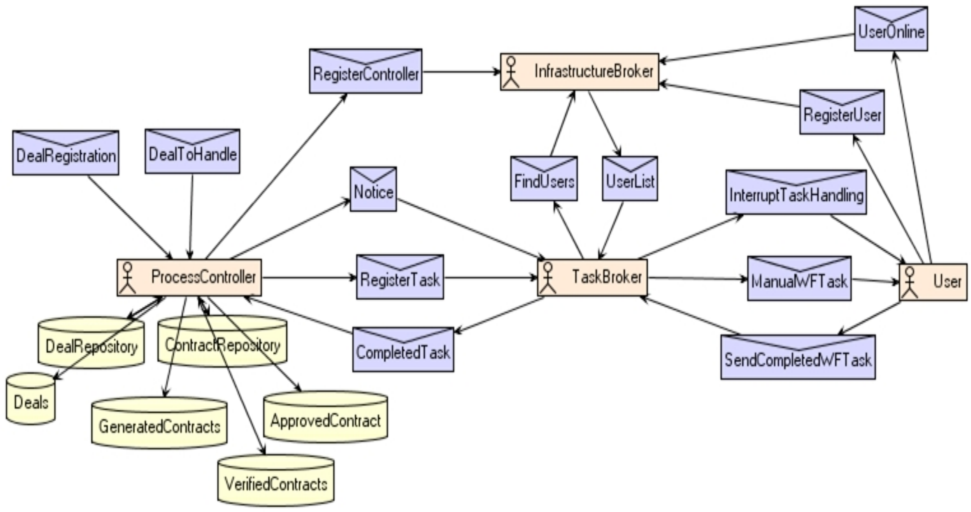

Figure 4: Design overview of the Workflow prototype

The Process Controller represents the business process, in this case Contract Management, and is responsible for dealing with tasks in the system. The tasks in this scenario are to generate the contract document and have it approved. A central idea in the system is that the Process Controller shall perform as many tasks as possible without user input, and in fact it is able to generate contract documents for most types of deals. This would be a huge improvement over today's systems, where all documents are user-generated. The tasks that the Process Controller cannot perform on its own must be forwarded to a User. This is done through a Task Broker and an Infrastructure Broker.

In the prototype there is one Process Controller, and several User agents that represent users with different roles such as Contract Responsible, Trader and Manager. The Infrastructure Broker keeps track of these users and whether or not they are active, i.e. logged on. The Task Broker is responsible for forwarding tasks to all relevant User agents, and monitoring the progress of a task. The User agent maintains a task list for its user, and as soon as the user opens a task on the list, the User agent notifies the Task Broker that the task has become active. The TaskBroker then removes the task from all other users' task list.

The 24 hour deadline is an important constraint for the design. The Task Broker monitors the progress of a task, and is able to retrieve the task from the user if the deadline approaches without completion, in order to resend it to other users. The Task Broker can also monitor the task load of the users as well as escalate the importance of tasks that has not been activated as the deadline approaches. The system is specifically tailored for the process, but can be generalized to handle other processes.

Lessons learned

The main lesson learned from this prototype was how the software agents could be used to follow up tasks. Experience was gained as to how an agent could maintain a "database" of tasks and their status, and also how the agents communicate to solve tasks. The auction model for distribution of tasks was also a new concept, and proved to be an effective method of handling conflict situations where multiple agent instances compete for the same tasks. This system was designed to be run on separate computers, and it proved the inherent capabilities of JACK to produce a truly distributed system. The communication mechanisms work exactly the same whether the agents run on the same machine or distributed across several servers and clients.

Mongstad Jetty Planner

The Jetty Planner was a demonstrator created as a training exercise over a fairly short period of time (weeks). The goal was to create a system that monitored the estimated arrival time of tankers coming into the Mongstad Terminal, and based on the relative cost, as well as size constraints of the vessels, plan the optimal (economic) use of different berths. Figure 5 shows a tanker berthing at Mongstad Terminal.

Figure 5: Tanker berthing at Mongstad Terminal

The loading time for the ship depends on its capacity and the capacity of the loading arms at the berth. Each berth also has only a limited set of products (Crude oil, different refined products and so on) available, so this was an additional constraint. This situation quite quickly becomes too complicated for human operators, and the current process is a fixed planned schedule that results in a considerable economic penalty when ships are not permitted to dock within their allotted time (demurrage).

Lessons learned

The Jetty Planner showed how agents can be used for scheduling of limited resources in a highly dynamic environment. We also learned that it was easily possible to create a simulator for the same environment that could be used for operator training or what-if analysis.

Heidrun Field Demonstrator

Heidrun demonstrator was the first attempt to use software agents for the purpose of supporting production engineers with advice, in line with PACT level 2. The potential from such a support system lay in two dimensions:

1. Improved monitoring of system state, i.e. sand detection, product quality (gas/oil ratio) and water management, where the improved insight into the system behaviour can be used to favour good producers in real time and thereby improve production.
2. Direct production engineers' focus towards the most important problems, i.e. produce advice such as: "Rising GOR detected at Well A-48, recommend choke-down from 20 to 15. Compensate lost production by choking-up Well A-6 from 15 to 25. Recommendation based on <list of facts>".

The project aimed at both dimensions with main focus on the development of a prediction model to support the production improvement advice. The idea was to utilize historical data for prediction of high-GOR well production, an attempt that proved difficult for several reasons. The necessary systematic historical data sets and reliable prediction models for the wells' response to changes in production conditions were lacking, resulting in a poor logical basis for the system's decisions.

The project recommended application of the solution to another field with better data quality and more rigorous well models. From a software engineering perspective such adoption would be quite easy due to the system's overall architecture, one agent representing the field and dedicated agents representing each individual well. This is illustrated in Figure 6. The well agent monitors the well state and reports anomalies to the field level. The field agent decides what action is to be taken based on reports from all well agents, and produces an advice which is presented to the user.

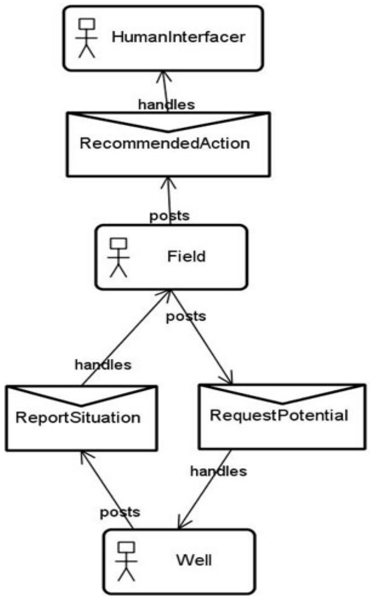

Figure 6: Software Architecture

Lessons learned

The main lesson learned from the Heidrun demonstrator was that it is possible to use software agents to provide decision support to production engineers. Their strength is their ability to process huge amounts of real-time data and compare real-world sensor readings with models.

The Heidrun demonstrator was to some extent unsuccessful in its chosen objective, as the test case was related to correlating GOR with choke setting, including the use of historical data for determination of optimal choke setting. This proved not to be a viable approach, due to the input data, although the model logic was correct. The old prediction model is planned to be replaced and tested on a field with a more complete historical database than Heidrun.

In parallel with the Heidrun demonstrator two master students at NTNU [5] pushed the ideas of autonomy in context of a production scenario to a new level. They developed a simulator to study the potential benefits of autonomy and computer control in production. The solution followed the same architectural outline as used for the Heidrun demonstrator.

The main difference between the students work and our own work at Heidrun was that in the simulated system the well agents were allowed to control the choke and directly influence the produced volume. The observations from several test runs were that agents performed better than humans (students) in the control loop, performing more rapidly and consistently. For us the students work documented the possibilities found in the technology.

Architecture for next generation production support

In this section some suggested concepts for inclusion in the next generation production support systems are presented. These systems should be based on use of delegated and variable autonomy. The suggested architecture is divided into three distinct layers illustrating the business concerns at each layer. The layers are shown in Figure 7.

Figure 7: Layering

Reservoir Management: Responsible for the long term objectives for each reservoir and defines the goals for the lower levels to implement. Uses reservoir models and supports what-if analysis of different scenarios.

Determines how each reservoir should be drained for the purpose of maximizing long term value in each reservoir and between reservoirs in a geographical area.

Field Operation: Responsible for the day-to-day operation of a single field. Uses field flowline models and simulations to find the "best possible" configuration.

Receives production goals for each of its reservoirs from reservoir management, and develops plans for how to manage the wells based on actual well state and designated production goals.

Well Monitoring & Control: Responsible for monitoring and control of the individual wells. Captures the uniqueness found in each individual well.

Compared with more traditional use of layers in software engineering, this approach differs as the components located in each layer will negotiate contracts as part of the delegated autonomy.

For simplicity we argue that the production is governed by two concurrent processes. There is one top-down planning process that is driven by the need to maximize value from the reservoir and there is one bottom-up monitoring and control process. The challenge is that these two processes are connected and they might have conflicting goals. This is illustrated in Figure 8.

Planning

The objective of the planning process is to establish a best possible configuration constrained by a production target and the current situation. This configuration will be challenged almost before it is put into operation due to the dynamics in the real world. Equipment will fail and situations emerge and new plans must be put into operation.

Monitoring& Control

The objective of Monitoring & Control is to observe the actual well conditions and to detect and correct anomalies. In the case where a well is unable to deal with the situation without breaking its obligations, it notifies the planning process of the failure. This will result in a gap with respect to the planned target and replanning will be triggered.

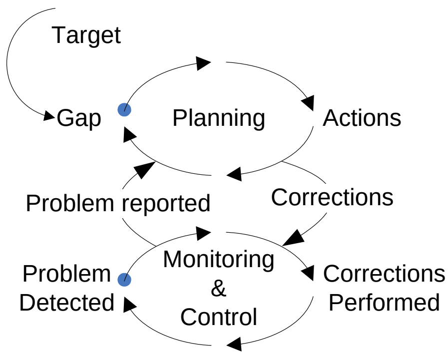

Figure 8: Conflicting processes

Proposed Agent Architecture

The introduction claimed that information overload and lack of shared situation awareness are the main challenges when the control room operators and production engineers are faced with the growing data streams provided by heavily instrumented production assets.

Based on experience with autonomous systems, combined with the experience from other industries, a conceptual software architecture may be developed which includes the key capabilities necessary in the next generation production support systems, systems which will require use of variable and delegated autonomy to be practically useful.

For humans to trust a computerised support system, the system must be designed with the human at its centre of gravity, and must behave in such a way that a human expert understands how it works. For the system to be regarded useful the experts must feel that the system behaves as a trusted assistant, helping them to do a better job with respect to the tasks designated to the system.

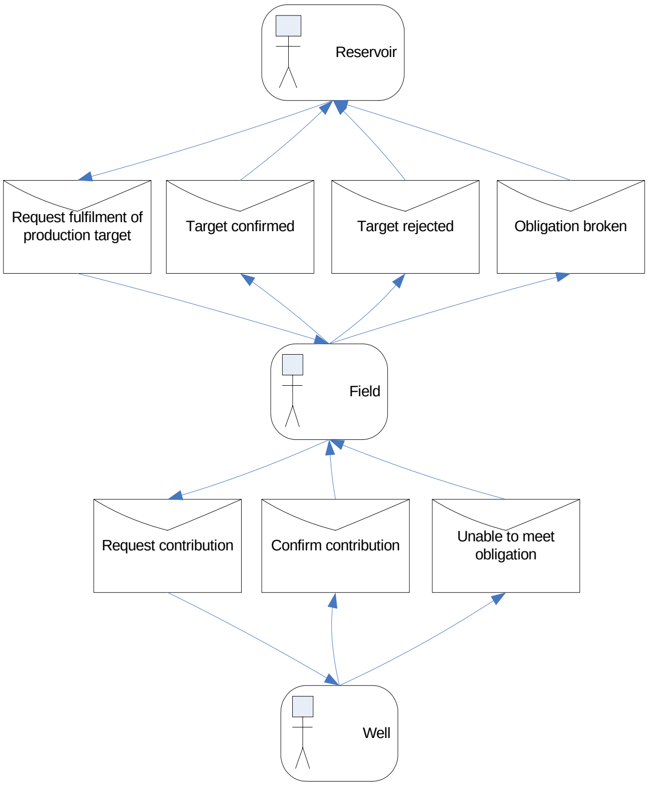

Figure 9: Conceptual Architecture

The conceptual design model of such a support system will be based on three agents representing the three process levels involved. See Figure 9 for details.

At the top there are agents representing the reservoir planning processes. The highest realistic level of autonomy for these agents is at level 3, i.e. an advisor. The main objective is to capture the decisions made by the humans and delegate these as production goals for the field agents' planning capability. With time the agent at this level can become a knowledge base due to the possibility for tracking decisions and aligning them with their effects. It may also be useful for the agents to interface with existing reservoir models.

In the middle, the agents representing each individual field are located. The responsibility of these agents is the execution of the field level planning process. This is a continuous process driven both from above (reservoir) and below (changes in well behaviour).

These agents will use a combination of existing (GAP) and new models. The field level agents must be able to operate on all autonomy levels between 2 (analyze situation and provide options) and 4a (analyze situation, suggest option, perform change if human confirms).

At the bottom we find agents representing each individual well. These agents will capture sensor readings such as temperature, pressures, product quality measures and sand sensor readings. These readings will be stored in the agent's memory and used to maintain individual performance curves of each well. When a well drifts off its actual setpoint, the well agent will try to push it back on track. Dependent of the degree of required change the agent will operate on all autonomy levels between 5a (analyze situation, perform change, inform human) and 2 (analyze situation, suggest options).

The agents are not meant to replace existing closed loop automations, but to handle situations that traditionally have required human intervention with respect to the responsibility of production engineers.

Odin Field Reference Model

To illustrate the involved dynamics we will use a simple field with three producers and one injection well and two reservoirs (formations) as a reference model. Our reference field is named "Odin" in accordance with Norwegian conventions based on Norse mythology. The two reservoirs are named after Odin's two ravens "Huginn" (thought) and "Muninn" (memory).

The Odin field is process-limited, implying that it is constrained by its processing capacity and the wells are given the names: P1, P2, P3 and I1, where we use P for production and I for injection.

The Huginn formation is low pressure and requires injection, while Muninn has sufficient pressure and is produced by natural depletion. The field with its wells and reservoirs is illustrated in Figure 10.

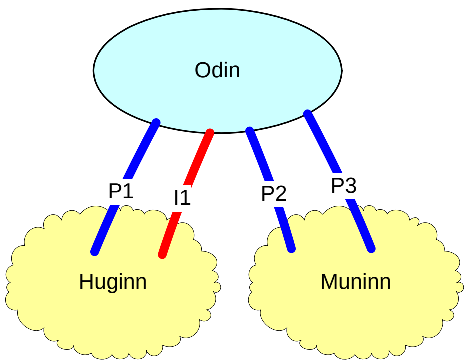

Figure 10: Odin Field

To make the example more complicated, each reservoir produces a different quality and to maximize market value the qualities must be blended. The mixing formula for Odin Blend is two parts Muninn quality and one part Huginn quality. The Odin field will be represented as a fixed platform with storage capacity and the production is loaded on tankers from a buoy.

Operating the Odin Field

The operation of the Odin Field will be described using scenarios, where one scenario describes normal production and the second scenario describes how to deal with unexpected change.

Scenario One: Normal production.

The scenario begins when the planning process receives a long term objective from reservoir management. The objective is to achieve a certain quantity with a defined market quality, combined with drainage policies for each reservoir. The scenario is triggered by change in the reservoir management strategy for the Muninn formation. The new production target is set to 50.000 barrels / day of Odin Blend.

Scenario Two: Pressure loss in Huginn

The scenario begins when the Huginn reservoir experiences a drop in pressure due to some change in the formation. The objective now is to deal with the situation locally while maintaining production quality.

Supporting Odin

The last and final step is to map the conceptual architecture into a specific implementation that supports the Odin field, and from that implementation describe how the system works to resolve our scenarios described earlier. The resulting architecture is depicted in Figure 11.

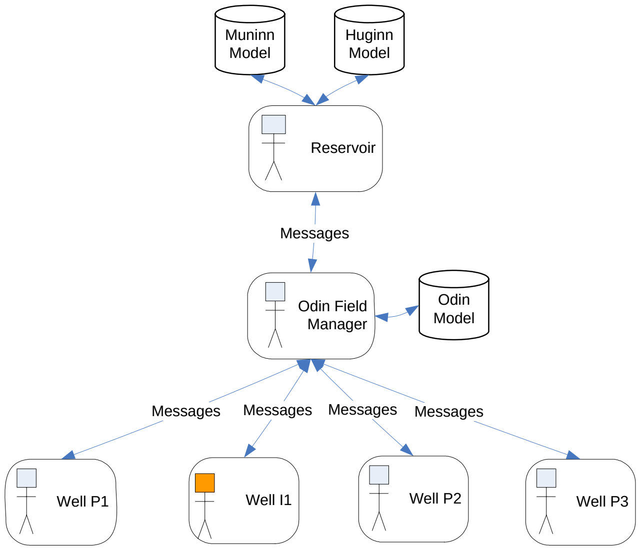

Figure 11: Odin agent architecture

Supporting scenario one: Normal production

Objective: Plan a configuration that meets the production objective of 50.000 barrels a day of Odin Blend.

Algorithm:

1. Field manager receives the request for production target.
2. Field manager runs a GAP model calculating optimal configuration.
3. Field manager request possibility for wells to produce according to optimal configuration.
4. Field manager evaluates responses from the wells and finalizes the configuration.
5. Field manager settles an agreement with each of the wells for a contribution.
6. Field manager conform commitment to reservoir.
7. System reaches a steady state condition where well agents adjust themselves accordingly to satisfy their obligation.

Supporting scenario two: Pressure loss in Huginn

Objective: Re-establish steady state condition, meet production target.

Situation: Increased loss of pressure in Huginn formation.

Algorithm:

1. Well I1 detects a pressure loss and starts to compensate by injecting more water. It starts at autonomy level 5a, and drops over the next couple of hours to level 4a.
2. Well P1 detects pressure loss and judges that it's not able to meet its obligation to the field. Report "unable to meet obligation to field manager".
3. Field manager has two objectives: 1) Produce the required volume. 2) Meet market quality. Objective number two, market quality, is regarded more important than volume, and while well P1 deteriorates due to the pressure loss the production from P2 and P3 is reduced accordingly.
4. Field manager reports to Reservoir that it's not able to meet requested production volume, but can meet quality. A new obligation is established.
5. The root cause of the pressure loss is fixed and I1 reports "pressure normalized" to field manager. Field manager informs Reservoir that it is able to re-gain its original production target.
6. Steady state production is re-established.

Conclusion

The architecture presented in the previous sections shows how an autonomous production support system can enable more efficient production for the reference field. The main contribution of the system is that it aggregates the individual sensor readings from the wells into decisions and perform adjustments in real-time. This enables more efficient production aligned with the actual well and process train conditions.

There are three properties in the suggested architecture that are especially important:

- The negotiation protocol between components at the different layers of the architecture
- The practical effects of variable autonomy and its impact on man-machine interaction
- The flexible deployment capabilities due to the JACK framework and the modular design

The claim is that the negotiation protocol between components in the different layers of the architecture facilitates scaling and reduces the danger for pushing one specific well too hard for a too long period of time. In other words, this will result in a much smoother allocation of wells, which in turn results in improved reliability.

Variable autonomy implies that the human operator can delegate more and more authority for decision making to the machine, as his level of trust in the system increases. Another aspect of this is the filtering that ensures that humans are involved in solving the most difficult problems, leaving the trivial stuff to the machine.

The JACK multi-agent platform allows us to deploy each agent to a separate machine. This means that if the well-completion includes a computer chip, the software agents can be deployed on the actual well-completion. This can open for better interaction with sensors and more efficient analysis of sensor data.

The proposed system architecture reduces the information load to a level more manageable by human operators, and thus contributes to establishing shared situation awareness between the process engineers and production engineers. Designing and implementing a full scale support system in line with the suggested architecture is not a trivial task. For such system to be perceived as a workable solution there is a need for sufficient and robust reservoir/well models, reliable sensor data and efficient human-machine interaction, keeping the human in control.

Further work

As outlined in the conclusion, there are three areas of concern that in particular must be addressed in the further work:

- Efficient human-machine interaction
- More reliable and complete sensor data (data fusion)
- More suitable near-wellbore reservoir (prediction) models

Adoption of autonomous systems depends on the human-machine interaction. In order to establish acceptance for increased computer support it is essential that the human operator feels in control, and is kept in the centre of the control loop. There is a need for more research on how to build efficient human-machine interaction for autonomous production support systems in our industry. Key questions to ask include:

- How to support the user in keeping focus on the asset that requires the most attention
- How to design user interfaces that make the rationale behind a decision / advice visible for the human at a glance

Increased instrumentation of wells is becoming the industry norm, and there is more data available from wells now than ever before. Furthering the development of sensor and communication technology in the well completions is important, but there is also the issue of older fields which will not necessarily be upgraded with new downhole instrumentation. Developing solutions which can tap into similar data for mature fields, and utilizing the local knowledge that has been collected over many years' operation of these fields, will be crucial in introducing similar control systems to mature fields. Key questions to ask include:

- How to provide and utilise state of the art data fusion algorithms for better utilisation of unreliable sensor data

In many cases suitable near-wellbore reservoir models are not available, and there will be a need for more and better models. Key questions to address include:

- How to best combine logical reasoning with mathematical prediction models

It's our hope that these research areas will be addressed in the foreseeable future.

Acknowledgements

The authors want to thank Jan Richard Sagli, Tor Gunnar Aksland, Adolfo Henriquez, Stian Lie, all StatoilHydro, and David Shepherdson and Andrew Lucas, both AOS, for their contributions and support.

References

[1] Effective Operator Engagement with Variable Autonomy, Fiona Cayzer, Peter Wilkinson, Alan Hill, BAE Systems, SEAS DTC Annual Technical Conference, 2007, http://www.seasdtc.com/events/2007_conference/mission_planning_and_decision_making.htm

[2] Architecture for Distributed Power Management for Autonomous Unmanned Vehicles, Andrew Lucas & David Shepherdson, AOS, SEAS DTC Annual Technical Conference, 2007, http://www.seasdtc.com/events/2007_conference/propulsion_power.htm

[3] An Introduction to MultiAgent Systems, Wooldridge, Wiley, 2002, ISBN 0-471-49691-X

[4] JACK, http://www.aosgrp.com

[5] Software agents applied in Oil production, Lise Engmo & Lene Hallen, NTNU, 2007, http://daim.idi.ntnu.no/show.php?type=masteroppgave&id=3368

[6] Space Shuttle Ground Processing with Monitoring Agents, Semmel et al, IEEE Intelligent Systems January/February 2006.

[7] First Flight – True UAV Autonomy At Last, AOS, July 2004, http://www.agent-software.com/shared/resources/pressReleases/Avatar-JACK-F040706USb.pdf

[8] JACK agents in industry, Austrailian Department of Industry, Tourism and Resources, http://www.industry.gov.au/assets/documents/itrinternet/jsfcapabilitiespresentations2007/company%20profiles/agent_oriented/profile_4.html

[9] Software Agents – An Emergent Software Technology That Enables Us to Build More Dynamic, Adaptable, and Robust Systems, Landre & Ølmheim et al, SPE 103354-PP, SPE Annual Technical Conference September 2006.