Thoughts on Software Dependability in O&G

Challenges and opportunities
Challenges
megaproject culture

Langeled pipeline:
- World’s longest subsea pipeline (1200KM)
- 17 Billion NOK (2.2 Billion EUR)
- 4 years development time
- On stream 9 years after discovery

Snøhvit (Snow White) field & LNG Plant:
- Subsurface field produced from land
- 35 Billion NOK (4.5 Billion EUR)
- 7 years development time
- On stream 25 years after discovery

Prestige or failure
big physical machines

Troll A, 472 meters high, the largest man made structure ever moved

Software an alien concept
strong belief in human expertise

Driller is king
but time is precious

Problem severity = function(time)
Failed Safety Critical Decisions
- Situational awareness
- Trustworthiness
- Culture
- Decision quality
the weakest point

Human brain - planets most sophisticated and vulnerable decision maker

- Emotions trumps facts (irrationality)
- Limited processing capacity
- Need to rest, easily bored
- Inconsistency across exemplars
- Creative, easily distracted
- Values (ethics and morale)
- Mental illness (irrationality)

How to avoid clusterfucks?
Opportunities
What is the best action to take?

- I have to make frequent decisions and many of them depend upon readings from sensors that can be correct, noisy, random, unavailable, or in some other state.
- The decisions I have to make often have safety consequences, they certainly have economic consequences, and some are irreversible.
- At any point in time there may be three or four actions I could take based on my sense of what’s happening on the rig.
- I would like better support to determine how trustworthy my readings are, what the possible situations are and the consequences of each action.
systems of action

Computer systems that

• Can sense or observe a phenomena, process or machine

• Process observations and search for anomalies, undesired state changes and other deviations that must be dealt with.

• Plan and execute / (recommend execution of) actions to bring the observed phenomena, process or machine back to its desired operational state.

• Monitor effects of actions and re-plan if action did not have intended effect on process state

making better decisions under stress and uncertainty
drilling - a case study

A manually controlled process

- Manual Control
  - Interpret data
  - Perform tasks

Drilling Control System
- Top Drive
- Hoisting Drive
- Mud Pumps
- Topside Sensors

Real Bore Hole State
add active computer support

Intelligent Drilling Assistant

- Drilling Control System
  - Top Drive
  - Hoisting Drive
- Mud Pups
- Topside Sensors

Real Bore Hole State

Recommend actions in context of process state

Real-time data

Manual Control
the drilling assistant

Intelligent Drilling Assistant
- Drilling Advisor
  - Uncertainty model
  - Causality model
  - Reasoning
  - Plans
- Drilling Simulator
  - Hydraulic model
  - Mechanical model
  - Temperature model

Drilling Control System

Real-Time Data

Actions

Action to be executed by human, but concept opens up for more computer control in the future.

i.e. Drilling advisor can be turned into “synthetic driller”.
What is the best action to take for the business?

What is the best action to take for control or safety?

What is the process state and where is it heading?

What do we know for certain and what are we estimating?

What can we infer about performance and changes in the physical system?

What are we measuring directly, with what accuracy?

Thanks to Dr. Matthew Barry, www.softism.com and Dr. Andrew Lucas AOS www.aosgrp.com for valuable contributions
more sophisticated technology

Global Action Optimization
Local Action Optimization
Situational Awareness
Uncertainty and Validation
Physical System Behavior
Physical System Sensing

Automated planning and scheduling

Machine learning (Bayesian) + Physics (Cyb)

Decision / game theory

Rational agent
• has goals
• models uncertainty
• chooses action with optimal expected outcome for itself
• Examples:
  − human (on a good day)
  − intelligent software agent
new challenges

Industry become software dependent

What parts are safety critical?
What parts are only business critical?
How to assess and protect against cyber threats?
How does failure in non-safety part influence safety and security?
What dependencies do we have?
How to design software that tackles mechanical failures?

Boundary between safety and business critical functions blurred
summary

Increased software dependency in critical functions
Software used in critical control loops, beyond traditional safety systems

Must understand 2\textsuperscript{nd} and 3\textsuperscript{rd} order failure effects
System behaviour is not linear

Software to be used to mitigate human weaknesses
Must be designed to enhance human capabilities

High Integrity System thinking needed
For more software than we traditionally think
Thank you