COMPUTER-AIDED LEAN MANAGEMENT

A TUTORIAL

By Roger Anderson and Albert Boulanger, Columbia University, New York, NY 10027

The purpose of Computer Aided Lean Management (CALM) is to enable operational innovation through the deployment of software algorithms. Lean management is a methodology for efficient enforcement of process rigor and discipline in order to dramatically cut costs and improve operations of an enterprise (see **http://leanenergy.ldeo.columbia.edu/ogj**). This software development will also reduce operating risk, enhance customer service and reliability, and increase the assurance that a new design introduced to the "market" will be effective. CALM is software-controlled lean systems integration that drives breakthroughs in cost and risk reduction. Operational innovation within an energy organization will be enabled through the integrated deployment of three major software systems that we call the Integrated System Model (ISM):

- Product modeling – High resolution model of physical infrastructure.
- Business process modeling – Capturing detailed process and work flow information in order to track and measure performance on a daily basis with a goal of optimizing these processes.
- Machine learning system - Diagnostic analysis of historical and operational data captured in existing data as well as from Product model and Business Process model outputs to predict and/or prioritize required operations and maintenance of an energy company's business.

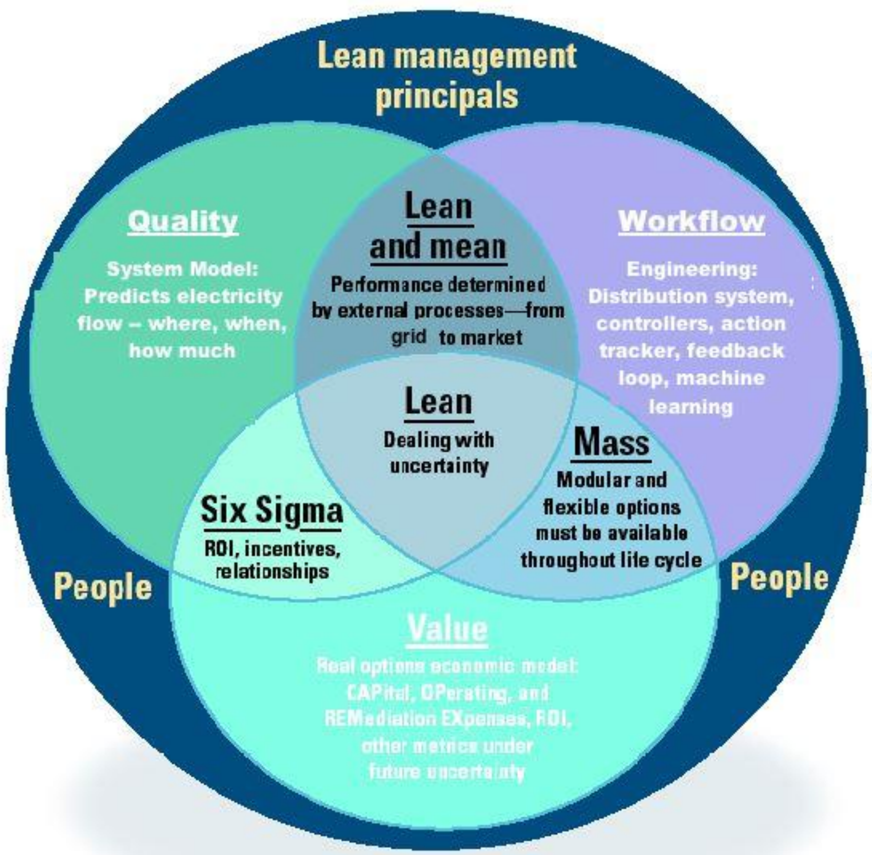
CALM is a methodology for running a business based on the common sense approach of measuring the results of actions taken and using those measurements in an experimental way to design new processes that drive out inefficiencies. In the ISM we will have models of the business where alternatives can be explored to find the innovations required to improve the company's performance.

The ISM will provide the tools needed to "see" the competitive landscape or environment the company operates in. Some the feedback to improve performance will be provided by the machine learning tools being developed in this project. The company will need these tools and will need to adopt CALM in order to become more adaptive and therefore better able to perform successfully in the future as the "business we are in" changes.

The adoption of CALM is expected to provide the enterprise with a significant competitive advantage in its existing core business of electric and gas utilities. This competitive advantage may be used in the future to enable mergers and acquisitions where significant synergies are achievable in both SG&A and operational departments (synergy savings traditionally unachievable) through deployment of this platform. In addition, the development of a professional services firm that can support the deployment and on going decision making for operations and maintenance support for other companies is a strategy of business growth that will be analyzed at a future date.

Chapter 1: What is Computer-Aided Lean Management

Computer-Aided Lean Management (CALM) is software-controlled Lean Systems Integration that drives innovation toward breakthrough cost and cycle-time savings. This methodology is used throughout the communications, chemical processing, aerospace, automotive, and other high-tech manufacturing and processing industries, but is not yet widely used in the energy Industry. This tutorial sets out the background, logic, and processes and systems engineering changes needed for a company to migrate to Lean Management. There are extensive websites for Lean accessible through Google, but two of the most relevant are maintained by the Lean Aerospace Institute at MIT (http://lean.mit.edu) and by us at Columbia University's Lamont-Doherty Earth Observatory (http://leanenergy.ldeo.columbia.edu).



Lean Management Background

Lean management is a methodology for efficient enforcement of process rigor and discipline in order to dramatically cut costs and improve cycle times of all operations of

an enterprise. It is an outgrowth of IDEF (Integrated Definition) modeling in aerospace manufacture that was pioneered by the US Air Force in the 1970's.
IDEF is a methodology designed to model the end-to-end decisions, actions, and activities of an organization or system so that costs, performance, and cycle-times can be streamlined and optimized. IDEF methods have been adapted for wider use in automotive, aerospace, military, pharmaceuticals, and even in the software development industries. Toyota is now the most famous practitioner, with their Lexus divisions "Continuous Pursuit of Perfection" its most prominent call-sign.

There are now 16 subsystems for IDEF that model function, information, and data flow and simulate processes, design, ontology, improvement, systems architecture, and the organization's networks. There is even an auditing IDEF. As an example of the process, IDEF methods are used to model the functions of an enterprise, creating a graphical model, or roadmap, that shows what controls each important function, who performs it, what resources are required for carrying it out, what it produces, how much it costs, and what relationships it has to other functions of the organization.

Computer-based IDEF simulation of the enterprise has been found to be efficient at streamlining and modernizing both companies and governmental agencies. IDEF methods are maintained by the National Bureau of Standards, through the Knowledge-Based Systems company (see http: //www.idef.com).

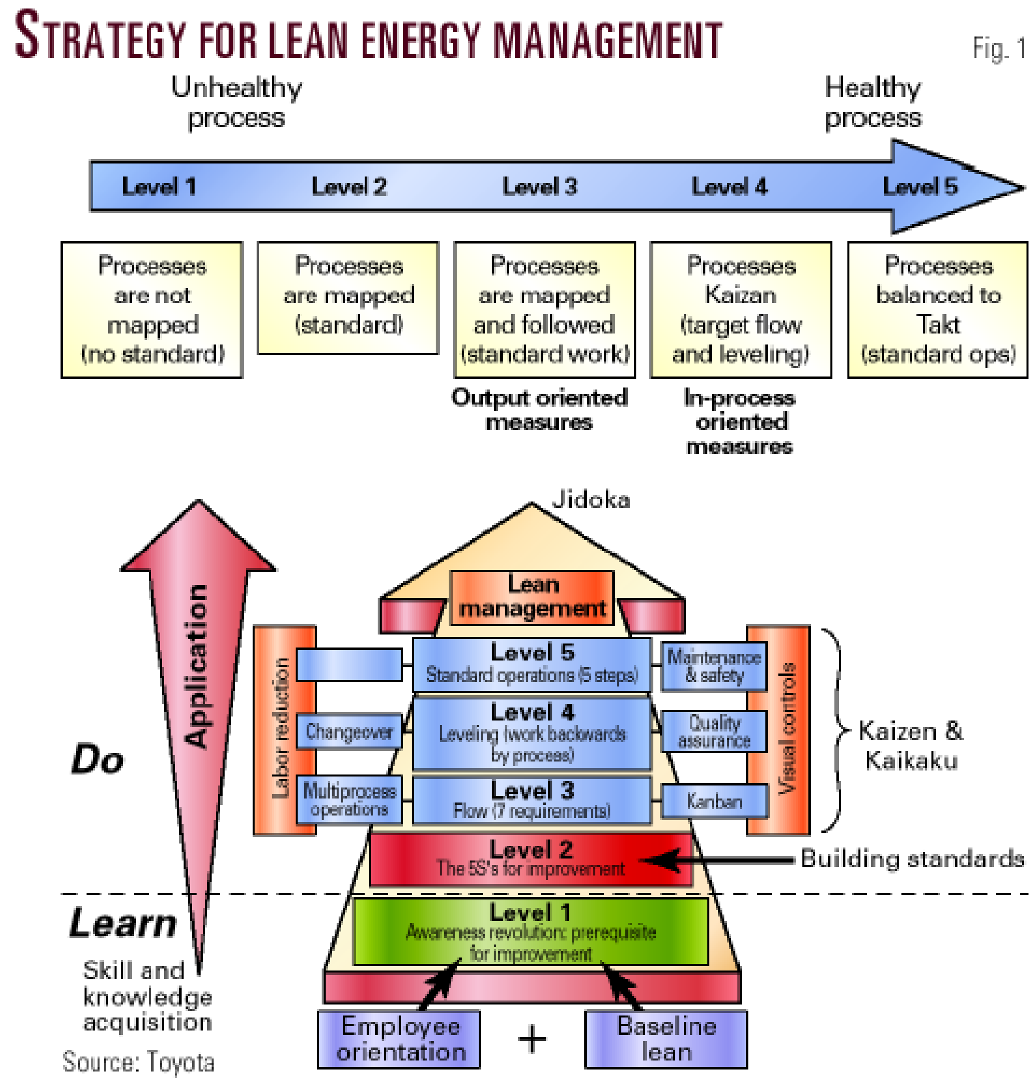
GE

First Motorola, then famously GE, developed Six Sigma principals of lean management that can be traced to roots in IDEF modeling. At GE, Six Sigma has grown into more general "lean engineering" principles that are rigorously enforced throughout the organization. Software is used to make the entire manufacturing system transparent and measurable, whether it's a light bulb, electric generator, or jet engine factory. GE requires process mapping of the "as is" condition of whatever system is to be improved, establishment of baseline metrics, identification of where the waste is occurring, planning of the improved "to be" process—all on the computer before change is authorized.

Then, software controls the implementation of the innovation plan, with constant reviews of performance metrics along the way. As good as its technologies are, GE does not differentiate through innovation itself so much as through execution of the systems integration processes required to manage innovation, whether it is new product or the manufacture of old, reliable light bulbs.

Toyota

The most famous IDEF derivative is the Lean Automotive engineering model at Toyota. The Japanese derivatives Kaizen, Kaikaku, and Judoka can, in turn, be found at the base of process improvement methodologies of most other major innovation companies, both inside and outside the automotive industry today. The Toyota process management methodology is "bottom-up," consisting of "learning steps" of skill and knowledge acquisition, followed by "standards building" so that metrics of improvement can be mapped out. Only then does the "do action" start (sound like Six Sigma?).



In particular, significant improvements in lean engineering were made by Toyota in dealing with its subcontractors. Toyota realized that everything from just-in-time inventory delivery to total-quality-management, to adoption of new innovations, all depend as much on the performance of its outside suppliers as with itself.

At Toyota, corporate investments in new technologies include not only acquisition and venture investments, but also loans, and sometimes, outright gifts to suppliers to get them to buy into the Toyota lean management system. Why go to all that extra expense? Metrics and performance standards can then be tracked all day, every day throughout the "greater organizational system."

Toyota begins with a road mapping of existing processes so that a plan for migration from "unhealthy to healthy" processes can be planned (Fig. 1, top). Boeing, GE, Lockheed-Martin, and countless other great corporations throughout the world have adopted and modified Toyota's model. In fact, if GM, Ford, and Chrysler can successfully convert to lean then surely the energy industry can, too. The level of software rigor increases steadily up this improvement ladder. Toyota has described how to manage the learning of lean processes with its Jidoka strategy that defines five levels of growth:

Level 1 requires the benchmarking of the existing manufacturing processes, whatever they are, to create a baseline to measure future progress. In addition, employees and subcontractors are all introduced to lean process theory. They are challenged to stop measuring specific actions and instead think of each and every process in terms of the whole system they are producing. It is only the cumulative results of all those actions that results in a quality product.

In Level 2, powerful software tools are put into place to build and enforce standards and identify and eliminate waste in materiel, machines, effort, and methods.

Level 3 tools escalate to the introduction of a common, 3D solid model for all to use to standardize and streamline work.

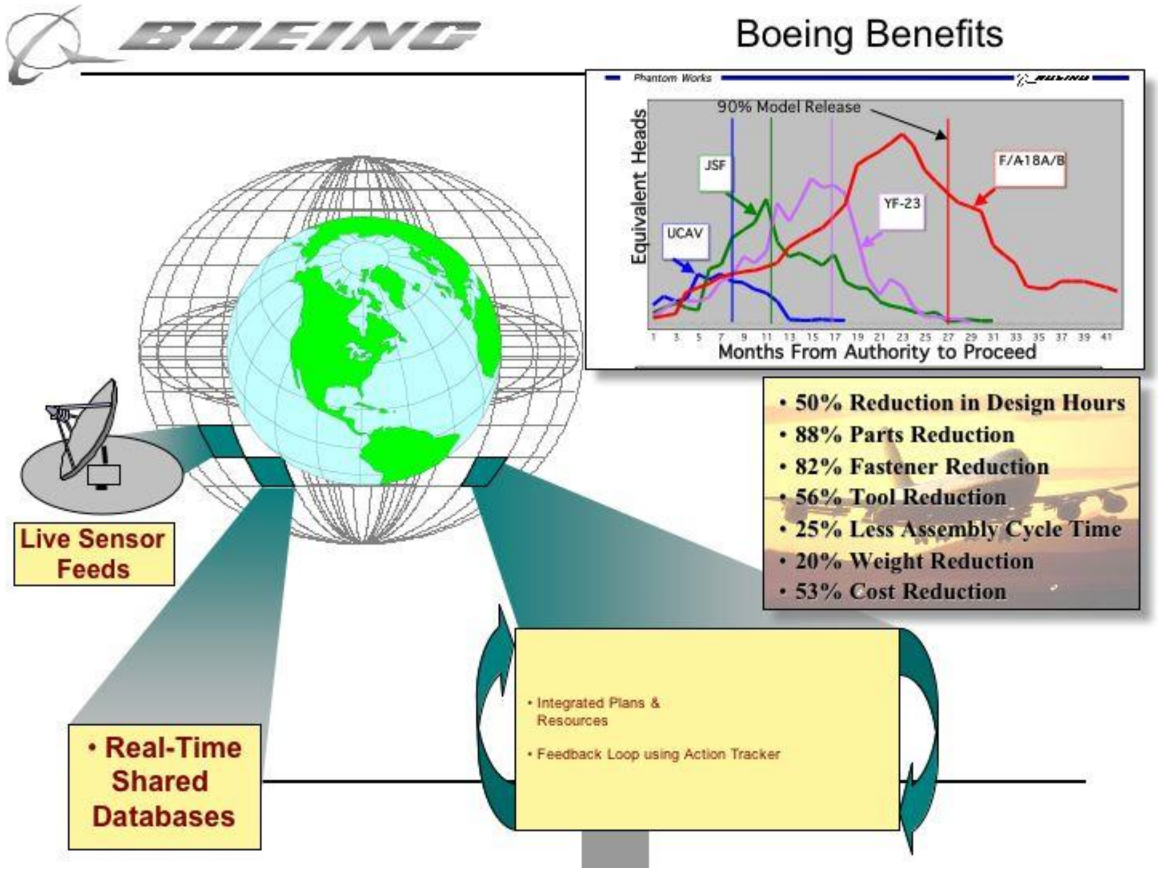
Level 4 introduces a continuous improvement plan.

Level 5 finally achieves lean management.

Boeing put Computer Aided into CALM

Perhaps the best developed evolution of the IDEF model is at Boeing. Their project life cycle process has grown into a rigorous software system that links people, tasks, tools, materiel, and environmental impact of any newly planned project before building begins. Routinely, more than half of the time for any given project is spent building the precedence diagrams, 3D process maps, integrating with outside suppliers, and designing the implementation plan, all on the computer. Once real activity is initiated, an "action tracker" is used to monitor inputs and outputs versus the schedule and to deliver metrics in real-time throughout the organization. When the execution of a new airplane design begins, it is so well organized that it consistently cuts both costs and build-time in half for each successive generation of airframe.1 And, of course, it is paperless. Boeing has found that these cost and cycle-time savings can even be accomplished on one-of-a-kind production projects such as the X-32, X-45, and X-50 experimental air vehicle.

At Boeing, for example, the ratio of computer-aided design to assembly line manufacturing time went from 1:5 before their experience with development of the 777 aircraft to 5:1 after using Lean processes for building that aircraft. They are 4 generations beyond that plane now, and they have succeeded in cutting the time AND cost by 50% for EACH new generation of plane using CALM.



Boeing success using lean management tools and success

Table 1

- **30-50% time savings on modular redesigns**
  - Digital parts library allows design optimization
- **30-50% time savings on initial layouts**
  - Library eliminates the need to search manually
  - Digital model exists for each standard module ("Lego Blocks")
- **Quality improved**
  - More time spent designing than building
  - Less time searching for parts
  - Commonality of parts between designers
  - Both design and instance reduction in part counts
- **Accurate build out model created in seconds, anytime, anywhere**
  - Parts list generator creates up-to-date procurement list

Why CALM for the Energy Industry

There are, believe-it-or-not, many similarities between the automobile, computer, and aerospace experiences and those in the energy industry. All require large-scale systems integration of complex engineering processes. They involve multiple suppliers that are global, and there are many common suppliers to each owner. Lean management provides for earlier engagement of integrated development teams, improved access to new technologies, earlier rigor to "go/no-go" decisions, and an enhanced resource base and skill level of its managers and engineers. In this time of the "graying-of-the-industry," a lean energy revolution would also open up an entirely new employment pool: aerospace, computer and automotive engineers and managers. These lean thinkers would bring a fresh look to the elimination of customization, complexity, and interface conflicts.

Lean Energy Management would lower costs through more commonality in supply, compress cycle times, produce projects that are all on time and to budget, allow no "train wrecks," drive innovation matched to need, and improve first year operability to 90% or better—all these improvements have been driven by the lean aerospace and automotive models, and can be expected if they can be adapted successfully into a "lean energy " management model for the energy industry.

Lean Energy Management a frontier much like landing on the moon was in the 1960’s. It is the equivalent of a “moonshot” for the energy industry. The challenge is twofold: Deal with the risks and uncertainties of the technological frontier, while at the same time reducing the costs and cycle time required to exploit the resources by half or more. Lean management processes and tools have created fundamental improvements elsewhere only because the "behavior" of the entire system was fundamentally changed.

As with automotive and aerospace workers before, the energy industry will encounter a predictable set of reactions to these lean energy concepts, processes and tools:

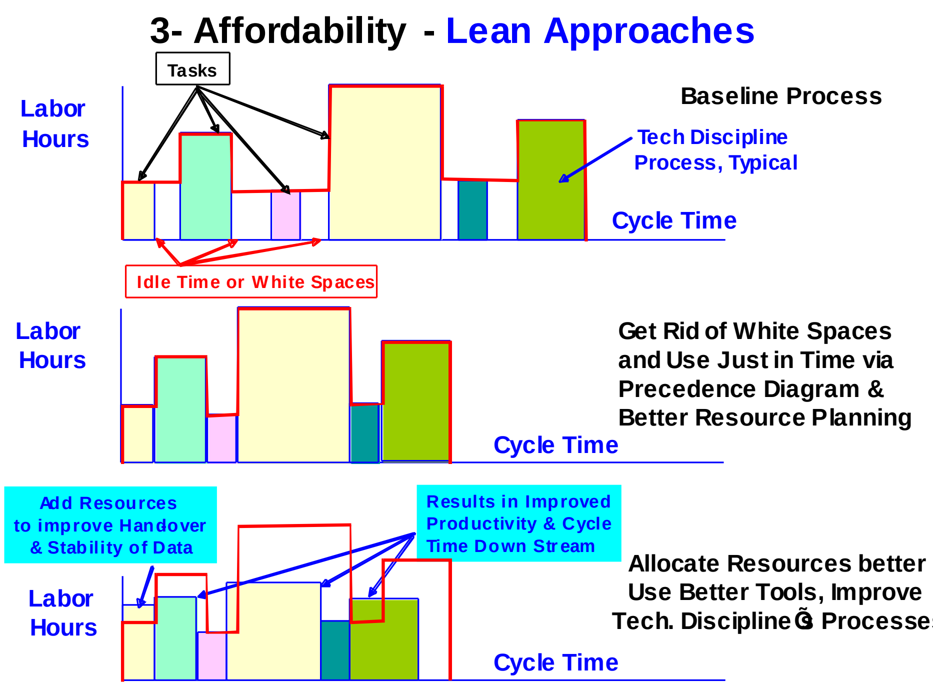
1. You don't understand our business; ours is harder; the offshore is different.
2. We don't need lean; we just need to be quicker and cheaper in what we do now.
3. We are already using lean; we're doing that, and that, and that U as each new level is introduced.
4. But only design/build will benefit from lean processes, not HR, not finance, and certainly not operations.

Lean energy management will take any company several years to fully implement, and the above reactions must be worked through. Examples of success become critical teaching tools to overcome the considerations of the risks involved in conversion. Honest awareness of previous "train wrecks" and a realization that technologies alone will not produce the step change improvements promised by lean management are two human barriers to overcome. Fundamentally, lean is a people-process, and "soft side" change is hard to achieve.

Chapter 2: What is Computer-Aided Lean Energy Management

Lean energy management is a methodology and software tool set that seeks to integrate all design processes into one openly shared, enterprise-wide model of the operations and processes of the company. CALM addresses planning, construction, installation, and maintenance, all together inside the same integrated systems model. However, the uncertainties unique to the industry's economic evaluation, appraisal, fabrication, installation, and design, will have to be integrated into the CALM if we are to succeed in transferring lean efficiencies to the energy industry. Such a paradigm shift to CALM should immediately affect positively on cash flow and profitability.

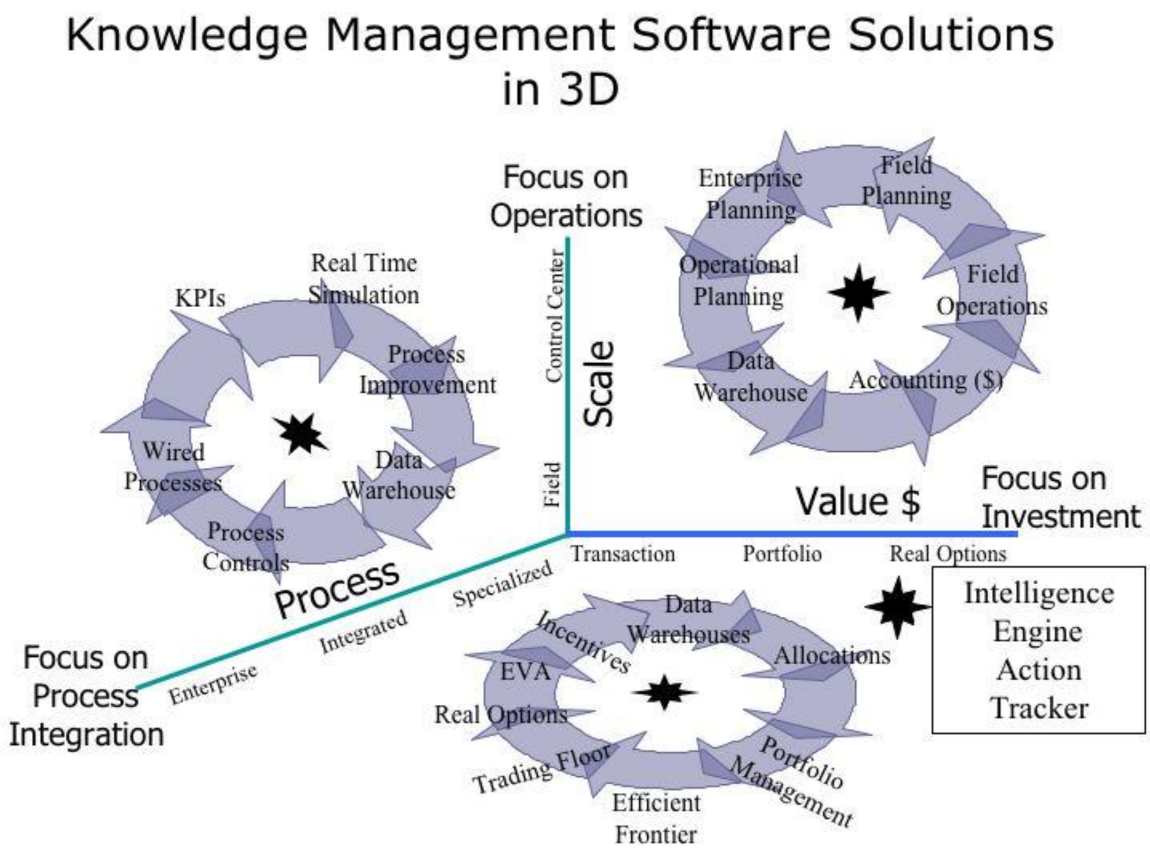
The initial objective of CALM will be to exploit the potential for significant improvements in cost and cycle time savings by addressing the integration of Operations and maintenance practices to eliminate what are called "White Spaces" – idle times in the workflow process when detection is missed, parts are not delivered on time, assets are not efficiently deployed, and silo's of responsibility are not effectively crossed.



CALM tightly integrates:

1. Lean tools for appraisal, planning, construction, operations, all the way to abandonment, with
2. Lean processes in the organization for planning, scheduling, supervisory control, regulation, management, security, and environmental impact.

Included are major cost reductions and operability gains from enhanced project visibility and transparency, increased innovation, and the sharing of knowledge and understanding throughout the enterprise. These gains can come only IF the vast majority of the people of the enterprise adopt lean management practices and break from the past to make it happen.



The implementation of CALM requires first and foremost, a better understanding of how lean processes and tools are able to perform more accurate design studies, account for uncertainty, and eliminate waste in the fabrication of offshore facilities.

The six major systems components of CALM are:

1. Product life cycle management, combined with
2. A systems engineering approach to assist in the selection of the best concept to go to the select phase, which leads to
3. Feature-based design, which is then used for parametric modeling, morphing, and standardization driven by digital component libraries, that in turn facilitate
4. An integrated analysis process
5. Virtual model simulations that visualize the environment, workflow, and generate electronic work-instructions, and finally
6. Preventive maintenance and supportability plans that are developed on the computer before any actual Operations and Maintenance (O&M) changes begin.

CALM shortens time and cost by paradoxically maximizing the digital design time and preserving options, should later understanding of uncertainty yield a revised set of operational parameters as the project matures. This dichotomy of shortening cycle time while providing more decision time is a hallmark of the Lean Management processes.

The six CALM processes eliminate waste and miscommunication, while assuring seamless information flow downward and amongst the owners and the contractors involved in any job. **However, this change requires major redefinition of Information Technology departments, in particular. No longer is "ownership" of data or computer processes confined to the IT department, CALM instead distributes it outward to the locations of the workflow itself.**

In another CALM paradigm shift that affects the IT department, integration of subsystems is enforced by software. When one engineer changes a design specification, it propagates instantly to all other users of the CALM software system. Earlier supervisor involvement, changed needs for inspections, and many fewer parts, outages, and particularly rework orders, are the result. Cost and schedule controls are transparent and available for all to see. A key to the success of CALM is the simulation of requirements and appraisal of needs using real options for quantifying cost versus added value of each decision along the critical path. Analyzing costs and defining the structures to be built or repaired, then planning the execution of the chosen designs, testing operations and evaluating the options required by customer and market uncertainties–all are done on the computer before the first piece of wire is laid or pulled.

The reason for real options as a preferred choice over net present value in the CALM decision-making economic model is that real options require flexible operating principles to be applied to the "factory" whatever it is. Real options are the modern way to quantitatively evaluate the costs and benefits of this flexibility.

Another primary weapon of CALM is modeling. Only if the system can be adequately simulated can we fully evaluate consequences before we take actions. Battlefield terms such as situational awareness, global visibility of assets and inventories, distribution and transportation options, and optimization of logistical supply chains, then become important to the energy industry.

Common actuators for CALM are single-source-of-product, standardization-across-platforms, parts-grouped-into-assembly kits, and above all else, large-scale systems integration. Resulting improvements are a digital library that allows attributes and geometries of all parts to be matched to create standard configurations and suppliers. Reduced design, tooling, and manufacturing instructions, increased procurement & manufacturing lot sizes, and efficiencies that minimize artificial shortages and surpluses in manufacturing and support follow (just-in-time). Below, we examine the six processes of CALM more directly and discuss the tools and processes required to enact each in the energy industry.

Project Life Cycle Management

Project life cycle management (PLCM) seeks to link all data and processes across the appraise, select, define, execute and operate stages of all projects. The result is faster, quicker, and better performance through enforced discipline, transparency and continuity of design and requirements over the full life cycle of each project. PLCM consists of a set of software tools that enforce this process rigor across the stakeholders all day, every day, throughout the life of each project.

ELEMENTS OF PROJECT LIFE CYCLE MANAGEMENT

Table 2

Key processes driven by the PLCM software tools are:

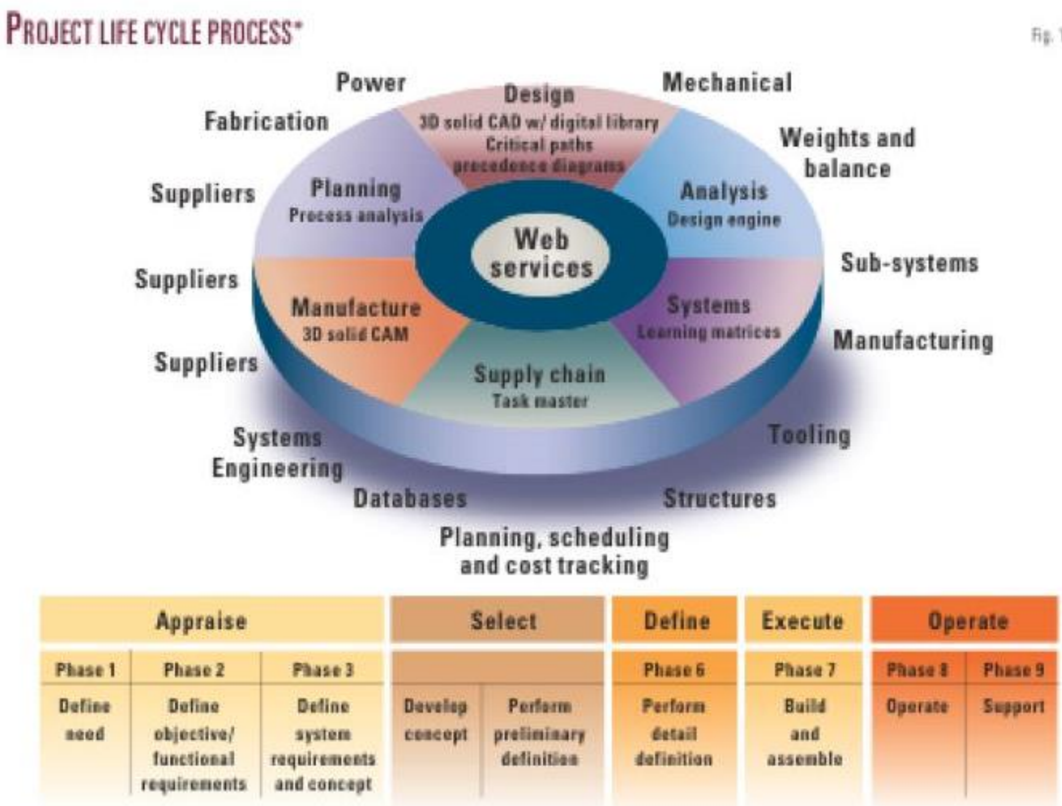
–Applies simulation techniques to predict system behavior
–Uses best practices and lessons learned to improve future performance
–Uses a paperless, totally digital, design anywhere, build anywhere concept
–Identifies specific owner-supplier relationships, including incentives
–Performs earliest possible studies of alternative responses to uncertain outcomes

PLCM improves capital efficiency by:

–Shortening time to "go/no go" decisions
–Allowing fewer changes which affect costs and schedule
–More effectively using internal and external suppliers
–Improving understanding of asset management
–Delivering performance metrics 24/7

For example, everything works off the same model of the entire system being operated. Automatic data updates assure linkages between analysis, simulations, and ultimate designs, shortening the definition and testing cycle. "Smart software" flags interferences among bulkheads, electrical wiring, and piping. A digital parts library is integrated into the conceptual design process to limit the number of components and increase commonality among projects. Requirements and objectives are clearly recorded for all to see.

Risk management tools are used throughout to identify, assess, categorize, and strategize through a formalized risk mitigation process. An integrated planning process creates precedence diagrams for all subsystems and then computes a resource load schedule for critical paths. It tracks actions and provides daily progress metrics to all indicating actual costs and time versus plan for suppliers as well as owners. Assembly is planned on the computer, and a process map is created from start to finish of the project and signed-off by all before physical construction begins. Budgets are prepared from this bottom-up process map. All this is done with commercial, off the shelf software (COTS) to assure availability throughout the chain of responsibility, including all subcontractors. Activity pert charts and electronic work instructions are then issued, and task load schedules are calculated so the project can be staffed. PLCM provides a single source for all procurements and bills-of-materiel.



Modularity is built into the PLCM system through software enforced distribution of the operating model. Electronic schematics, routing and installation layouts, subsystems analyses, and virtual prototyping are all shared with relevant suppliers by the software so that version mismatches, modifications that are incompatible with each other, and module conflicts are identified in the model long before any construction or repair occurs.

Lean Systems Engineering

In aerospace, avionics engineers have traditionally used systems engineering to do their work, whereas vehicle and hardware engineers have designed components first and evaluated their performance later. The latter methodology is closer to that used by electrical engineers. Lean energy management uses the systems engineering approach throughout the enterprise, a process that has proved far more efficient from both a cost-saving and time-saving standpoint. It requires that all significant owner and contractor requirements are thoroughly understood before a specific solution is developed to address these requirements. All disciplines (structural, manufacturing, quality, security, support, etc.) participate in the CALM systems engineering analysis.

Effects of modifications to one system can then be analyzed for adverse effects to other systems. Conflicting issues always emerge among fabrication, weight, cost, accessibility, maintenance, safety, and schedule. As these conflicts are resolved on the computer, the form, cost, and performance of the likely optimal system design or modification slowly emerges. Lean systems engineering data flow starts with the customer needs, objectives, and requirements described in terms of real options. This leads to requirements analysis and carpet plots that evaluate the uncertainty window. A large scale, full system scenario is then simulated that evaluates the responsiveness of the design to variances caused by uncertainty, such as price volatility or unusual component wear and tear. Real options are used to identify and assess the relative value of solutions to the uncertainties in the simulation.

The key is to design-in flexibility in both the surface and subsurface "factory floor" and to use real options in the decision procedure to enforce quantitative rigor. Gap analyses are conducted to further refine the concept and add lessons learned from the best practices database. Operational, safety and environmental issues are then examined. Performance metrics are designed to monitor implementation. Metrics that track owner/contractor relationships and cross-organizational performance are particularly important to define, and efficiencies can then be proved. All this is done on the computer before any construction or repair begins.

Feature-Based Analysis

The emergence of design conflicts leads to more and more in-depth computational analysis, or feature based design and integrated systems analysis. These analyses focus in very specific areas, addressing unique issues that arise through simulations of variability. The critical conflicts usually cross disciplinary boundaries and involve three or more processes that in the "old way" would seldom have discussed alternatives with each other until after the need for painful reworking became obvious.

As the re-configuration matures, many of the most difficult issues re-appear. Electronic signoff then freezes features one after another until the "preferred design" emerges. A configuration baseline emerges in the feature based design process, whereby conceptual assembly layouts and "build to" packages are designed using the digital parts library. These, in turn, automatically define supportability requirements. Integrated systems analysis is then used to test functional capabilities of the new assemblies, their ties to central control, and multiple subsystem integration capabilities. Functional electrical schematics and logic diagrams are then automatically generated for each subsystem. A virtual prototype of the new configuration emerges, ready for the supportability simulation.

Virtual Supportability

Virtual supportability tools then simulate the "factory floor" or field operations to layout the most efficient construction or repair sequence for each project. Virtual supportability takes model scenarios and adds operational and maintenance evaluations. It incorporates simulations of complicated maintenance tasks, including placing human sized repair crews into a virtual reality environment to make sure tolerances will allow maintenance access. The supply chain is laid out along with procurement sequences. Historical requirements, forecasts, and consolidation of demand (turn rates) are used to determine appropriate sourcing and just-in-time delivery schedules. Virtual support begins the operator training process using operational and maintenance tasks even as the reconfigurations are being constructed.

Evaluation of Cost-Cycle Time Gains

How can the energy industry effectively evaluate the risks associated with conversion to CALM processes and tools? Until there is a substantial track record of CALM improvement, cost and cycle time impact can be estimated by considering the following steps applied to all the process subsystems:

- Use previous project experience of your personnel to describe the as is state of project subsystem tasks.
- Define savings based on other lean industry performances at similar subsystem tasks.
- Add uncertainty factors for industry differences (may initially be as high as 50%, but will drop over time.
- Add in worst case delays and confusion from first-time use of CALM tools and processes.

Contrast the economic value of these risks against estimates from your own personnel of the positive value of:

- Reduced instruction ambiguity.
- Faster reviews of options.
- Earlier scheduling decisions.
- Better supplier and fabricator collaboration.
- Improved visibility of the total project.
- Improved change-order management.
- Reduced site queries.
- Better health, safety, and environmental impact assessments.
- Improved interfaces between contractors for hookup and commissioning.
- Supply chain improvement.

The major difference between CALM and what the energy industry uses today is that the tools and processes for each task are not currently integrated to implement a system-wide model. In today's xml and web-services world of COTS software, tools can be integrated to produce the required seamless digital environment. The value of that transformation should become evident as the checklist described above is worked through. Once installed, the CALM system must then be kept open to best-in-breed improvement solutions. Most importantly, people must re-invent their workplaces. All stakeholders must work together before the required cost and cycle time savings from lean energy management can be fully realized.

Chapter 3: Implementation of CALM

In order to implement Lean Energy methodologies, process and economic models must be constructed that integrate horizontal stages of development from appraisal, through planning, construction, and operations all the way to abandonment, with vertical levels in the organization from planning, through scheduling, supervisory control, regulation, reservoir management, engineering decision making and environmental impact. Industries such as automotive, aerospace, pharmaceuticals and the military have cut CAPEX and OPEX costs and cycle times up to 25% **yearly** through Lean practices (Anderson, et al, 2001, Anderson and Esser, 2000, Saputelli, et al, 2000). Such a paradigm shift to Lean Energy systems will immediately affect cash flow and profitability of energy developments to the same degree as in these other industries.

Pain that Lean Energy Management and Economic Models address:

1. Eliminates the "Wish I could have seen it coming"
2. Failure Models become predictive
3. Expedites cost out analyses
4. Estimates risk & return on all investments
5. Identifies solutions quickly
6. Selects only the most efficient processes
7. Verifies that work is being done on schedule and on cost
8. Eliminates latency in getting the right data to and from the right people
9. Improves the whole system performance for its whole life-cycle

Lean Energy Management requires significantly more integrated software tools than are currently utilized by the petroleum industry to assure that all gaps are filled and connectivity maintained among a large number of contracted principals, as far-field industries have discovered over the last 20 years. As we have seen, Lean processes have spread throughout not just the aerospace (Boeing and Lockheed Martin) and automotive (Ford and Toyota) industries, but also in computers (IBM, Dell) and general manufacturing (GE, and United Technologies).

Most "energy factories," such as refineries and petrochemical plants, nuclear and electric power generation facilities, and energy transmission and distribution systems have processes that are centrally controlled by operators monitoring supervisory control & data acquisition (SCADA) sensors distributed throughout the field operations. The National Aeronautics & Space Administration's Mission Control exemplifies this type of control center methodology that uses computers to keep operators up to date with measurements that monitor the real-time conditions of the system, whatever it is. Next generation, the central control centers are migrating to the field operations themselves. Two-way communications is required in real-time. Not only is information on the state of the field operations now being delivered to all users in near real-time, but support has begun to be distributed to process control decisions being made at any given time and condition at each remote site.

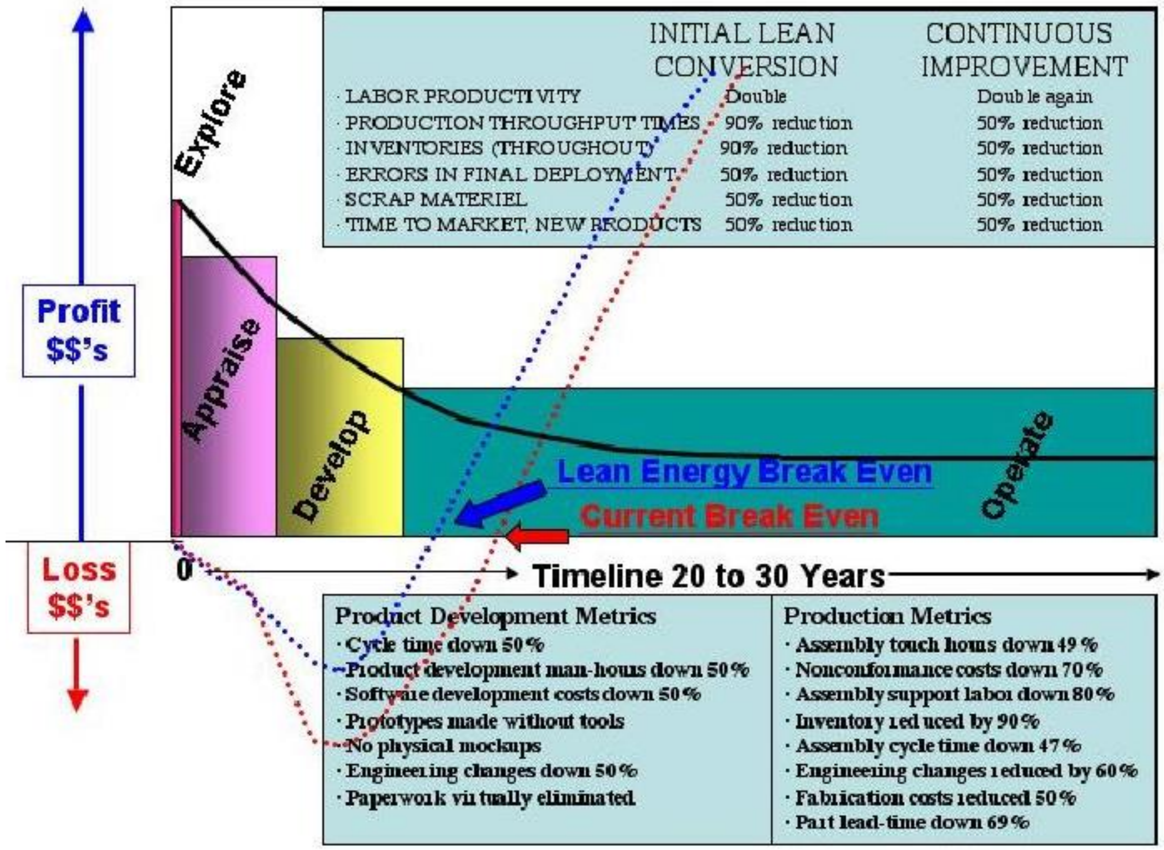


Figure. Predicted impact of Lean Energy methodologies based on aerospace results.

A general term for this progression in computer support is, from "information" to "knowledge" management. Optimization of the enterprise is being augmented with "learning systems" that "close the loop" to empower control center personnel to take actions to modify processes based upon the lessons learned from analysis of past-performance and real-time market conditions.

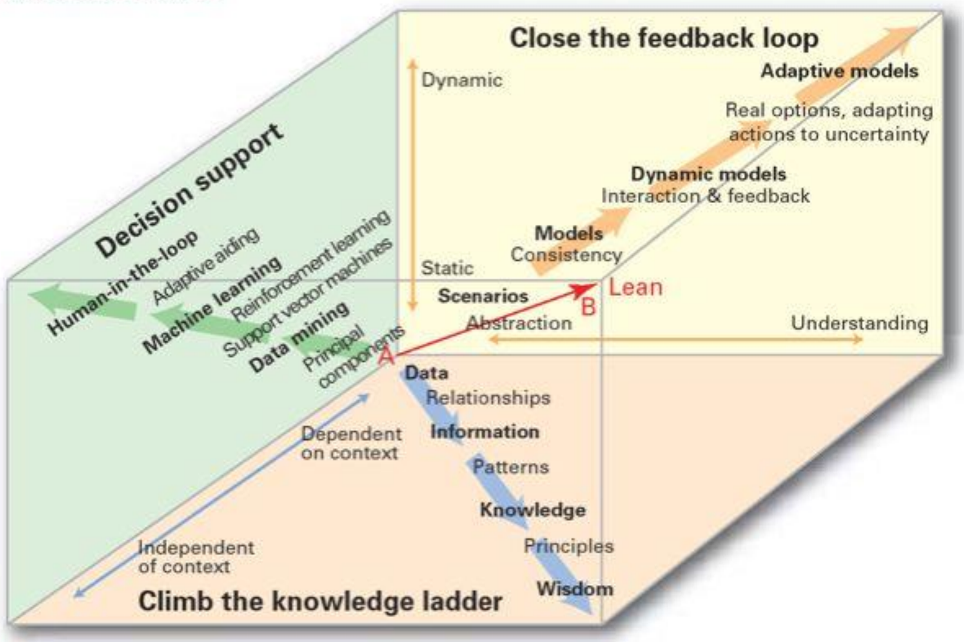
Modern Knowledge Management in CALM

This evolution in knowledge management can be illustrated in terms of common games. The TV show "Jeopardy" can be won consistently if information is managed efficiently. A laptop computer these days can win every game. All that is required is a good encyclopedia in its information database and a "Google-like" rapid data mining capability. Backgammon is a step up from "Jeopardy" in complexity. A computer again can be trained to win every game, but it must be programmed to deal with the uncertainties introduced by the throw of the dice. Moves must be re-computed at each move based on a changing board position. Chess is another matter, however. Strategies, end-games, pattern recognition, and tactics must all be mastered, in addition to managing information and uncertainty. A few years ago, IBM tied for the World Chess

Championship, but it required a dedicated supercomputer, "Big Blue," to sort through all possible future moves as the game progressed. Last year, an Israeli minicomputer beat the human chess champion of the world. The machine used machine learning (ML) technologies to recognize patterns, evaluate likelihoods of success of move sequences, and optimize board positions and strengths. It did not compute every possible move for every possible contingency to the end of every possible game, as Big Blue had.

This illustrates the progression first from information management to knowledge management and now to what is called "machine learning." ML promises to further revolutionize process control in all of the diverse industries mentioned above, from aerospace and the military, to refinery and petrochemical plant control, to the pharmaceutical and automotive assembly. How does the energy industry adapt its decision support systems to this new knowledge management paradigm? Progress in three dimensions is required. We must simultaneously climb the knowledge ladder, close the feedback loop, and add ML to our enterprise management systems.

KNOWLEDGE MANAGEMENT IN 3D



The knowledge ladder

The best known plane of the cube for improving process control is the "knowledge ladder." Migrating from data management to information management allows relationships among data to be defined. Adding pattern recognition converts the information to more actionable information about how the system works. But this is still not knowledge.

Knowledge requires people. There have been many failures from not recognizing this human requirement to get from information to knowledge. That said, there is a role for knowledge management tools and best-practices capture at this level, but it must be recognized and is a core concept of the lean approach, that knowledge is fundamentally in the minds of people.

Closing the feedback loop

In order to attain the conversion of data into wisdom, computer models must evolve from just providing possible operational scenarios to being capable of modeling whole system performance. Consistency and linkages among the many "silos" of management responsibility must be modeled first before they can be understood and implemented.

Then models must become dynamic rather than static. That is, time-lapse information must be used to continually update the control model with system interactions and feedback.

Finally, adaptive models have been developed that execute the process based on continually evolving evaluation of performance. The feedback loop must take incoming data, translate it into actions, evaluate the effectiveness of those actions, and then modify future actions based upon a likelihood of success.

Machine Learning

Perhaps the least understood of the knowledge planes is ML. The natural progression is from understanding data, to modeling the enterprise, to the addition of new ML technologies that learn from successes and failures so that continuous improvement is not only possible but is the operational dictum. We know the energy industry has enough data and modeling capabilities to evolve to this new lean and efficient frontier.

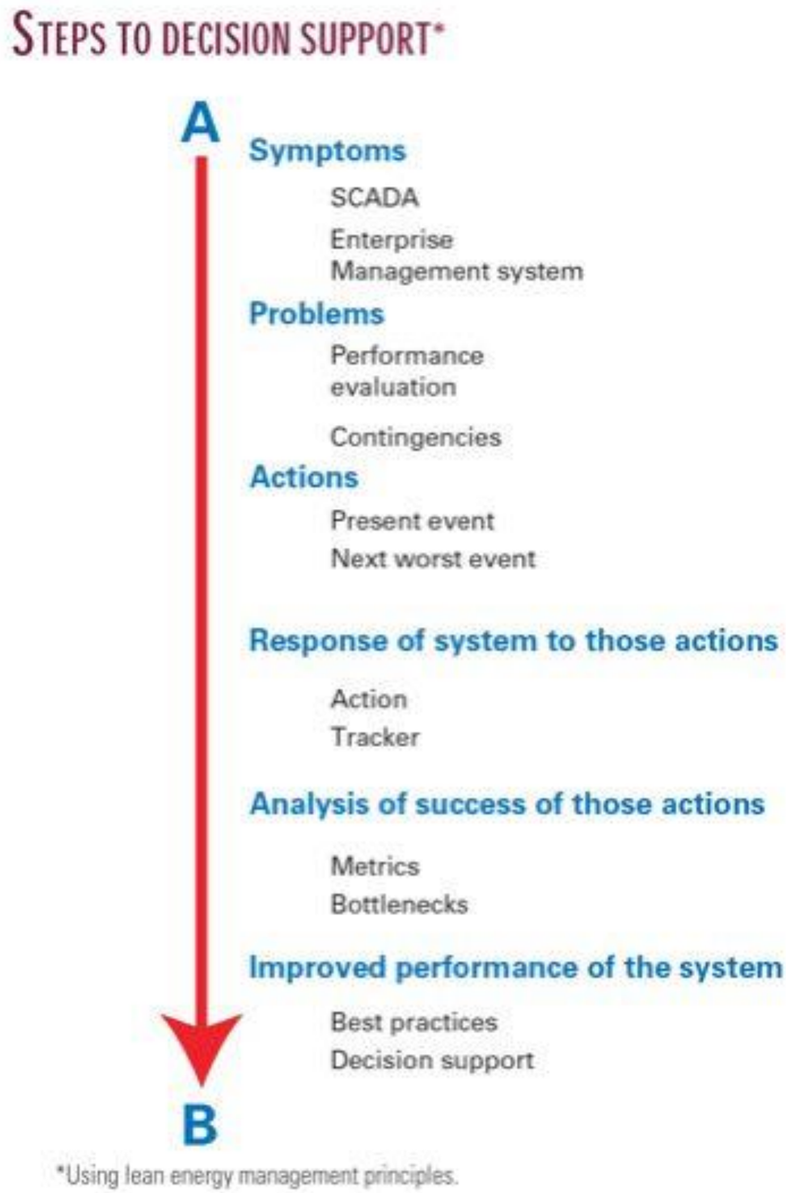
Whenever an event happens, such as the Aug. 14, 2003, Northeast US blackout or a well blows out, we set up a study team that quickly reviews all incoming field data, models the system response, identifies exactly what went wrong where and when, and develops a set of actions and policy changes to prevent the event from happening again.

Progress up the ML plane requires that this data analysis, modeling, and performance evaluation be done all day every day. Then the system can be empowered to continuously improve itself before such catastrophic events happen to the system.

ML uses computer algorithms that improve actions and policies of the system automatically through learned experiences. Applications range from data mining programs that discover general rules in large data sets such as principal component analysis to information filtering systems that automatically learn users' interests.

Importance of CALM for field oriented industries

This revolution is just beginning to be spread into the exploration and development of oil and gas fields because it is difficult to execute lean energy management at remote locations where "unplanned events" are the ordinary. Lean processes require extensive planning and simulation by integrated teams. Five times more work must be done on the computer to simulate possible outcomes as is done on the physical work in the field.



It is easy for teams that are not fully integrated to fall into the "silo trap" by allocating support tasks to subgroups of narrowly focused experts. The tendency is to isolate the exploration of "what if" contingencies into those teams that are most comfortable with them. Only by supporting lean planning with vigorous software checks and balances that require integrated inputs across all teams can this trap be avoided. Anyone who has worked a very large development project offshore can probably recognize that cost and time overruns always seem to happen because of unforeseen events. Why were they unforeseen?

Lean energy management requires that the exploration space for the risking of problems be ever expanding, with a feedback loop so that every new event from wherever it occurs in the company's worldwide operations is captured in computer memory and can never be unforeseen again.

In such a lean world, customization tendencies are fought at every turn, interface issues are dealt with continuously, and everything is measured so that it can be scored and formally managed via software support. This progression is well known in other industries as a natural path up the knowledge management cube. This description of lean energy management seems complicated, but in fact it is simple. There are five basic rules:

1. A 5:1 ratio of modeling and simulation versus field operations U compared to standard practices in our industry of a 1:5 ratio! Spend 5 times more than today in upfront computing of possible outcomes and the plan to deal with them becomes a moneymaking proposition.

2. The same operations model must be used by all. Everyone has the same model, and it is updated in real time.

3. Metric everything, all the time! Only then can it be managed.

4. Constantly ask whether you are making more money? And above all else,

5. Long-term commitment from management is required. If you say you are already doing 1 to 3 but are not saving 50% in both cycle time and costs, then your company hasn't gotten the concept of lean yet.

At Columbia, we have developed a series of implementation steps to migrate a company that is already good at knowledge management to process optimization and lean energy management using Machine Learning.   We begin by populating a series of matrices that are adept at learning, which we call suitability matrices because they describe the sequence from symptoms-to-problems and then problems-to-solutions for any given contingency. Composition and hierarchy are derived through the matrix formalization and used to guide the ML feedback loop towards optimal decision support. We map the "as is" processes before using the ML algorithms and human-in-the-loop feedback to recommend the "to be" state.

Chapter 4: Implementation Steps of CALM

Step 1. Encode Learning into a Simulation Model: It is absolutely necessary that all levels of “execution tree” understand and implement the lean practicum developed by the other industries. A model capable of simulating the operations of the “plant” is required. Capturing the decision trees from actions to subsequent reactions are therefore central to accurate simulations necessary for proper evaluation of profitable production scenarios throughout the life-cycle of ultra-deepwater plants. Columbia uses a knowledge elicitation method we call the Suitability Matrix as a tool for building the Lean practicum and populating the simulator. The Suitability Matrix captures and encodes the knowledge from best practices elicitations from experts and knowledge bases. The suitability matrix tool set includes a user interface that uses 3D visualization and web-based explanation with a graphical user interface to collect feedback from the experts in each company.

Suitability Matrices also perform Gap Analyses, problems-to-solutions tracking, and construction of a "House of Quality", a widely used method for mapping from customer requirements to product strategy. The latter contains the How, What, Where, Why, and How Much information that is used to evaluate Lean strategies and verify that there is not overlap of technologies or services.

Step 2. Develop Real Options Capabilities for the Economic Model: Real Options are used in industries such as pharmaceuticals, aerospace, and on wall street as a guide to make quantitative evaluations of what is most likely to be useful to systems improvement through all life-cycle stages of a product Decision and financial science discipline must be developed and applied at all levels of the Lean system in the energy Industry, as well. Decision-making-under-uncertainty and conceptualization-to-product tools are absolutely necessary ingredients in the Lean Economic Model. They include not only Real Options, Gap Analysis, House of Quality, and Suitability Matrices (neural nets), but also EVA, Balanced Scorecards, and other more advanced metrics to actionable response tools such as the Metrics Thermostat and The Virtual Customer.

Step 3. Overlay a Reinforcement Learning Harness: Adaptive learning algorithms to "close the loop" are then required to assemble a Lean, gap-free integration of these core business processes. Once process integration is wired, it can be measured and online adaptive learning algorithms used to first monitor, then learn, based on gradients in the processes measured. Both the Real Option framework of Step 2 and the adaptive control in the Learning Harness of Step 3 have used dynamic programming as a means of evaluating future actions under uncertainty. These are then integrated into a Reinforcement-Learning-Real-Option Evaluator that is then used to adapt and find systemic synergies among project efforts via simulations. The learning's from this system are applied to broader use through the Learning Harness. Parallel design evaluations then allow the Real Option valuator to drive cross-system optimization. Processes can then be transferred *literally* to new development planning for the Ultra Deepwater.

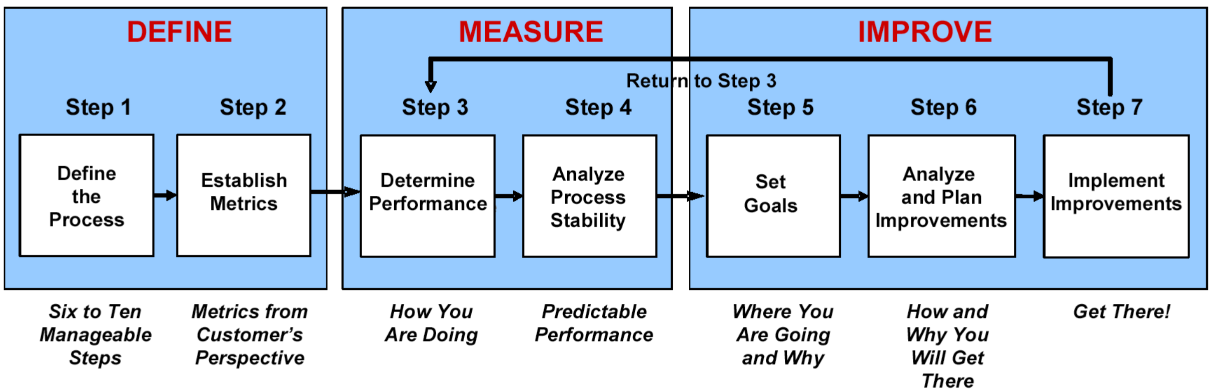
Step 4. Create Transparent Performance Metrics: Performance metrics and their efficacy in tracking the improvements in innovation processes are critical to Lean process improvement. Traditional implementations like balanced score cards can be used with the learning harness to develop an adaptive control based Metrics Thermostat.

CALM Implementation Details

In order to implement CALM for a project, a company needs:

- Identification of the Business Capability and enumeration of its objectives that define the required performance improvement

- Road maps, chained matrices and simulation Models of the processes and workflows comprising the desired Business Capability
- Tracking of the actions that effect the processes of the Business Capability – an Action Tracker
- Metrics that quantify the response of the system to those actions
- Identification of the locations of flexibility in the system, and Real Options for improvement of the Business Capability utilizing that flexibility
- Continuous reassessment of internal and external Risks and Uncertainties contributing to the Business Capability
- An automated means of then generating Steering Signals at all levels of the operation to drive the system towards more and more positive objectives – the feedback loop required for Machine Learning.



The CALM principals are discussed in more detail below:

Identification of the Business Capability

The organizational barriers and relationships must first be described, and technologies and processes used by both system components and people in the project mapped.

A Model of the System

An axiom of control theory is: "**to control a process, one first must understand the system well enough to model it**"– either implicitly or explicitly. Using the chained matrices approach, one can first map the processes involved in a business capability. Later, a higher fidelity computer model representing behavior of an airframe, or how electric power flows, or a fluid simulation of a reservoir, must be integrated with these more discrete models. For an oil and gas field, both the above-ground "plant and facilities" model and the below ground "reservoir" model are necessary for CALM. They, in turn, must communicate with each other and give compatible results.

Action Tracking

Untold loss of intellectual and real capital occurs because of the lack of the recording of all the actions affecting a process. We have found that realizing this capability often is a **major** IT problem because it requires tracking actions from application to application and person to person across the software inventory and people workflow processes of the Business Capability. One must build a database of these software and people interactions in order to track actions, as well as build an archive of the context of each action. They must then be "playable again" in a simulation mode, so that incorrect actions can be analyzed, understood, and prevented from happening again.

Scoring the Responses to Actions

CALM requires that metrics must be produced to score the response of the business capability to each action taken, so that the system can learn from and be optimized to better and better actions. It is an amazing fact that in non-Lean organizations such as those in our industry, bad actions are consistently repeated, over-and-over again.

Flexibility and Real Options

This important step is at the core of the constant reevaluation of a Business Capability as Lean organizations improve and adapt to a world of increasing volatility. Since under a real option framework, there is value to adding flexibility under uncertainty, any system will be driven to maximize its locations of flexibility in lean implementation. Several previous parts of this continuing series in Oil & Gas Journal have dealt with the need for real options evaluation methodologies in our industry.

Continuous Reassessment of Risk and Uncertainty

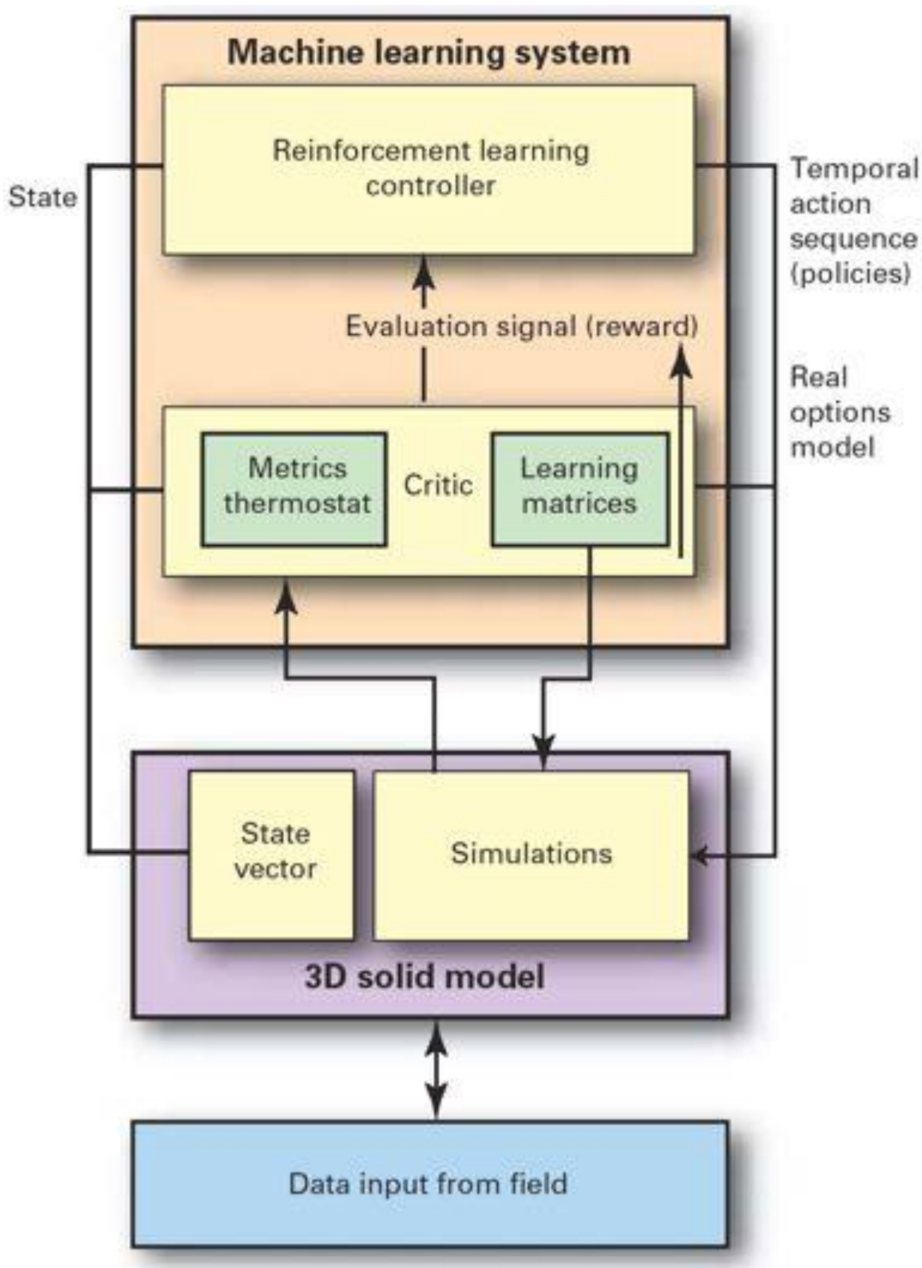
Likewise, risks and uncertainties must be continuously reevaluated. We envisage that oil and gas fields of the future will remotely use live price and expense data streams, and their volatilities, to generate control signals via CALM to optimize performance automatically.

Automated Steering Signals to Optimize Performance

The generation of steering signals is the critical part of CALM that is often missing from organizations. We have discussed one mechanism for generating such steering signals using the Metrics Thermostat (see Part 5 of our Lean Energy Management series). The Metrics thermostat is a way of using implicit, rather than observable, explicit models for generating the steering signals. A more robust and scaleable approach is to map both the processes and steering signals at multiple levels using chained matrices based on the back-propagation algorithm from neural networks (Figure 7). A method of generating steering signals in all kinds of business capabilities is presented in Webs, 1974. In general, these steering signals must anticipate and adapt to external and internal uncertainties and locations of flexibility within the Business Capability, simultaneously, which is a step beyond both the metrics thermostat and back-propagation.

Putting it all together into a CALM System

CALM has steadily evolved from simple process control to information and then knowledge management and currently to real-time optimization of the product being manufactured. Lean systems not only tell the decision maker what might happen next, but they also present contingency information in a clear and concise way at all times. Operators particularly need help when multiple areas have significant problems at the same time, and lean decision support provides not only what is likely to happen next but also what are the risks and ramifications of different preventive remediation sequences. "Pain indices" that record actions taken then provide a basis for future ML so that the system gets better and better at its decision support and training jobs.



The CALM System can be used to advise or guide operators on what actions to take in the same manner that common car navigation systems are used to guide or direct car drivers. The Lean controller also continually tracks the current state of the drilling systems to provide look-ahead contingency analyses, a common practice in other industries. The Lean controller is configured to evaluate opportunities as real options (see previous parts of this series). The learning system uses feedback to generate actions or decisions that are always in the money (i.e., a martingale in business terms) with respect to both financial profitability and engineering efficiency.

Chapter 5: The Future of Computer-Aided Lean Energy Management

Assembled together, the Encoded Learning, Real Options Reinforcement Learning Harness, and Performance Metrics tracking form an Economic Model that drives all management implementation decisions. Lean Innovation Management then has an Economic Model or "Engine" that will continually address the following basic challenges:

- The functional process-mapping cuts through much of the complexity that clouds understanding of the real drivers in product design, fabrication and operation under constantly changing uncertainty.
- It is easy to modify/update so that the mapping of new architectures can be incorporated easily to keep the fully integrated system dynamically viable.
- The Lean Energy system builds a "held-in-common" process rigor for the ultra-deepwater industry to advance the pace of innovation, and couples this capability with the option to expand the system to enable broader collaborations and new "out-of-the-box" architectures.

Micro-Options: In turn, Lean Energy Management lays the foundation for silicon-based *Micro-Options* for the ultra-deepwater in the future. Within the next several years, inexpensive silicon will be attached to every component of all complex systems in aerospace, automotive, and general manufacturing. These ubiquitous chips will have the following general characteristics:

- Geo-located via GPS or other means such as wireless triangulation, inertial, etc. (There is the remaining challenge to communicate and geo-locate underwater components.)
- Sensors, such a temperature, humidity, power consumption, light level, tilt, and acceleration will be incorporated into the silicon.
- Product specifications -- physical, operational, and fiscal. Eventually enough silicon to even simulate different uses for itself. The real options available to each component will be embedded in its own silicon – thus enabling Micro-Options.
- 2-way Wireless (eventually UWB) to each component to form dynamic peer-to-peer, self-healing, aware networks. The network will bring the market to the last mile for real options valuation.

- Grid computing using these networks will be used for real option evaluation of the hierarchal system of components. Plug and play modular design components will have a silicon face to them. These networks will be self-organizing, and self-aware during the different phases of the lifecycle of a complex system deployment.

The enterprise-wide use of these coming silicon-based Micro-Options for the ultra-deepwater will include:

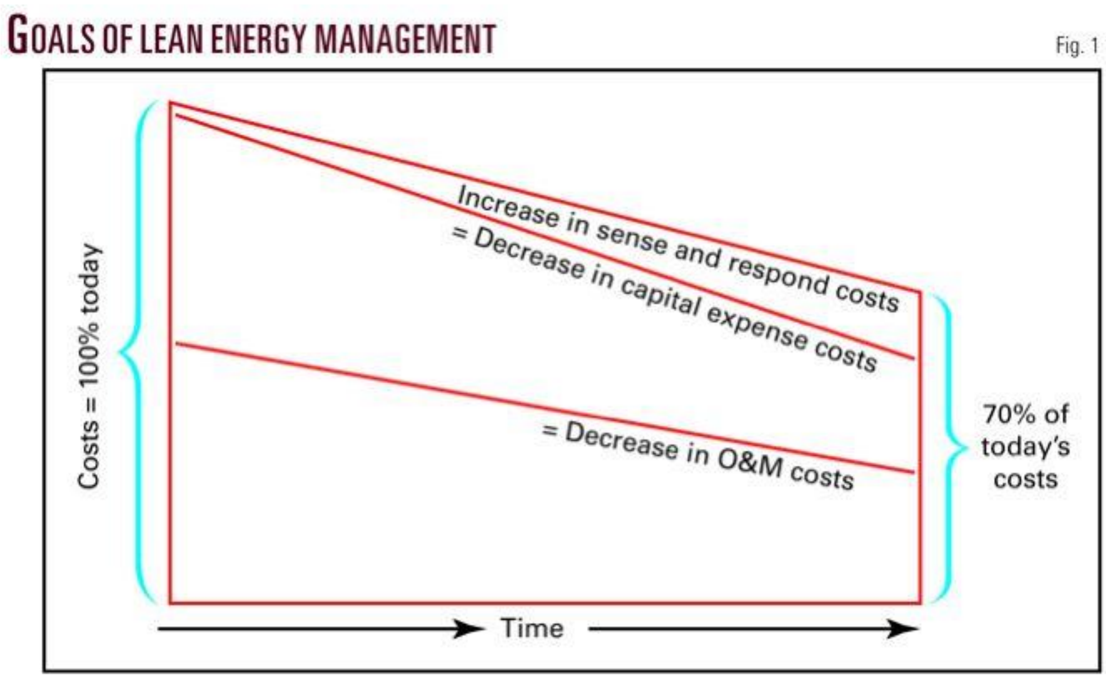
- **Design-Build:** The silicon will be created as the design of components develops. The silicon will form a grid-computing network to simulate the operation and to optimize design decisions. This includes the economic and risk side. Training on future operations will use this network.
- **Fabrication:** The silicon will be attached to the component and will be used to track its completion. The master database for the whole platform will be the embedded silicon in all the components being fabricated.
- **Shipping and Deployment:** When shipped, the silicon attached to each component will form a vigilant self-organizing network aware of intrusion, theft, alteration, etc. During deployment the attached silicon will be vigilant for dangerous or harmful situations caused by bad construction, assembly methods, or cheating by the contractor. It will also be vigilant for ISO 9000/14000 and other future standards violations in deployment.
- **Operation:** Once deployed the system of components will form a sensing and computational grid that will monitor itself and its environment. Smart Sensors and smart materials incorporated will form a vigilant system looking for variances from plan and simulating its own physical and economic performance. Real options valuation of operational decisions will be done at the lowest level and propagated upwards the system hierarchy. The peer-to-peer networked system will be able to simulate itself for adaptive control. Based on the reinforcement learning harness, the adaptive control will use dynamic programming to optimize its own operations. Real options valuation will be used as metrics along with others such as regulatory measures for safety, MMS, environmental and other regulations, etc. The master database of the operation will be the life-cycle silicon attached to each component.
- **Abandonment:** The silicon will be vigilant for a safe and environmentally sound decommissioning and eventual recycling of certain components. Even in the scrap yard, the silicon has use.

In the future, the distinctions of the life-cycle phases will become blurred. With Lean Energy Management infrastructure in place and operational in the ultra-deepwater, modular design, fabrication, assembly and operation will be driven by Micro-Options rigor and economic return-on-investment will skyrocket – the irony is that only such a “Lean Energy Revolution” will make it truly affordable in the first place.

Chapter 6: Machine Learning in CALM

CALM requires that this data analysis, modeling, and performance evaluation be done all-day, every-day. Then and only then can the system be empowered to continuously learn in order to improve performance. The increased costs for migrating to this new "sense and respond" operational framework are easily offset by subsequent decreases in Capex and O&M costs that have been documented in industry after industry.

Add computational machine learning to the data analysis loop and you have what is termed an *adaptive aiding system*. A car navigation system is one common example of this. The car's GPS system learns when you make a wrong turn and immediately re-computes a new recommended course-correction. It is the feedback loop of such CALM systems that contains the newest and most unfamiliar computational learning aids, so we need to step through the progression of technological complexity in more detail. Actions taken based upon information coming in are objectively scored, and the metrics that measure the effectiveness of those actions then provide the feedback loop that allows the computer to learn. In other words, once software infrastructure is in place, the continual recycling between decisions and scoring of success or failure throughout the organization are used to guide operators to take the best future actions.



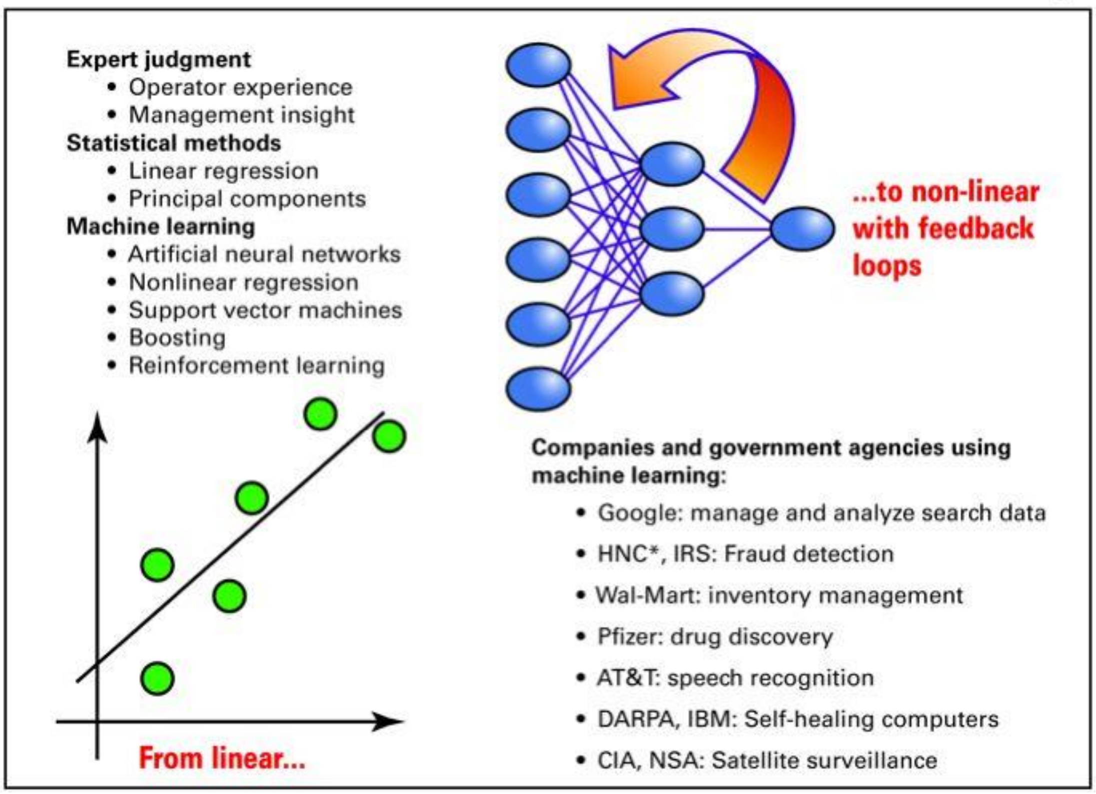
Computational Learning

Computational or machine learning has proved effective at predicting the future in many industries other than energy (c.f. http://www.berkeley.edu/users/breiman). Luckily, the field of computational machine learning has recently extended the range of methods available for deriving predictions of future performance (Figure 2). Examples of successful computational learning abound. It is now used to interpret user queries in

Microsoft Windows, and to choose web advertisements tailored to user interests in Google and Amazon.com, for example. In aerospace, computational machine learning has driven the progression from flight simulators that train pilots, to computers that fly the plane completely on their own, and now to the newest Unmanned Combat Air Vehicles like the X-45 that can dogfight with the best "Top Gun" pilots. Other successes are found in the progression from speech recognition to synthetic conversation, and now to synthesizers of journalism itself (see http://www1.cs.columbia.edu/nlp/newsblaster). In the automotive industry, there are new navigational aids that can park a car, not just assist the driver. Examples of other successful uses of machine learning are given in the on-line appendix for this article, available from http://www.ogjonline.com .
Machine Learning methods effectively combine many sources of information to derive predictions of future outcomes from past performance. Individually, each source may only be weakly associated with something that we want to predict, but by combining attributes, we can create a strong aggregate predictor. Additional advantages of computational machine learning over traditional statistical methods include the ability to take account of redundancy among the sources of evidence in order to minimize the number of attributes that need to be monitored for real-time assessment and prediction.

EVOLUTION IN COMPUTATIONAL LEARNING METHODS

Fig. 2



Suppose we want to classify a data stream into like-performing characteristics. If we have a lot of data about what we want to predict, we will need a complex function that uses almost everything we know about the object, and still we will have imperfect accuracy. To accomplish this, we must start with some already-classified example data we can use for *training* a Machine Learning (ML) system. ML techniques allow the system to find good classifying and ranking functions in a reasonable amount of computer time for even the largest of data sets. Although there are a wide variety of machine learning approaches, they have common features:

1. Adding more data over time improves accuracy. With just a few points, the ML algorithms can make an educated guess. As the number of data points rises, the confidence and precision of the results rises also.
2. Each ML technique prefers some explanations of the data over others, all else being equal. This is called an "inductive bias". Different ML approaches have different biases.

ML techniques mathematically combine the results of computations that produce a score for each object reflecting the estimation of its likelihood of doing something, like failing, from an evaluation of all the variables of the data for that object. Those with the highest scores may be viewed as being predicted to fail soon. The scores can be sorted, resulting in a priority order ranking for taking preventive actions among many similar objects. Which compressor in a large production facility to overhaul next is a common example.

The data we wish to analyze will always have many *dimensions* -- each dimension is an attribute of the data. For instance a compressor has a large number of attributes, such as its age, what it is used for, its load, its configuration, etc. Every object analyzed by a ML algorithm is described by a large vector of these data points. For example, compressor #433 has an age = 20 years; its location is 40's field; its peak load to rating is 80%, …etc. If there are only two or three attributes per data point, we can view each reading as a point on either a plane or in 3-dimensional space. We will usually have many more attributes than that, and thus be working with many dimensions. High-dimensional mathematics works in a similar way to the math of two and three dimensions, even though our visual intuition fails. The techniques described below can be extended to thousands, or even hundreds of thousands of dimensions. Further, special algorithmic techniques have been devised to make the larger dimensionality problems manageable.

Below, we describe the sophisticated mathematical approaches to machine learning in the simplest two and three-dimensional context so they can be easily understood. Among the most basic problems attacked by ML is learning how to sort items into classes. The same techniques can then be expanded from classification to rankings. For instance, ML can begin by classifying which compressors are at extreme risk and which are not, at significant risk or not, at moderate risk or not, etc, and then the data can be used again to calculate a ranking of the risk of imminent failure for every compressor in the inventory.

Modern ML methods called Support Vector Machines (SVMs) and Boosting have largely replaced earlier methods such as the so-called Artificial Neural Networks (ANNs), based on a crude model of neurons. ANNs and other early machine learning methods are still widely used in the oil and gas industry. The modern methods have significant advantages over earlier ones. For example, ANNs require extensive engineering-by-hand – both in deciding the number and arrangement of neural units and in transforming data to an appropriate form for inputting to the ANN. ANNs have been replaced in the financial, medical, aerospace, and consumer marketing worlds by SVMs and Boosting because the new techniques use data as is and require minimal hand-engineering. Unlike ANNs, both SVMs and Boosting can deal efficiently with data inputs that have very large numbers of attributes. Most importantly, there are mathematical proofs that *guarantee* that SVMs and Boosting will work well under specific circumstances, whereas ANNs are dependent on initial conditions, and can converge to solutions that are far from optimal. Details and web links to more information on ML are in the on-line appendix.

Support Vector Machines (SVMs)

Support Vector Machines were developed under a formal framework of ML called [Statistical Learning Theory](). SVMs look at the whole data set and try to figure out where to optimally place category *boundaries* are that separate different classes. In three-dimensional space, such a boundary might look like a plane, splitting the space into two parts. An example of a category might be whether or not a compressor is in imminent danger of failure. The SVM algorithm computes the precise location of the boundary -- called a “hyper-plane” -- in the multi-dimensional space. By focusing on the points nearest the boundary, which are called “support vectors,” SVMs define the location of the plane that separates the points representing, in our running example, compressors that are safe, from those that are in imminent danger of failure. SVMs work even when the members of the classes do not form distinct clusters.

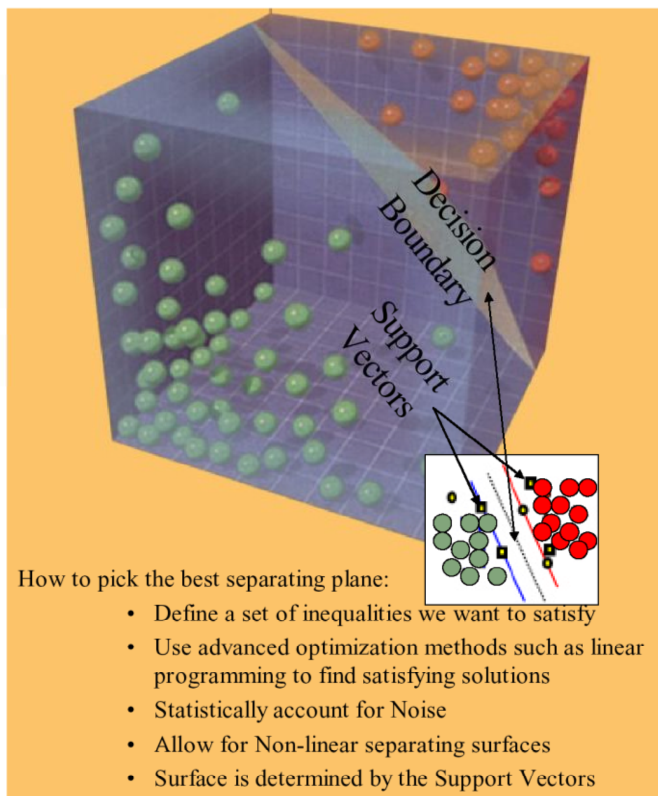


Figure 14. Simplify the discovery of a decision boundary between a grouping of “go-decision” case-history outcomes in green versus “no-go” in red by cutting the dimensionality of the data being mined (whatever it is) so that the decision space can be described by only a few “support vectors” that define the space -- with error -- between the green and red clusters. The computer algorithm that defines the surface is the “Support Vector Machine”

In the Monte Carlo simulation approach, system reliability evaluation is performed by determining the state of each component and, by the application of an SF, assessing if the system succeeded or failed. A single simulation run or data captured from a crisis generates either a system success or failure, and multiple simulation runs or events
can be used to determine the reliability estimator. Since the method requires a large number of SF evaluations, it is convenient to substitute this evaluation with a fast, approximated, algorithm?]

SVM is an estimation algorithm (“learning machine”) in which the training phase involves optimization of a convex cost function, hence there are no local minima to complicate the learning process. Testing is based on the model evaluation using the most informative patterns in the data (the support vectors). Performance is based on error rate determination as test set size tends to infinity.

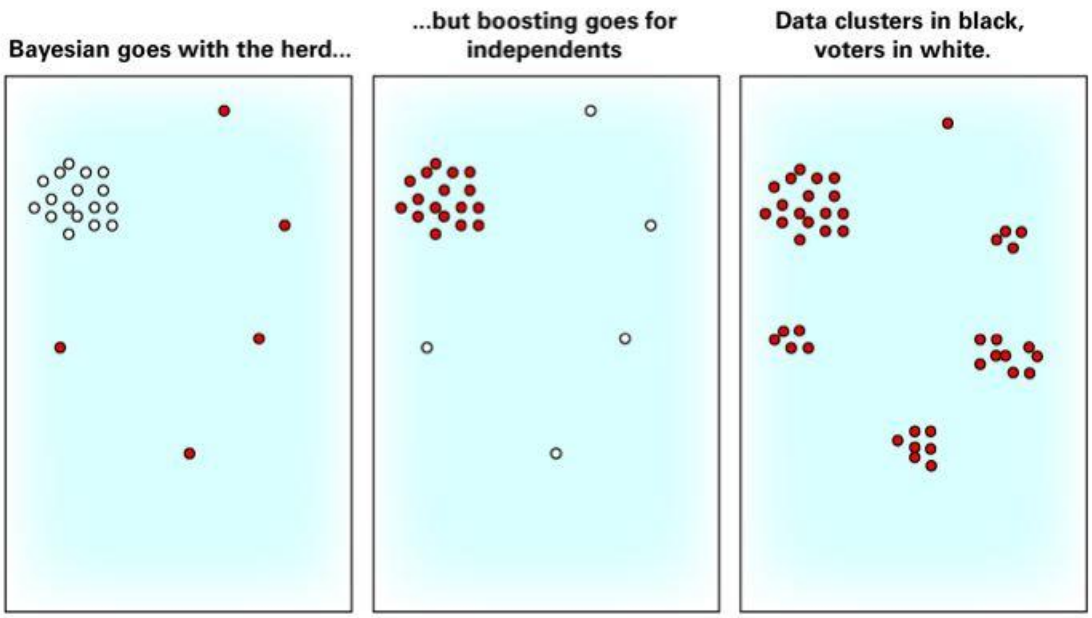
One of the simplest methods for classification is to:

- View a training dataset as points, where the position of each point is determined by the values of its attributes within a multi-dimensional space.
- Use SVMs to find a plane with the property that members of one class lie on one side, and members of the other class lie on the other side of the plane.
- Use the hyper-plane derived from the training subset of the data to predict the class of an additional subset of the data held out for testing.
- After validation, the hyper-plane is used to predict that future data is of the same class as those in the training set if it falls on the same side of the hyper-plane.

Finding a hyper-plane that separates two classes is fairly easy, as long as one exists. However, in many real-world cases, categories cannot be separated by a hyper-plane, as illustrated in Figure 3. In SVMs, extra variables are computed from the original attributes to further separate the data so a classification plane can be found. For example, two categories that can't be separated with a simple line in one dimension (Figure 3) can be separated if we convert them into two-dimensional space by adding an extra variable that is the square of the original variable. The data can now be separated by a hyper-plane.

CATEGORIES CANNOT BE SEPARATED BY A HYPERPLANE

Fig. 3



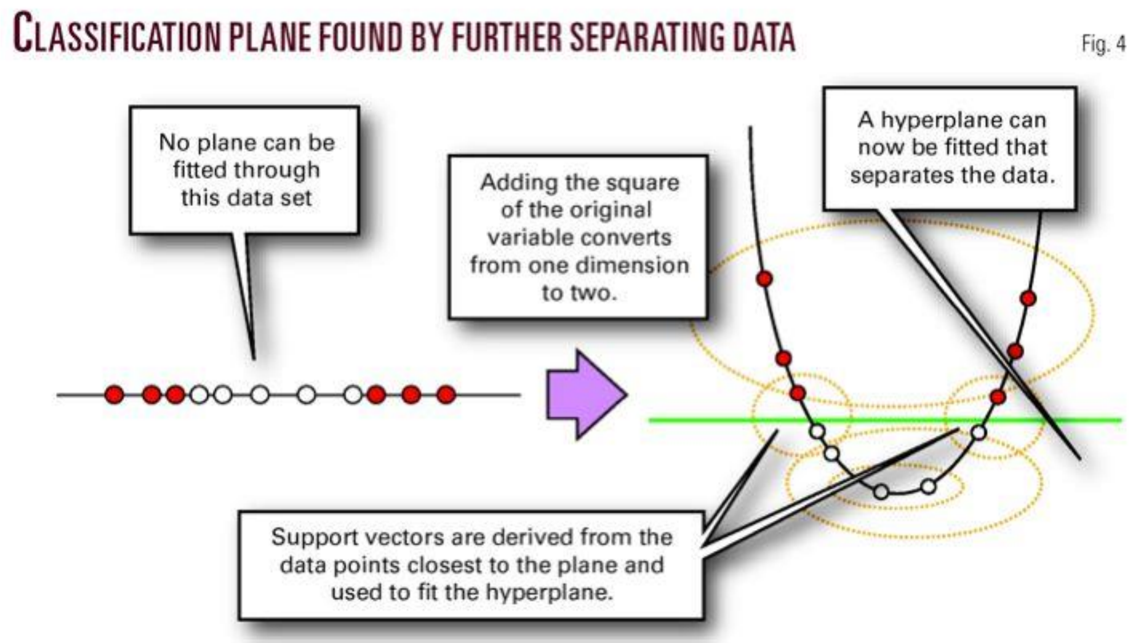
The breakthrough with SVMs was the discovery of how to compute a large number of variables, embed all that data into very high-dimensional space, and find the hyper-plane *effectively* and efficiently. Another important aspect of SVMs is that the hyper-plane is chosen so that it not only correctly separates and classifies the training data, but ensures that all data points are as far from the boundaries of the hyper-plane as possible. Avoiding overdependence on borderline cases protects against "over-fitting," a dangerous condition that produces "false positives" because the ML algorithm's output has captured insignificant details of the training data rather than useful broader trends. We might then conduct preventive maintenance on the wrong compressor in our running example.

Boosting

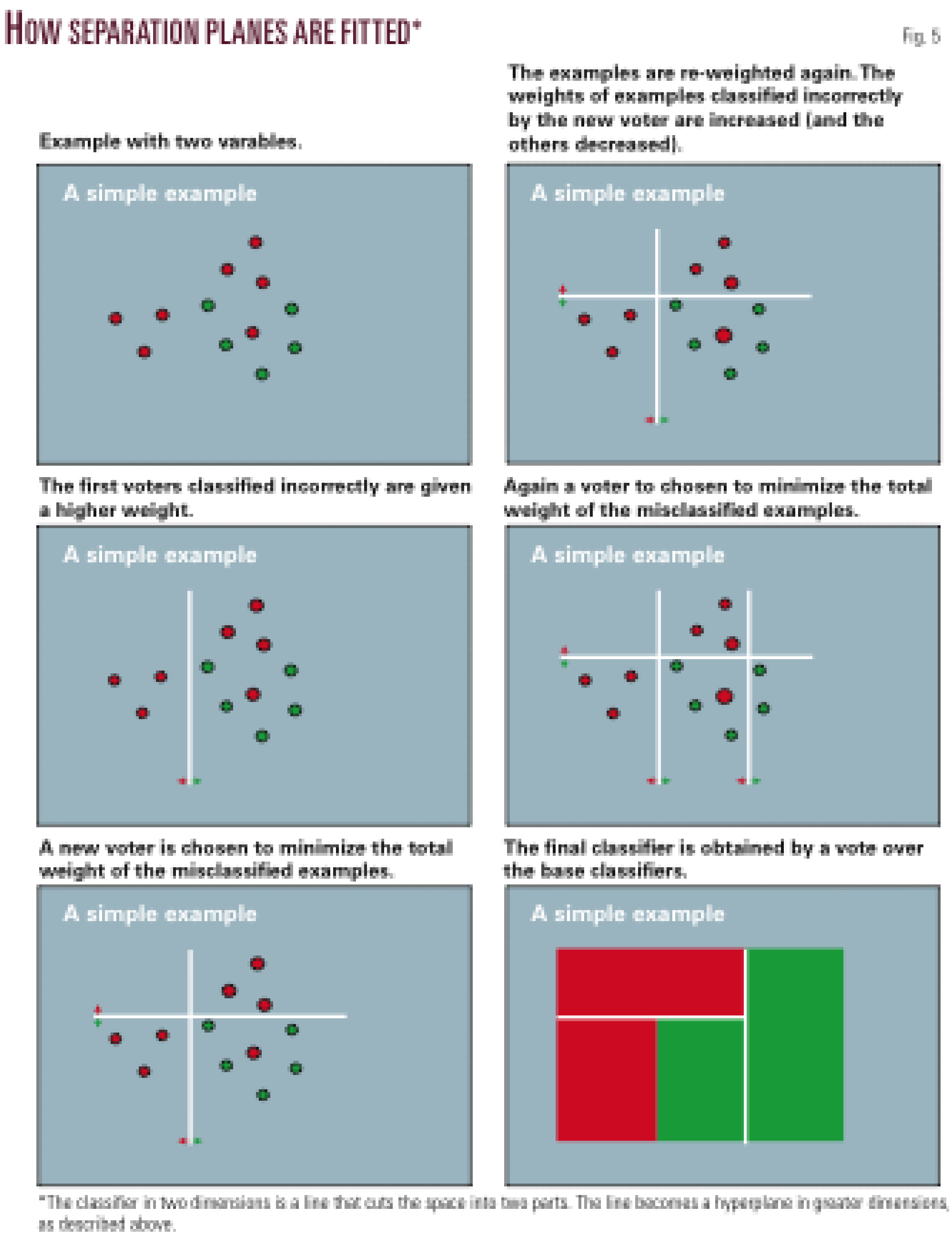
All companies struggle within limited O&M budgets to deploy enough information sensors to make preventive maintenance predictions, so it is critical to determine what attributes are really needed versus those that are less useful. A ML technique called "Boosting" is especially good at identifying the smallest subset of attributes that are predictive among a large number of variables. Boosting often leads to more accurate predictions than SVMs, as well.

Boosting algorithms seek to combine alternative ways of looking at data by identifying views that complement one another. Boosting algorithms combine a number of simpler classification rules that are based on narrower considerations into a highly accurate aggregate rule. Each of the simple rules combined by boosting algorithms classifies an object (such as a compressor) based on how the value of a single attribute of that object compares to some threshold. An example of such a rule is that anyone in a crowd is a basketball player if his height is above 7 feet. While this rule is weak because it makes many errors, it is much better than a rule that predicts entirely randomly. Boosting algorithms work by finding a collection of simpler classifiers such as these, and then combining them using *voting*. Each voter is assigned a weight by how well it does in improving classification performance, and the aggregate classification for an object is obtained by summing the total weights of all the classifiers voting for the different possibilities.

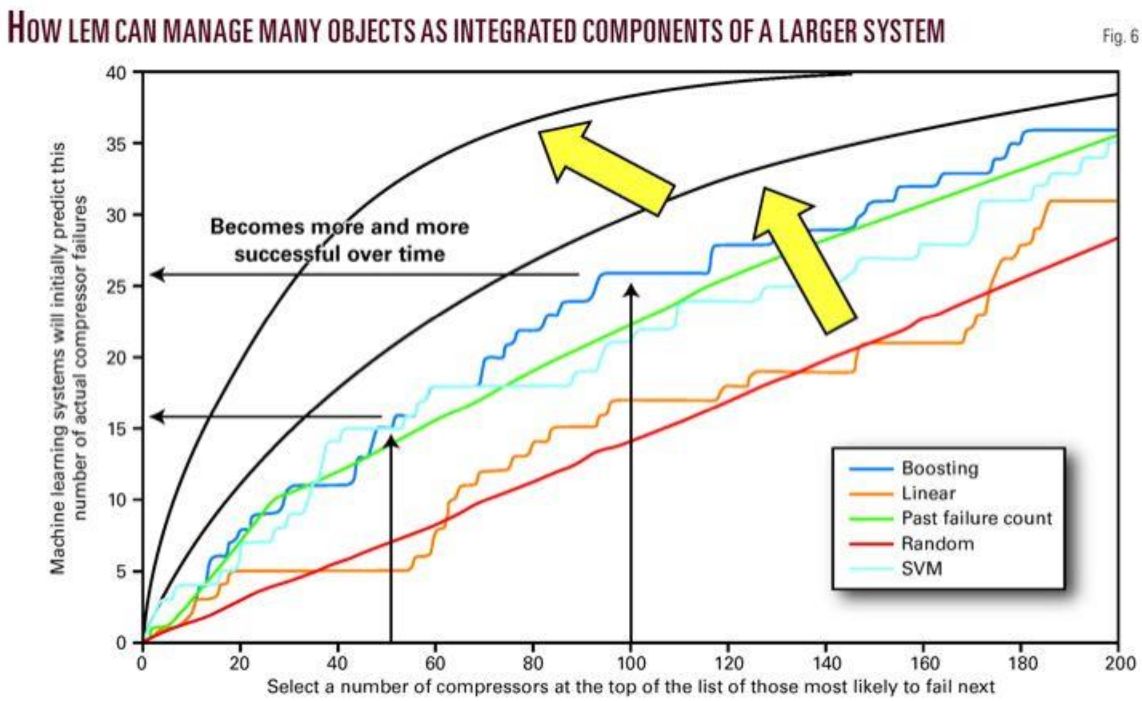
Voting increases reliability. If each voter is right 75% of the time, taking a majority vote of three voters will be right 84% of the time, five voters 90%, seven voters 92%, and so on. However, this only works if the voters are independent in a statistical sense. If there's a *herd* of voters that tends to vote the same way, right or wrong, it will skew the results, roughly as if there were fewer voters. Boosting algorithms try to only pick a single representative voter from each herd: undemocratic, but effective. In general, boosting tries to select a group of voters based on two competing goals: choosing voters that are individually pretty good and choosing voters that independently complement one another as sources of evidence of the correct class of a typical object. In the following diagrams (Figure 4), the points in the left and center represent voters (not data), and the distance between voters corresponds to the extent of similarity in their behavior.



The series of diagrams in Figure 5 illustrates the way the separation planes are fitted. The simple example is run for only three rounds on only two variables. In actual usage boosting is usually run for thousands of rounds and on much larger datasets. Boosting adds an additional classifier to the list for each round. Before a given round, the *examples* are re-weighted to assign more importance to those that have been *incorrectly* classified by previously chosen classifiers. A new classifier is then chosen to minimize the total weight of the examples that it misclassifies. Thus, boosting looks for succeeding classifiers that make errors in *different places* than the others, and thus minimizing the errors.



In CALM, we use the machine learning tools that best fit each problem. We try several, and usually one will work better on some portions of the data than the others. Our purpose in classifying is to gain fundamental understanding of the relationships between cause and effect among the many attributes contributing to the performance of each object. CALM can then manage many objects as integrated components of a larger system. Consider the real-life example in Figure 6. A good initial predictor can be derived by just looking at the number of past failures on each component (green curve), but nothing more is understood about cause-and-effect from an analysis of only this one variable. If in addition, we apply both SVM and Boosting, we gain insights into how the many attributes contribute to failure beyond just whether the component failed in the past or not. SVMs have an inductive bias towards weighing variables equally. In contrast, the inductive bias of Boosting penalizes members of what appear to be "herds."



SVMs often produce accurate predictions, but it is frequently difficult to get deeper intuition from them. In other words, SVM classifiers are often "black boxes." Some insights may be possible if there are very few support vectors, but the number of support vectors rises rapidly with the amount of input data and the noisiness of the data. Boosting has a better chance of giving actionable results, and it is particularly good at identifying the hidden relevance of more subtle attributes – an important benefit when looking for what is important to measure for preventive maintenance programs.

The above has described only static variables that do not change over time. ML analysis of the sequence of changes over time of dynamic variables is an important additional determinant for root cause analysis in CALM. Magnitude and rate of change of the variables can be used to derive the characteristics leading up to failure of a group of objects so that precursors can be recognized for the overall system. For compressor failure, for example, accumulated cycles of load can be counted and added as an attribute. Such dynamic variables are important in fields such as aerospace, where airplane take-off and landing cycles are more critical predictors of equipment failure than hours in the air. As these dynamic, time varying attributes are added to the ML analysis, new solution planes can be re-calculated every time a new data update occurs, and the differences between past solutions analyzed. Over time, the results migrate from prediction to explanation, and the feedback continues to improve prediction as the system learns how to optimize itself.

Examples of successful machine learning abound in medicine and transportation, in particular. Consider the progression from storing patient medical records to synthesizing patient data to reach a machine driven diagnosis. Or the progression from flight simulators that train pilots in how to fly to flight controllers that fly the plane on their own – even dogfight. Other successes are in the progression from speech recognition to synthetic speech conversation, and from driver aids such as navigation to a car that can park itself.

Reinforcement Learning Detail

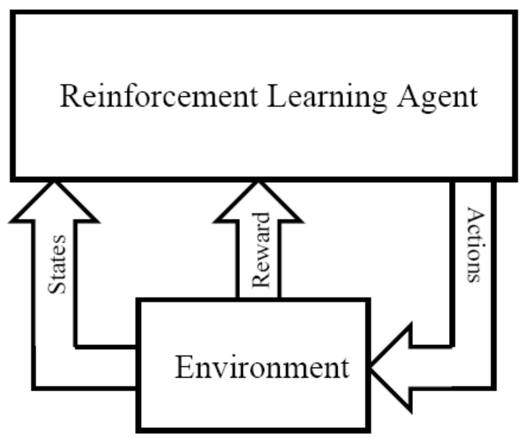


Figure 15. Reinforcement learning closes the loop between *sensing* and *acting*.

Reinforcement learning is learning what to do---how to map situations to actions---so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. ***Thus reinforcement learning can "close the loop" and as such represents a key step in fully realizing the potential of Know Management systems as discussed above (see Close the Feedback Loop).*** In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards. These two characteristics---trial-and-error search and delayed reward---are the two most important distinguishing features of reinforcement learning.

Reinforcement learning is different from supervised learning, the kind of learning studied in most current research in machine learning, statistical pattern recognition, and artificial neural networks. Supervised learning is learning from examples provided by some knowledgable external supervisor. This is an important kind of learning, but alone it is not adequate for learning from interaction. In interactive problems it is often impractical to obtain examples of desired behavior that are both correct and representative of all the situations in which the agent has to act. In uncharted territory---where one would expect learning to be most beneficial---an agent must be able to learn from its own experience.

One of the challenges that arises in reinforcement learning and not in other kinds of learning is the tradeoff between exploration and exploitation. To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions it has to try actions that it has not selected before. The agent has to *exploit* what it already knows in order to obtain reward, but it also has to *explore* in order to make better action selections in the future. The dilemma is that neither exploitation nor exploration can be pursued exclusively without failing at the task. The agent must try a variety of actions *and* progressively favor those that appear to be best. On a stochastic task, each action must be tried many times to reliably estimate its expected reward. The exploration--exploitation dilemma has been intensively studied by mathematicians for many decades.

- A **policy (?)** is a stochastic rule by which the agent selects actions as a function of state. Also known as a control strategy. An **optimal policy (?*)** maximizes award. There can be several optimal policies. Reinforcement learning methods specify how the agent changes its policy as a result of experience.
- A **value function** is the expected return from each state (V), or state-action pair (Q), given that the agent follows a particular policy. Also known as the cost-to-go function.
  - Š An optimal value function and greedy actions produce an optimal policy ?*
- A **trajectory** is a sequence of actions guided by the policy over time to reach the reinforcement learning agentÕs goal.

Table 5. Reinforcement Learning Terms

One of the larger trends of which reinforcement learning is a part is that towards greater contact between artificial intelligence and other engineering disciplines. Not all that long ago, artificial intelligence was viewed as almost entirely separate from control theory and statistics. It had to do with logic and symbols, not numbers. Artificial intelligence was large LISP programs, not linear algebra, differential equations, or statistics. Over the last decade this view has gradually eroded. Modern artificial intelligence researchers accept statistical and control-theory algorithms, for example, as relevant competing methods or simply as tools of their trade. The previously ignored areas lying between artificial intelligence and conventional engineering are now among the most active of all, including new fields such as neural networks, intelligent control, and our topic, reinforcement learning. In reinforcement learning we extend ideas from optimal control theory and stochastic approximation to address the broader and more ambitious goals of artificial intelligence.

A good way to understand reinforcement learning is to consider some of the examples and possible applications that have guided its development:

- A master chess player makes a move. The choice is informed both by planning---anticipating possible replies and counter-replies---and by immediate, intuitive judgments of the desirability of particular positions and moves.

- An adaptive controller adjusts parameters of a petroleum refinery's operation in real time. The controller optimizes the yield/cost/quality tradeoff based on specified marginal costs without sticking strictly to the set points originally suggested by human engineers.
- A gazelle calf struggles to its feet minutes after being born. Half an hour later it is running at 30 miles per hour.
- A mobile robot decides whether it should enter a new room in search of more trash to collect or start trying to find its way back to its battery recharging station. It makes its decision based on how quickly and easily it has been able to find the recharger in the past.
- Phil prepares his breakfast. When closely examined, even this apparently mundane activity reveals itself as a complex web of conditional behavior and interlocking goal-subgoal relationships: walking to the cupboard, opening it, selecting a cereal box, then reaching for, grasping, and retrieving the box. Other complex, tuned, interactive sequences of behavior are required to obtain a bowl, spoon, and milk jug. Each step involves a series of eye movements to obtain information and to guide reaching and locomotion. Rapid judgments are continually made about how to carry the objects or whether it is better to ferry some of them to the dining table before obtaining others. Each step is guided by goals, such as grasping a spoon, or getting to the refrigerator, and is in service of other goals, such as having the spoon to eat with once the cereal is prepared and of ultimately obtaining nourishment.

These examples share features that are so basic that they are easy to overlook. All involve *interaction* between an active decision-making agent and its environment, within which the agent seeks to achieve a *goal* despite *uncertainty* about its the environment. The agent's actions are permitted to affect the future state of the environment (e.g., the next chess position, the level of reservoirs of the refinery, the next location of the robot), thereby affecting the options and opportunities available to the agent at later times. Correct choice requires taking into account indirect, delayed consequences of actions, and thus may require foresight or planning.

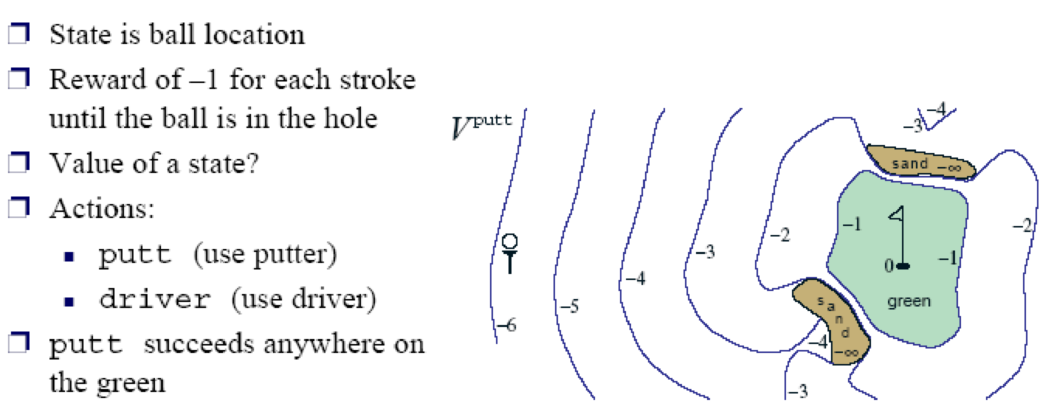


Figure 16. Golf as a reinforcement learning problem. The value function is depicted as contours.

At the same time, in all these examples the effects of actions cannot be fully predicted, and so the agent must frequently monitor its environment and react appropriately. For example, Phil must watch the milk he pours into his cereal bowl to keep it from overflowing. All these examples involve goals that are explicit in the sense that the agent can judge progress toward its goal on the basis of what it can directly sense. The chess player knows whether or not he wins, the refinery controller knows how much petroleum is being produced, the mobile robot knows when its batteries run down, and Phil knows whether or not he is enjoying his breakfast.

In all of these examples the agent can use its experience to improve its performance over time. The chess player refines the intuition he uses to evaluate positions, thereby improving his play; the gazelle calf improves the efficiency with which it can run; Phil learns to streamline his breakfast making. The knowledge the agent brings to the task at the start---either from previous experience with related tasks or built into it by design or evolution---influences what is useful or easy to learn, but interaction with the environment is essential for adjusting behavior to exploit specific features of the task.

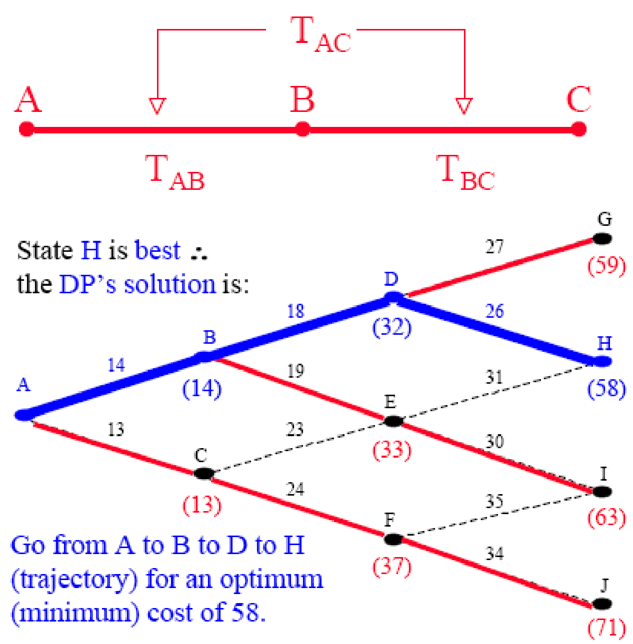


Figure 17. Bellman's principle of optimality: Given an optimal trajectory T_{AC} from A to C, the portion T_{BC} of that trajectory from any intermediate point B to point C must be the optimal trajectory from point B to C. An example of an optimal trajectory (ABDH) identified by dynamic programming by applying this principle is illustrated by the graph.

The term ``optimal control'' came into use in the late 1950s to describe the problem of designing a controller to minimize a measure of a dynamical system's behavior over time. One of the approaches to this problem was developed in the mid-1950s by Richard Bellman and colleagues by extending a 19th century theory of Hamilton and Jacobi. This approach uses the concept of a dynamical system's state and of a value function, or ``optimal return function,'' to define a functional equation, now often called the Bellman equation. The class of methods for solving optimal control problems by solving this equation came to be known as dynamic programming (Bellman, 1957a). Bellman

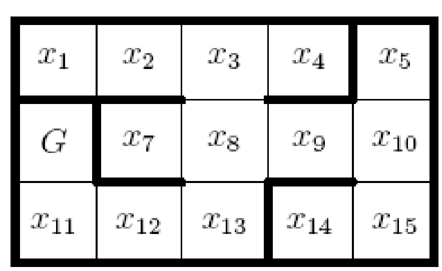


Figure 18. Example of a reinforcement learning task: from any starting state x, move in the maze to reach the goal state G, without crossing the heavy lines, which represent walls

(1957b) also introduced the discrete stochastic version of the optimal control problem known as Markovian decision processes (MDPs), and

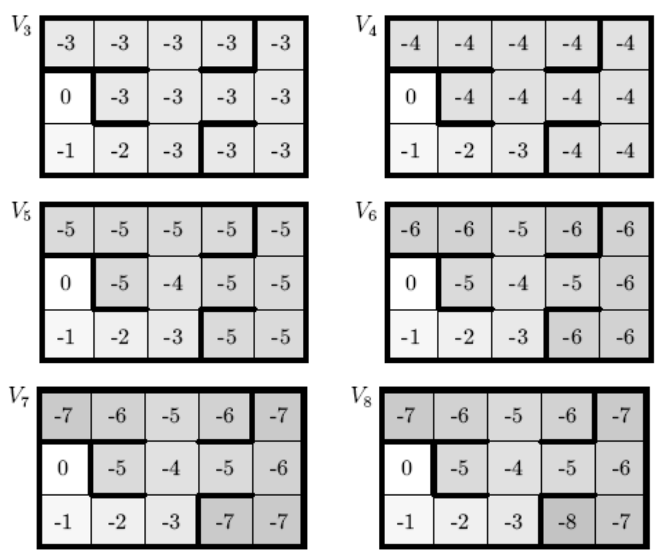
Ron Howard (1960) devised the policy iteration method for MDPs. All of these are essential elements underlying the theory and algorithms of modern reinforcement learning.

Dynamic programming is widely considered the only feasible way of solving general stochastic optimal control problems. It suffers from what Bellman called ``the curse of dimensionality," meaning that its computational requirements grow exponentially with the number of state variables, but it is still far more efficient and more widely applicable than any other method. Dynamic programming has been extensively developed in the last four decades, including extensions to partially observable MDPs (surveyed by Lovejoy, 1991), many applications (surveyed by White, 1985, 1988, 1993), approximation methods (surveyed by Rust, 1996), and asynchronous methods (Bertsekas, 1982, 1983). Many excellent modern treatments of dynamic programming are available (e.g., Bertsekas, 1995; Puterman, 1994; Ross, 1983; and Whittle, 1982, 1983). Bryson (1996) provides a detailed authoritative history of optimal control.

We consider all of the work on optimal control to also be work in reinforcement learning. We define reinforcement learning as any effective way of solving reinforcement learning problems, and it is now clear that these problems are very closely related to optimal control problems, particularly those formulated as MDPs. Accordingly we must consider the solution methods of optimal control, such as dynamic programming, to also be reinforcement learning methods. Of course, almost all of these methods require complete knowledge of the system to be controlled, and for this reason it feels a little unnatural to say that they are part of reinforcement *learning*. On the other hand, many dynamic programming methods are incremental and iterative. Like true learning methods, they gradually reach the correct answer through successive approximations.

Reinforcement Learning (RL) is a general algorithmic approach to solve stochastic optimal control problems by trial-and-error (from Power Systems Stability Control: Reinforcement Learning Framework, Damien Ernst, et al, IEEE Transactions in Power Systems, 19, 427, February 2004).

Figure 19. Application of value iteration in reinforcement learning: the value function is initialized with null values (V_0), and Bellman's equation is applied iteratively until convergence.



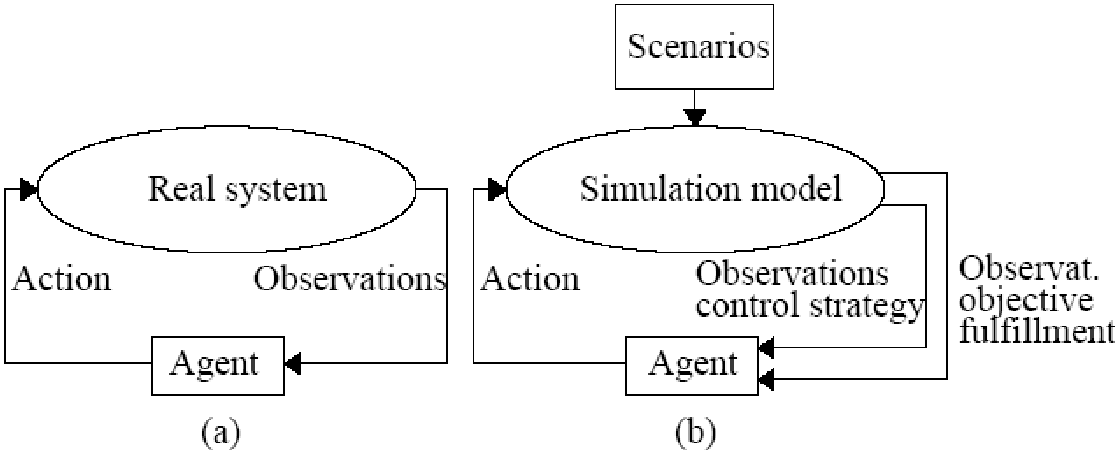


Figure 20. Two modes of application of Reinforcement Learning methods: (a) on-line, (b) off-line

• RL methods do not make any strong assumptions on the system dynamics. In particular, they can cope with partial information and nonlinear and stochastic behaviors. They can therefore be applied to design many, if not all, practical types of control schemes.

• RL methods use closed-loop control laws known to be robust. This aspect is important notably when the real power system is facing situations that were not accounted for in simulation models.

• RL methods open avenues to adaptive control since the RL driven agents learn continuously and can adapt to changing operating conditions or system dynamics.

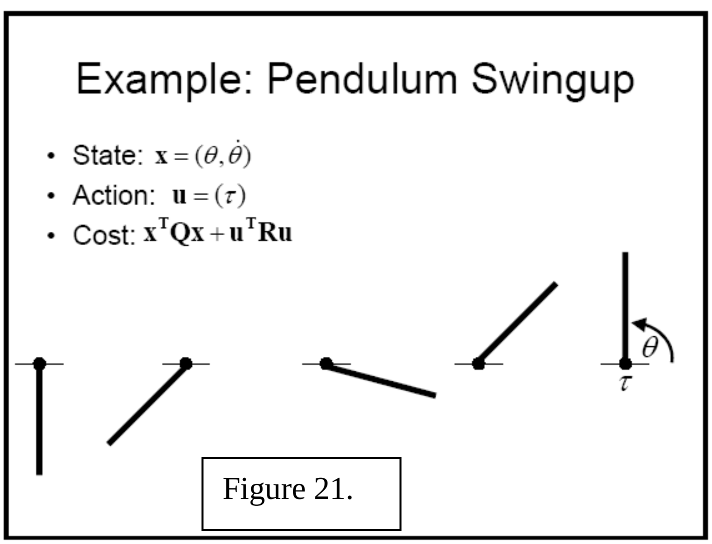


Figure 21.

• RL methods can be used in combination with traditional control methods to improve performance. As an example, they could be used to determine parameters of control policy for a grid device such as a capacitor bank. The RL driven agent does not control directly the device, but rather offers decision support to the transmission operator who actually controls actions on the device.

RL offers a portfolio of

methods that allow agents to learn a goal oriented control law from interaction with a system or a simulator. The RL driven agents observe the system state, take actions and observe the effects of these actions. By processing the experience they accumulate in this way they progressively learn an appropriate control law i.e., an algorithm to associate suitable actions to their observations in order to fulfill a pre-specified objective. The more experience they accumulate, the better the quality of the control law they learn. The learning of the control law from interaction with the system or with a simulator, the goal oriented aspect of the control law and the ability to handle stochastic and nonlinear problems are three distinguishing characteristics of RL.

…. Importance of Dynamic Programming

Reinforcement learning is a difficult problem because the learning system may perform an action and not be told whether that action was good or bad. For example, a learning auto-pilot program might be given control of a simulator and told not to crash. It will have to make many decisions each second and then, after acting on thousands of decisions, the aircraft might crash. What should the system learn from this experience? Which of its many actions were responsible for the crash? Assigning blame to individual actions is the problem that makes reinforcement learning difficult. Surprisingly, there is a solution to this problem. It is based on a field of mathematics called *dynamic programming*, and it involves just two basic principles.

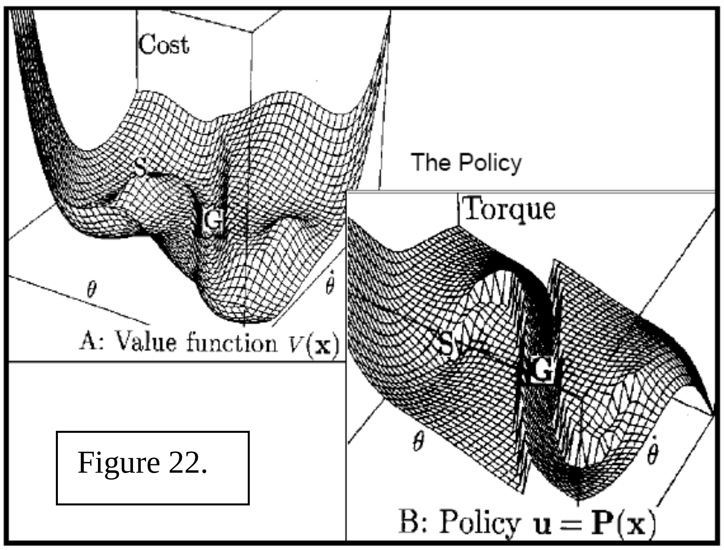


Figure 22.

First, if an action causes something bad to happen immediately, such as crashing the plane, then the system learns not to do that action in that situation again. So whatever action the system performed one millisecond before the crash, it will avoid doing in the future. But that principle doesn't help for all the earlier actions which didn't lead to immediate disaster.

Second, if all the actions in a certain situation lead to bad results, then that situation should be avoided. So if the

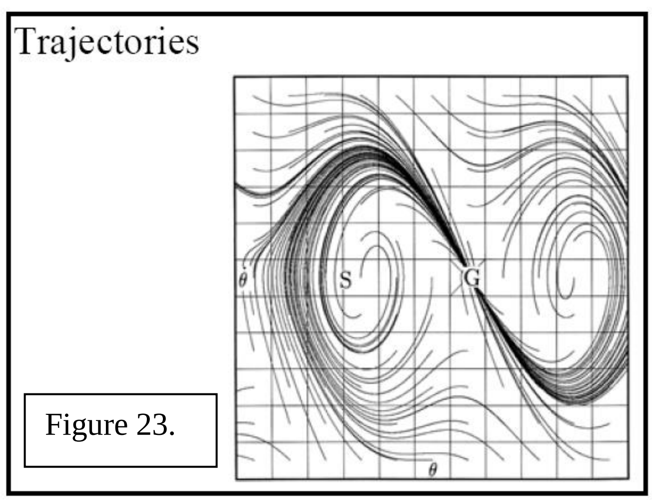


Figure 23.

system has experienced a certain combination of altitude and airspeed many different times, whereby trying a different action each time, and all actions led to something bad, then it will learn that the situation itself is bad. This is a powerful principle, because the learning system can now learn without crashing. In the future, any time it chooses an action that leads to this particular situation, it will immediately learn that particular action is bad, without having to wait for the crash.

By using these two principles, a learning system can learn to fly a plane, control a robot, or do any number of tasks. It can first learn on a simulator, then fine tune on the actual system. This technique is generally referred to as dynamic programming. So, how do we devise an algorithm that will efficiently find the optimal value function?

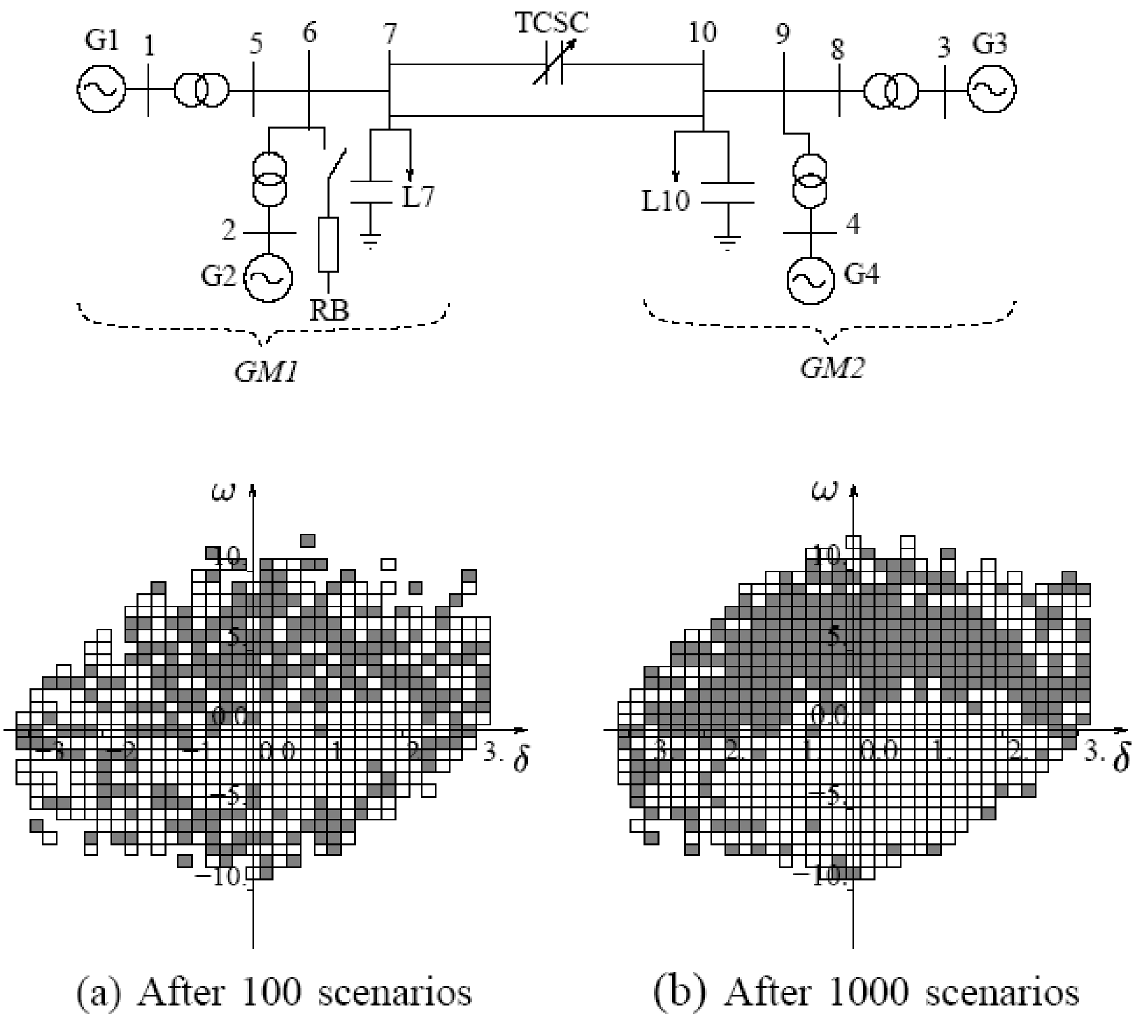


Figure 24. The learned control strategy for a resistive dynamic brake (RB) in a power network with 2 hydro (G1, G2) and 2 thermo generators (G3 G4). δ and ω, the relative angle and speeds of the two classes of generators are the state variables. δ is expressed in *rad* and ω in *rad/sec*. Often the art is finding a representation such as these relative ones that reduce the dimensionality of the problem – from a potential 60 dimensional state space to a two dimensional one. Grey boxes represent states where the brake is applied. From Power Systems Stability Control: Reinforcement Learning Framework, Damien Ernst, et al, IEEE Transactions in Power Systems, 19, 427, February 2004

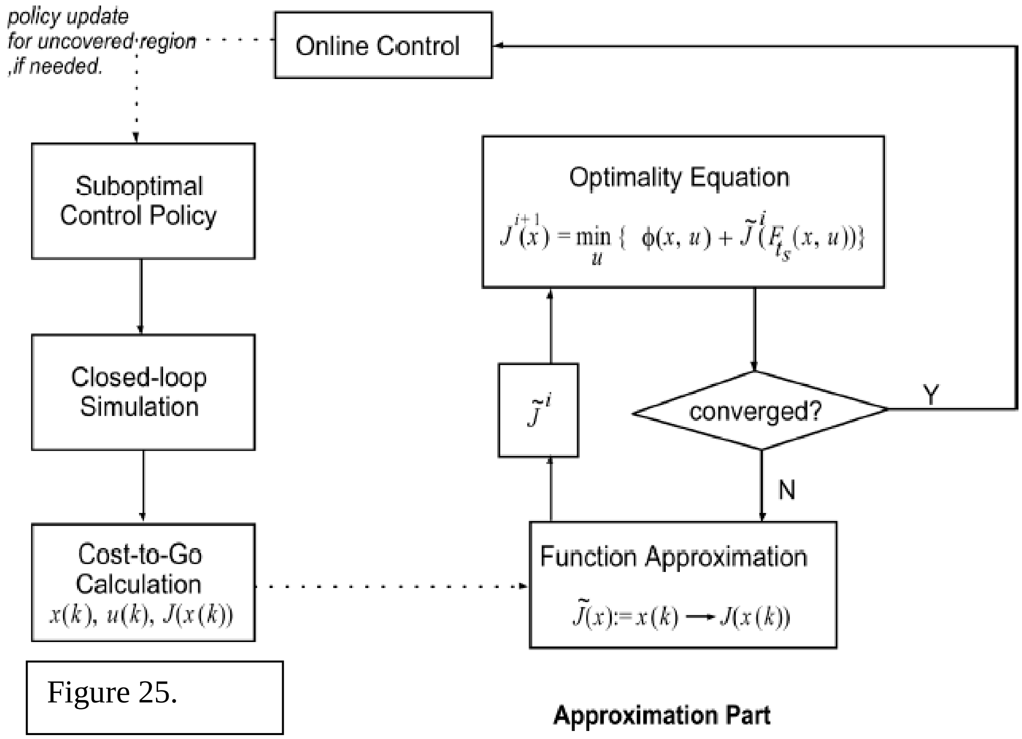


Figure 25.

….. Adaptive Learning with Real Options

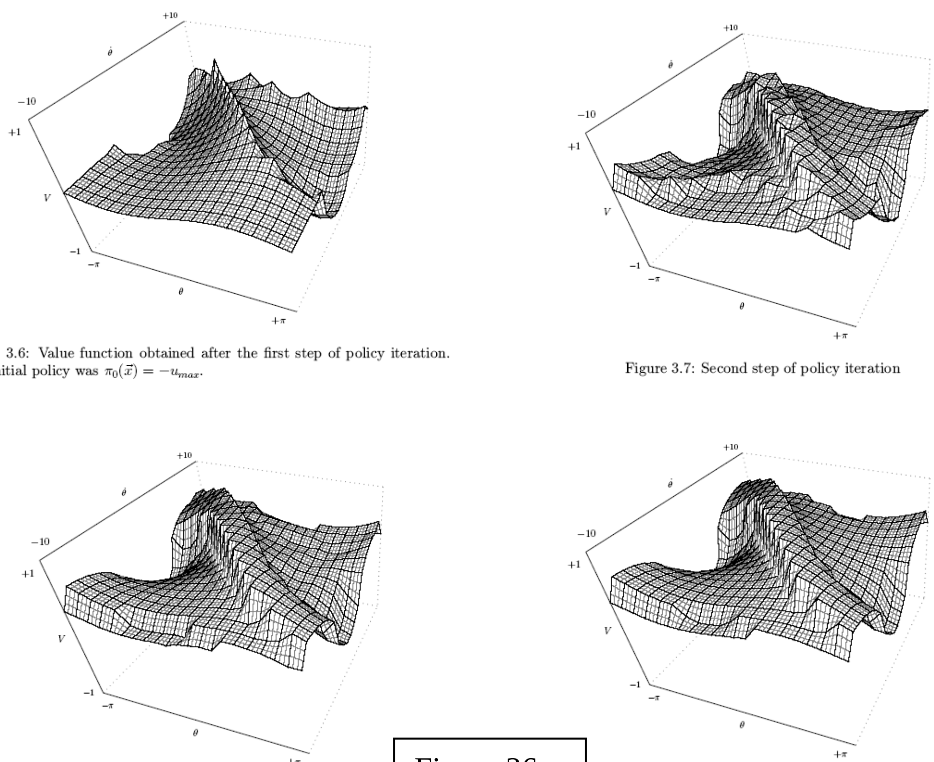


Figure 3.6: Value function obtained after the first step of policy iteration. The initial policy was $\pi_0(\vec{x}) = -u_{max}$.

Figure 3.7: Second step of policy iteration

Figure 3.8: Third step of policy iteration

Figure 3.9: Fourth step of policy iteration

Figure 26.

On the capital investment and long-term strategy side of Decision Support, another Dynamic Programming tool called Real Options adds value to uncertainty and choice-over-time to the more traditional Net Present Value (NPV) technique that discounts options into the future. These added benefits arise from the nature of supply choices, the nature and timing of the demand, the transmission capacity available from adjacent grids, and the specific regulatory characteristics of a market.

In common use, the word *option* is used to suggest alternatives or choices. For real and financial options, the word has a different meaning.

An *option* is the right, but not the obligation, to take an action. For example, an option contract is the right to buy (or sell) a stock at a date specified in the future. Options are valuable when there is uncertainty. For example, an option contract traded on the financial exchanges will be *exercised* (used) only if the price of the stock on that date exceeds the specified price. The value of the contract comes from this upside potential. Real options are created by investment - today's investment creates the right to make a decision later. The value of the investment includes these real options.

Real options is the extension of financial option theory to options on real (nonfinancial) assets. In contrast to the valuation of financial options --where decision-making it is a matter of shopping for the best deal on a specified contract -- the valuation of a real option requires that it be identified and specified. Moving from financial options to real options requires a way of thinking, one that brings the discipline of the financial markets to internal strategic investment decisions.

The real options approach works because it helps managers with the opportunities they have to plan and manage strategic investments. Stewart Myers of MIT coined the term "real options" to address the gap between strategic planning and finance. "Strategic planning needs finance. Present value calculations are needed as a check on strategic analysis and vice versa. However, standard discounted cash flow techniques will tend to understate the option value attached to growing profitable lines of business. Corporate finance theory requires extension to deal with *real options*." (Stewart C. Myers, Sloan School of Management, MIT (1984), p. 13).

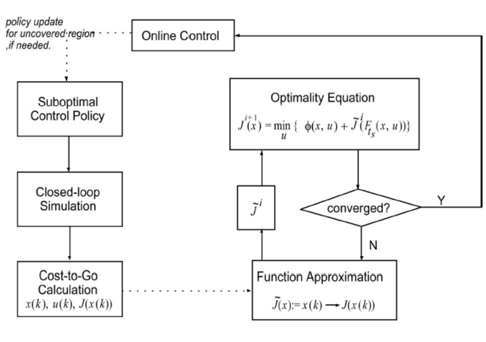


Figure 27. Surprise! This RL architecture is also an architecture for simulation-based real option valuation.

Real options value is estimated with ranges of possible outcomes for waiting-to-invest, growth, flexibility, exit strategies, and learning-over-time options. Pricing of real options on energy is fast becoming essential in budgeting analysis for the construction of any large capital investment. The capital cost is evaluated by estimating the value over time of a strip of call options, where the exercise price is the marginal operating cost at which the equipment is offered into the supply curve (merit stack). If there is no variation in prices, there will be no opportunity to earn economic quasi-rents equal to the difference between market price and marginal cost. It is the value of these quasi-rents (options) that pays for the capital value of the plant. Thus, understanding the market price process is essential in the real option analysis. It is also important in the more obvious field of risk management and hedging, and critical here because we are evaluating the capital costs of products that store energy when demand is low and provides energy when need, and thus when price is high. The uncertainty of the market produces unrealistically constrained valuations using the classical NPV model. Specifically, subjective inputs are replaced by objective evaluations from project analysis, including a range of outcomes.

Data Required for Real Options Analysis

From The Real Power of Real Options, McKinsey Quarterly, 3, 1997.

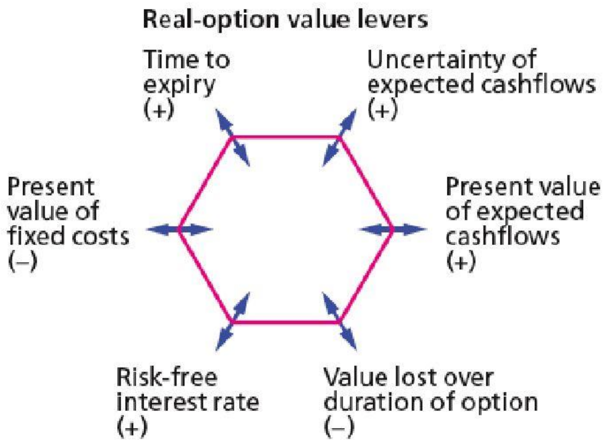


Figure 28.

A crucial aspect of real options is that most of the input data can be objectively verified. From http://www.real-options.com/overview_faq.htm, the following inputs are the only information you need to value a real option:

- *The current value of the underlying asset*, which is observed in the market.
- *The time to the decision date*, which is defined by the features of the investment.
- *The investment cost or exercise price* (also called the strike price), which is defined by the features of the investment.
- *The risk-free rate of interest*, which is observed in the market.
- *The volatility of the underlying asset*, which is often the only estimated input.
- *Cash payouts or non-capital gains returns to holding the underlying asset*, which are often directly observed in the market, or sometimes estimated from related markets.

Information that is *not* needed to value a real option contributes greatly to its power.

- *Probability estimates* are not needed because these are captured by the current value of the underlying asset and the volatility estimate.
- *An adjustment to the discount rate for risk* is not needed because the valuation solution is independent of anyone's taste for risk.

The expected rate of return for the underlying asset is not needed because the value of the underlying asset and the ability to form tracking portfolios already captures its risk/return tradeoff.

Option	Description	Type of Flexibility	Guide to Literature
Deferral	Similar to an American Call option. Exists when management can defer the decision about the investment for a certain period of time. They are important in natural resource extraction industries, real estate development, farming and others.	Upside Potential	McDonald and Siegel (1986); Paddock, Siegel and Smith (1988); Tourinho (1979); Titman (1985); Ingersoll and Ross (1992); Dixit (1992)
Timing or Staging	Relates to the possibility of staging investments as a series of outlays to create both growth and abandonment options. Each stage can be viewed as an option on the value of subsequent stages (compound option). They are important in R&D intensive industries, capital-intensive projects and start-up ventures.	Upside Potential and Downside Protection	Brennan and Schwartz (1985); Majd and Pindyck (1987); Carr (1988); Trigeorgis (1993)
Altering Operating State	If market conditions are better than expected, a company may decide to increase its output level by investing in scaling-up the production plant either temporarily or permanently. Equally, if market conditions are adverse the firm might decide to temporarily shutdown production. Both cases are similar to call options. Important in natural resources industries where prices of output may vary constantly, commercial real estate, and in other cyclical industries such as fashion apparel and consumer goods	Upside Potential and Downside Protection	Brennan and Schwartz (1985); McDonald and Siegel (1985); Trigeorgis and Mason (1987); Pindyck (1988)
Growth	A growth option is similar to a European or American call. They exist when early investments in R&D, undeveloped land or reserves of a natural resource, information, create the opportunity of generating further revenues (I.e., developing a product and selling it in the market, exploiting the acquired reserves, and others). Growth opportunities are compound options, whose value depends on a pre-existing option.	Upside Potential	Myers (1977); Brealey and Myers (1991); Kester (1984, 1993); Trigeorgis (1988); Pindyck (1988); Chung and Charoenwong (1991)
Abandon ment	Similar to an American Put. If market conditions deteriorate, management can abandon current operations permanently and recoup the salvage value of the asset. It is important in capital-intensive industries with second-hand markets for their assets, such as the airline industry, railroads and financial services.	Downside Protection	Myers and Majd (1990), Sachdeva and Vanderberg (1993)
Switching	A combination of calls and puts that allow its owner to switch between two or more modes of operation, inputs or outputs. These options can create both product flexibility and process flexibility. They are important in facilities that are highly dependent on an input whose price varies constantly (E.g., oil, or any other commodity), and consumer electronics, toys, and autos industries where product specifications are subject to volatile demand.	Upside Potential and Downside Protection	Magrabe (1978); Kensinger (1987); Kulatilaka and Trigeorgis (1993)

Table 6. **Real Options (adapted from Trigeorgis, ed., 1995, pp. 4-5)**

A differentiator to the real option evaluation approach is the ability to simulate or model the system by using parameterized engineering models being driven by both engineering, environmental, and financial uncertainties and allowing optimal investment and operating decisions. These include a detailed stochastic model for the price and cost processes as well as the representation of technical (engineering) risks separately. The jumps in price process and associated jump volatility will occur due to a variety of factors including but not limited to grid congestion, environmental factors and reliability events. The analysis also allows us to create optimal operating rules to maximize value -- addressing a long standing issue in realizing the real options value in actual operation. Unlike common real option valuation methods such as binominal, our approach using approximate dynamic programming is non-parametric. It directly samples the possible paths via simulation instead of first building a parametric model of the distributions. The promise is that not only economic interactions, but also engineering and environmental interactions can be incorporated and policies enacted to avoid downside outcomes.

One other advantage of the approximate dynamic programming with simulation approach in systems that require technical design is that real options in technical designs should differ from those that treat the technical systems as "black boxes". It is useful to distinguish between options "in" and "on" systems -- between those that do and do not deal with design elements. The valuation of real options "in" and "on" systems should differ, because the specifics of the technical system may mean that the financial assumptions used to calculate option values may not apply (de Neufville 2004). We can simulate both the baseline and the new design operating characteristics that are not market related. Such processes will have an impact on reliability and failure rates (and associated technical risks in real options analysis). Pairing a power flow simulator and Real Options add on, we can combine the stochastic operating conditions of the system (non-market related) with real options valuation. Such an approach will capture both the technical as well as market uncertainties in a holistic way as well as allow optimal decisions considering all aspects of the decision process.

Real option values the following aspects of capital investment and operations:

- option value of capital investments in terms of development and deployment of new technology,
- option value of arbitrage,
- option value of supply and demand management,
- option value of greater network reliability &,
- option value of environmental benefits and (negatively) drawbacks

Each of these applications of real options can be represented in a decision options tree showing investments needed, expected timing, operating costs, technical risks and benefits derived. Benefits derived are represented as a function of the price process that is modeled in detail. A decision tree construct is used to represent the time to first operation as well as the time of operation during the life of the asset being valued. Actual time depends on demand, supply and price characteristics. Real Options software is capable of analyzing the optimal timing of operation based on the modeled (and expected) price processes.

Finally, the value of the real option is calculated from the cash flows created from the price, costs and timing of operation for the various applications under consideration. This can be viewed as a series of independent options to operate the asset or as ‘swaptions’ based on the conditions and needs. Additionally, the upside potential and downside risk of each application can also be calculated and represented. The value from the portfolio of applications can then be combined to calculate the overall value of the proposed asset deployment.

Chapter 7: Use of Matrices in CALM

A concern we have heard repeatedly is “We seem to have many, but not all of the Lean principals and processes you write about. What should we do next to close the loop?” Below we describe an appropriately “Lean” process to identify the Business Capabilities where Lean principles will have the most value through a formal analysis methodology focused on how well technology, processes, and organizational interrelationships are being integrated in projects within your company (Figure 1).

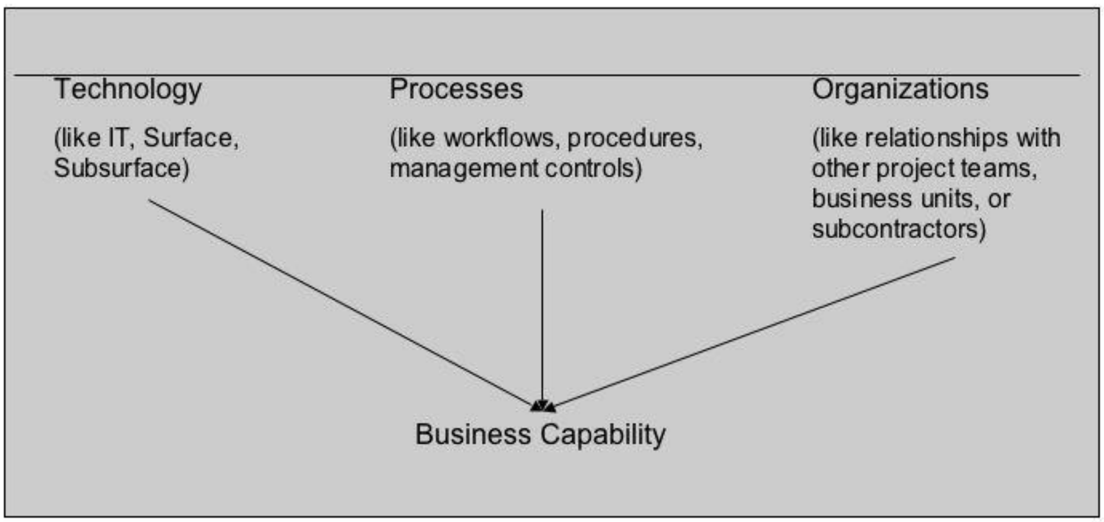


Figure 1. From: http://www.eller.arizona.edu/~finhome/lam/535/mt1.ppt

We call this migration to Lean processes and tools “Computer Aided Lean Management”, or CALM, which sounds complicated. Instead, it all starts with rather simple matrices. Matrices are powerful tools for evaluating the relationships between variables in the familiar form of rows versus columns (as in a spreadsheet). Matrices are indeed simple on first look, but chain them together and you get neural networks or rule-based expert systems. These, in turn, are then superseded by even more advanced machine learning tools that force integrated connectivity and quantitative rigor. They provide an effective methodology for defining how a company can migrate to Lean Energy Management. Furthermore, they offer a solid path to the oil patch becoming an adaptive enterprise as real time data begins to stream into field control rooms all over the world.

Quality Function Deployment

One area of Lean Management that has developed extensive use of chained matrices for Lean evaluation purposes is termed Quality Function Deployment (QFD). (More information on QFD is available in the online version of this paper at www.ogjonline.com). QFD is widely used to develop implementation strategies for new or redesigned products. It forces the Lean alignment of technology development with internal processes and customer needs. Obviously getting the product right is a very risky enterprise, and thus the need for Lean rigor.

A QFD Implementation Matrix plots problems as rows against columns that provide solutions common to successful projects (Table 1). QFD drives the user to document and list everything. In addition to chained matrices, the basic tools of QFD are the project roadmap and these documents and lists. A project roadmap defines the flow of data through a QFD project. Documents are required to record all background information for the project. Lists form the input rows and output columns of the matrices. Examples of lists include: user benefits, measures, basic expectations, functions, and alternative concepts. Lists generally have related data associated with them. For example, the priorities and perceived performance ratings resulting from market research can be associated with the list of benefits. Importance values are associated with measures and functions.

QFD Problems to solutions Matrix							
	Practical Solutions for applying QFD						
		Demands on company resources	Select team members carefully	Involve Sr management all along the way	Trust your Intuition about whether results are right	Be Flexible, Build to your company's needs	Limit cross-functional conflicts
misinterpretations		limit size of initial matrices to 8x8 max			Be sensitive to classified and hard to get information	Document issues raised and adjust matrices as appropriate	
	mixing technical measures with customer requirements		chose members with closest links to customer, whatever it is		check with intuition of team		usually shows up first here
	not properly sorting the interview data		allow individual team members to enter matrix info from interviews individually, but review as group regularly		check with intuition of team		usually shows up first here
	interpreting chained matrices as serial developments				check with intuition of team		
time constraints		So limit the length and number of meetings					
	training	train in teambuilding if function not used to it		resources must be provided as needed	have team evaluate its own training needs at first		train about silo-busting criticality
	facilitation		make sure same definitions of tasks and terms understood by all	resources must be provided as needed	facilitator should be pro with no vested interest in outcomes		
	market research		don't read into the VOC what you think customer will say.	resources must be provided as needed	expect the unexpected, but add follow-on investigation to determine if enexpected is as it appears at first.		
resource constraints	demands on key functional represeantatives	limit to small teams at first	Identify key relationships early	resources must be provided as needed			educating other silos important to consensus building
	matrix building				Matrices are live documents that must be continually updated with new learnings for all to see.		
	process requires hard thinking	So limit the length and number of meetings					

Table 1. QFD Implementation Matrix adapted from Lowe and Ridgeway, 2000.

A matrix is simply a format for showing the relationship between two lists, and thus a matrix deploys, or transfers, the importance from the input list to the output list. For example, a common matrix relates "performance measures" to "user benefits." Another matrix can then be created to relate “user's benefits” to “product options.” A chain of matrices has been created. See the below website for an excellent animation of how a QFD product identification plan chains matrices together: http://www.gsm.mq.edu.au/cmit/hoq/QFD%20Tutorial.swf

Matrices provide very powerful tools for representing, mapping and modeling Lean principals in terms of existing organizational structures, processes, and expert knowledge comprising the Business Capabilities being studied. For example, rule based expert systems, decision tables, and neural networks can be represented using matrices (Figure 2).

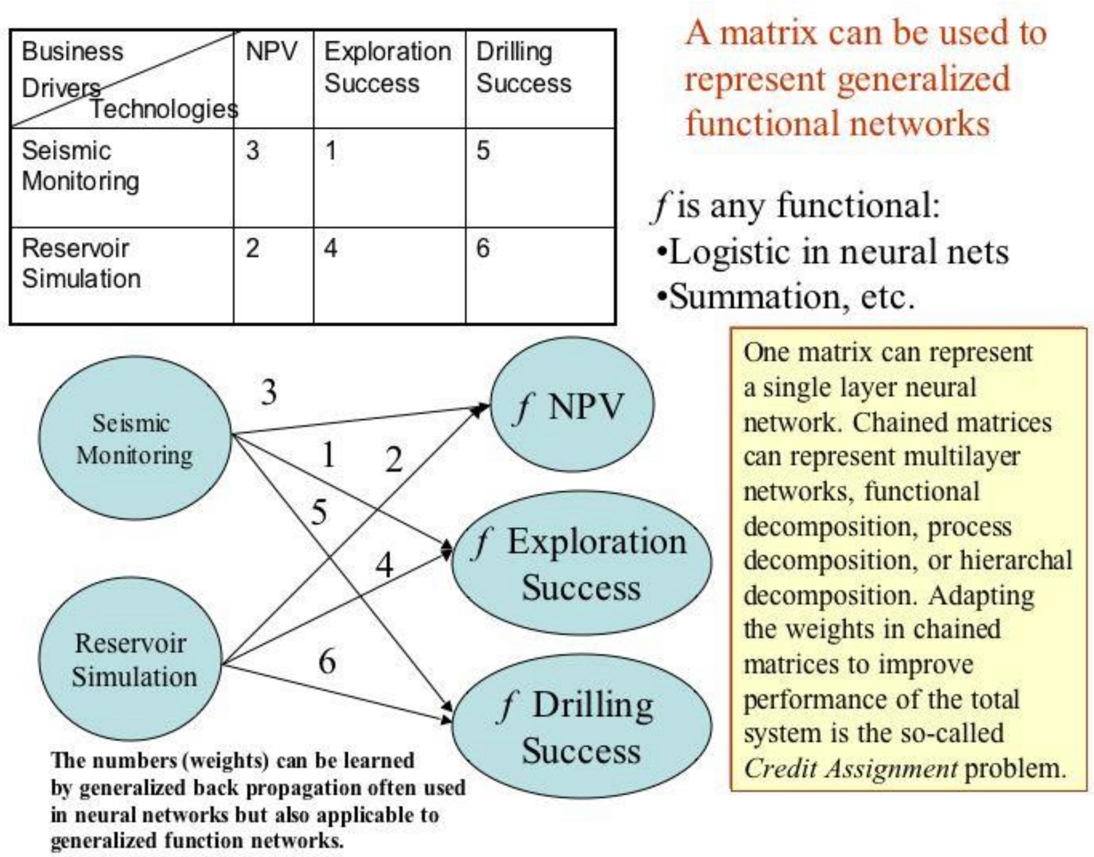


Figure 2.

The Pugh Matrix

In QFD matrices, indicators are associated with each measure and function. The indicators placed in each cell of the matrices can be as simple as a binary “yes” or “no” answer, or as quantitative as weights, probabilities, or confidence scores that are real numbers. Often, an evaluation begins with binary indicators, which are then replaced with numbers to make the mapping a more accurate representation of the process being modeled. Even probabilistic or “fuzzy” processes can be represented with error estimates to incorporate an additional measure of uncertainty.

The “Pugh Matrix” (Figure 3) is a part of Stuart Pugh’s Total Design methodology, and it determines which potential solutions are more important or 'better' than others to solve any give problem. The Pugh Matrix is used for concept selection in which options are assigned scores. The Pugh Matrix contains evaluation criteria (rows), plotted against alternative product variations (columns), and a weighting of the importance of each criteria is placed into each cell. Scores from 1-10 for each alternative are then “weighted” for overall system importance to derive a total score. The final design selection is made based on the highest consolidated score. The company is forced to consider many options so that they must choose the best among many. It can only be used after the Voice of the Customer (VOC) has already been captured, which in Lean nomenclature means after product planning, but before the design phase begins.

Criteria	**Weight**	Alternative A	Alternative B	Alternative C
Number 1	20	8	4	7
Number 2	30	4	7	5
Number 3	10	5	6	2
Number 4	40	2	4	6

Calculate the average score for each alternative by multiplying the score by the weight and dividing by the total of the weights which in this case is 100.

Figure 3. From: http://www.rpi.edu/~castil2/hand1.html

Chained Matrices

A key to the use of matrices for Lean evaluation is to systematically map processes within the Business Capabilities of an enterprise by chaining, or cascading multiple matrices. For example, chained matrices are a vehicle for mapping the complex cause-to-effect relationships from symptoms to problems and then from problems to solutions among technologies, processes, and organizational boundaries that compose the many levels of every project, as illustrated in Figure 4. Another example of chaining two matrices like this is the decision table from decision theory (above right in Figure 4). The methodology of chained matrices is to make the vertical columns of the first matrix become the horizontal rows of the second, and so on through the series. As we will encounter in the next section, these chains can be formed in multiple directions, depending on the dimensional links. How the matrices are linked is guided by the QFD roadmap. As an example, a user can enter the decision table with a symptom, then define the most likely problem, and select among several solutions in the last of the chained matrices.

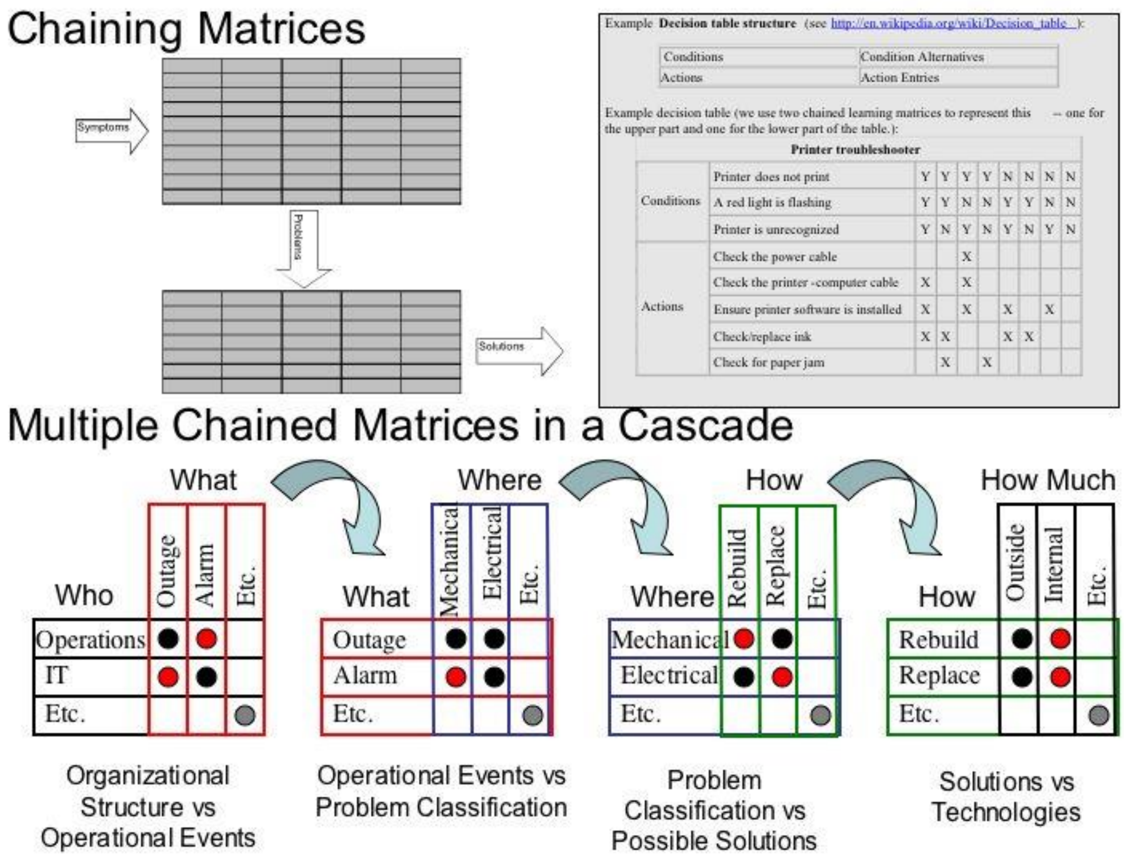


Figure 4.

The House of Quality

The House of Quality (HOQ), which has become synonymous with QFD, is a technique for chaining multiple matrices used in QFD product development or redesign to ensure that the customer's wants and needs are the basis for the improvement (Figure 5, top, from Karsak et al, 2002). HOQ is a highly structured approach that starts with customer surveys to establish the Voice of the Customer (VOC) and ends with detailed engineering solutions to design requirements based on the VOC. The Expanded House of Quality (Figure 5, bottom) chains even more matrices together, and depending on the complexity of the QFD project roadmap, the "rooms" of this expanded house will change. There are three common "blueprints" or templates, available for laying out HOQ rooms:

1. The American Supplier Institute Four Phase Approach of cascading houses
2. The Expanded House of Quality of Figure 5, bottom, developed by International TechneGroup Incorporated (producers of QFD/Capture)
3. The Matrix of Matrices methodology developed by GOAL/QPC which is likened to the modular design of homes (c.f. King, 1989 & http://www.GOAL/QPC.org)

These methods are included in QFD software such as QFD/Capture http://www.qfdcapture.com & Qualica QFD and http://www.qualica.de

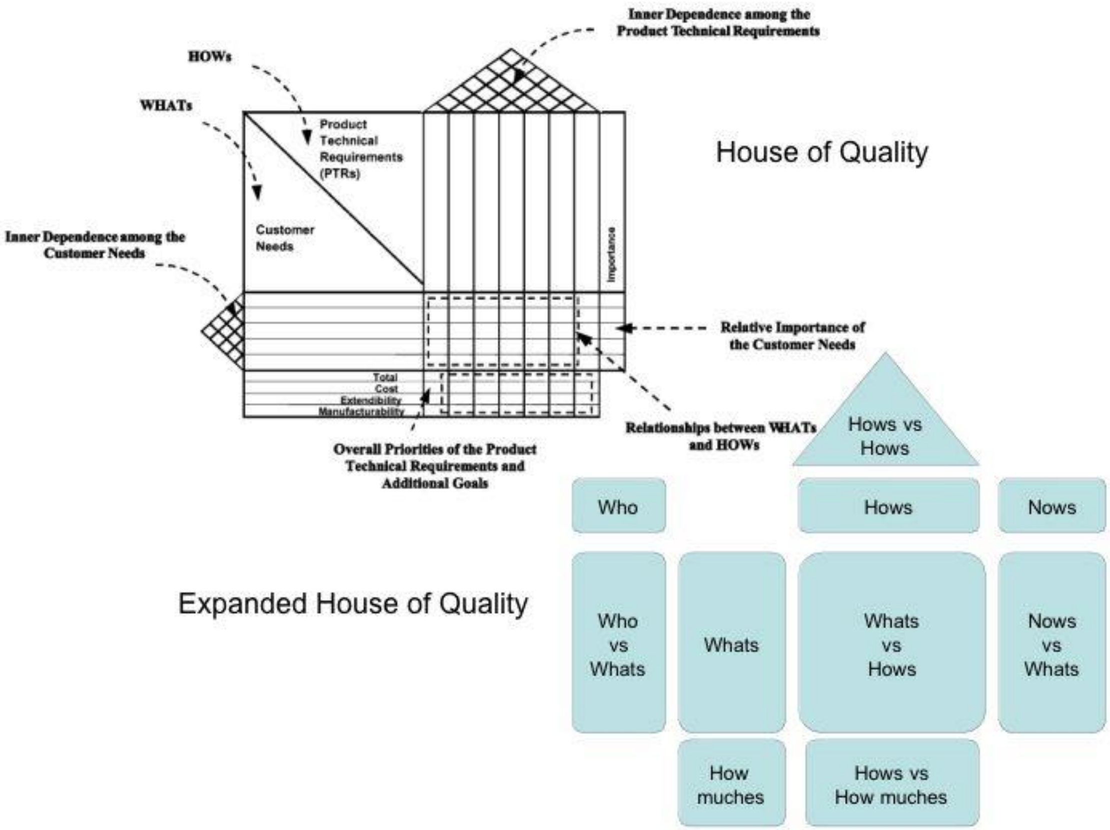


Figure 5 From: http://mielsvr2.ecs.umass.edu/virtual_econ/module2/HOQ_Frames.htm & Karsak, et al., 2002.

The Roof of the House of Quality

Like in many other industries, we have a general need to represent "common mode" interactions and failures. In the HOQ, these are represented by a row (column) of a matrix that interacts in either a beneficial or antagonistic way with another parallel row (column). These interactions occur in relatively few situations. If this is the case, then separate entries representing the synergy can be added to the matrix. If, however, there are many interactions, then this approach becomes unwieldy and another approach must be used. Borrowing from neural network theory, a two-layer, chained matrix network can represent such interactions. Thus the symptoms->problems matrix in Figure 4 can be replaced with two matrices if there are common mode interactions amongst the systems. Likewise the problems->solutions matrix can be replaced with two matrices if there are interactions amongst the solutions.

The roof of the House of Quality is an example of this later method of a matrix used to capture synergies and antagonisms of possible common mode failures that are at the end of the HOQ matrix chain (see the roof above the what's->how's matrix in figure 5). These relationships are symmetric, so only half a matrix is needed – the 60º triangle roof is used to represent this half matrix. An example of the "main rooms" of the HOQ is given for the design of a new fountain pen product (Figure 6). Note the roof of the House of Quality, where the + represents synergy between two columns of the "How" matrix, and the – represents conflict between the diameter to weight trade-off in those "How" columns. Also notice the side "roof" that captures the interactions of the "What" rows represents interactions among the customer needs.

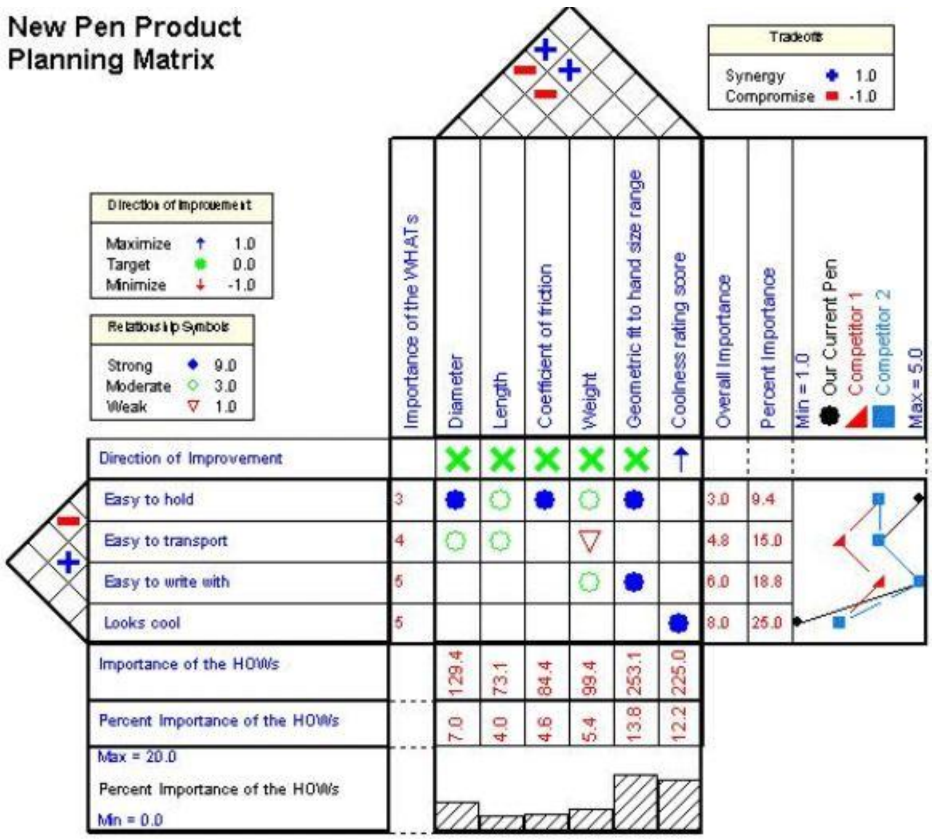


Figure 6. From the QFD/CAPTURE website http://www.qfdcapture.com/

Chapter 8: Closing the Feedback Loop

A Lean Management System continually seeks perfection in performance. This aggressive learning of improvement (termed Kaizen by Toyota) requires feedback loops that are a key concept of Lean. The goal of Lean Energy Implementation is to reach Kaizen through rigorous enforcement of feedback loops that first predict outcomes and then make corrections based upon objective scoring of the predictions versus actual events (Figure 7). Again, the Lean process is solidly footed in theory. Derivatives calculated by the chain rule of basic calculus are the source of the steering signals used for optimization via a generalized "back-propagation" learning method from, again, neural network theory, which allows the feedback loop to be closed. Computer Aided Lean Management (CALM) is the rigorous enforcement of the feedback loop using software to automatically track all operator actions, score of outcomes of those actions, and back-propagate corrections of the numeric values in the matrices to the system performance model that optimizes performance from such feedback loops.

In our experiences with many Lean implementations, most of the tools and methods of Lean can be in place at energy companies, but this feedback loop is usually the one critical piece that is missing. Companies must implement an automated way to track actions, measure performance, and rigorously adjust the system to then improve performance. People alone are not enough.

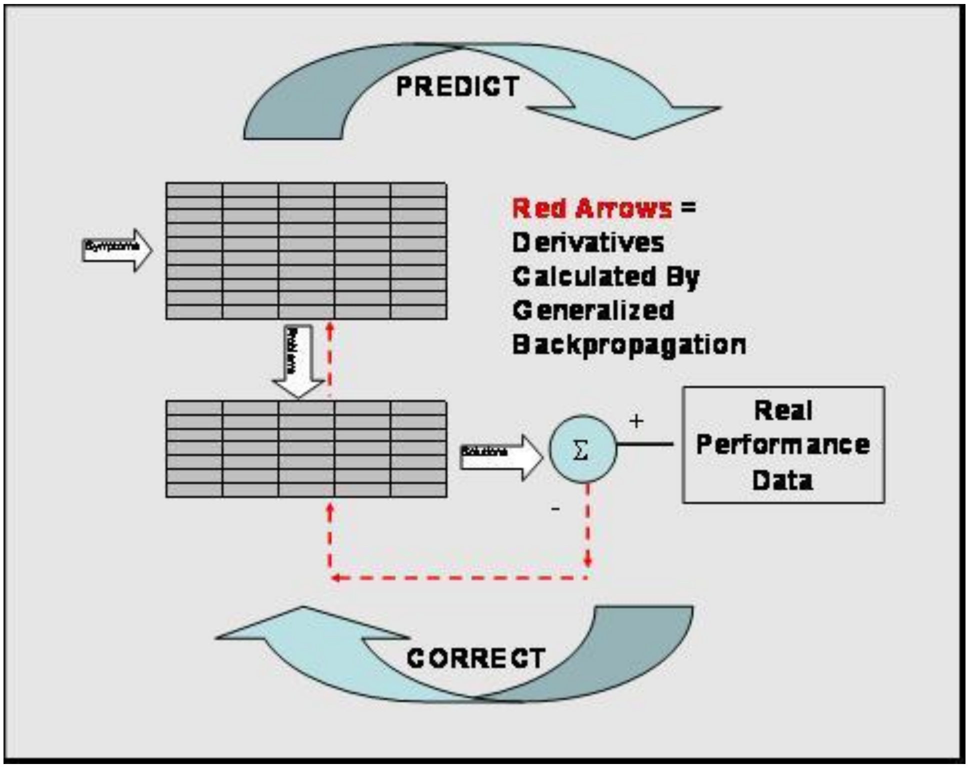


Figure 7.

The Balanced Scorecard is an example of a way many energy companies improve human performance, but it does not improve the linkage among human, machine, and computer models in the unified, integrated way required by Lean Energy Management. In contrast, CALM uses ideas from stochastic control, option theory, and machine learning to build a software support system that forces the optimization of business and engineering objectives simultaneously under uncertainty.

History of QFD

QFD was developed in Japan in the late 1960s by Professors Shigeru Mizuno and Yoji Akao. At the time, statistical quality control, which was introduced after World War II, had taken roots in the Japanese manufacturing industry, and the quality activities were being integrated with the teachings of such notable scholars as Dr. Juran, Dr. Kaoru Ishikawa, and Dr. Feigenbaum that emphasized the importance of making quality control a part of business management, which eventually became known as TQC and TQM.

The purpose of Professors Mizuno and Akao was to develop a quality assurance method that would design customer satisfaction into a product *before* it was manufactured. Prior quality control methods were primarily aimed at fixing a problem during or after manufacturing.

The first large scale application was presented in 1966 by Kiyotaka Oshiumi of Bridgestone Tire in Japan, which used a process assurance items fishbone diagram to identify each customer requirement (effect) and to identify the design substitute quality characteristics and process factors (causes) needed to control and measure it.

In 1972, with the application of QFD to the design of an oil tanker at the Kobe Shipyards of Mitsubishi Heavy Industry, the fishbone diagrams grew unwieldy. Since the effects shared multiple causes, the fishbones could be refashioned into a spreadsheet or matrix format with the rows being desired effects of customer satisfaction and the columns being the controlling and measurable causes.

At the same time, Katsuyoshi Ishihara introduced the Value Engineering principles used to describe how a product and its components work. He expanded this to describe business functions necessary to assure quality of the design process itself.

Merged with these new ideas, QFD eventually became the comprehensive quality design system for both product and business process.

From http://www.qfdi.org/what_is_qfd/history_of_qfd.htm

Expanded House of Quality

From: http://www.proactdev.com/pages/ehoq.htm

As seen in Figure 4, this "Expanded House Of Quality" consists of multiple "rooms." Four of the rooms form the basic axes of the house. These are lists of "WHATs", "HOWs", "WHYs", and "HOW MUCHes". Four of the rooms consist of relationships between these lists. A brief explanation of each room follows.

WHATs - This is a list of what the customer wants or what is to be achieved. When the "Expanded House of Quality" is used with end user requirements, these would be customer statements about what they want to see in the product. Hint: A common problem is that a lot of customers tend to state their requirements in terms of a possible solution. It is important that you understand the true requirement rather than accepting customer statements at face value.

HOWs - This is a list of what your company can measure and control in order to ensure that you are going to satisfy the customer's requirements. Typically, the entries on this list are parameters for which a means of measurement and a measurable target value can be established. Sometimes HOWs are also known as Quality Characteristics or Design Requirements. Hint: It is best to try to keep these entries as concept-independent as possible. Failure to do this will lock you into a particular design solution that will almost never be what you would arrive at if you do QFD correctly. For example, if you were developing the lock for a car door you might be tempted to define HOWs such as "Key insert force" and "Key turn torque". These both imply that the lock will be key actuated. You will have immediately eliminated concepts such as combination locks that might have security and cost advantages for your particular application. A better HOW might be "Lock/Unlock Work" which could be measured for both key operated and combination locks.

WHYs - Conceptually, this is a list that describes the current market. It is a way of explaining why this product needs to exist. It indicates what data will be used to prioritize the list of WHATs. Commonly included are lists of the customer groups your product must satisfy and their importance relative to each other. Also included are lists of products that will compete with yours in the marketplace.

HOW MUCHes - This list is used to specify how much of each HOW is required to satisfy the WHATs. Commonly it contains a listing of the products on which testing will be performed. This testing helps establish realistic target values for the HOWs. It also includes entries where the priority of each of the HOWs can be established. In general, WHYs and HOW MUCHes are very similar. WHYs lead to the importance of the WHATs while HOW MUCHes document and refine the importance of the HOWs.

WHATs vs. HOWs - This is a relationship matrix that correlates what the customer wants from a product and how the company can meet those requirements. It is the core matrix of QFD. Relationships within this matrix are usually defined using a strong, medium, weak, or none scale. If a HOW is a strong measure of compliance with a WHAT, then the WHAT and HOW are strongly correlated. Similarly, if a HOW provides no indication as to whether your product complies with the WHAT, there is probably no relationship. Filling and analyzing this matrix will likely take a large portion of the time you spend in QFD meetings.

WHATs vs. WHYs - This is a relationship matrix that is used to prioritize the WHATs based upon market information. Usually, the data in this matrix consists of ratings on how important different customer groups perceive each of the WHATs to be. Ratings of how well competitive products are perceived to meet each of the WHATs can also be included here. Averaging the stated importance ratings and factoring in where your product is perceived relative to your competition helps establish the overall importance of each WHAT.

HOWs vs. HOW MUCHes - This is a relationship matrix that helps you decide what the next step in the project should be. Typically, this matrix includes calculated values which identify the relative importance of each of the HOWs. Also included is information about how your competition performs relative to each of the HOWs. This information can lead you to establish realistic and measurable target values which, if met, will ensure that you meet the customer's requirements.

HOWs vs. HOWs - This matrix forms the roof of the "Expanded House of Quality" and gives it its name. It is used to identify the interactions between different HOWs. The relationships in this matrix are rated as Strong Positive, Positive, Negative, Strong Negative, and None. If two HOWs help each other meet their target values, they are rated as Positive or Strong Positive. If meeting one HOW's target value makes it harder or impossible to meet another HOW's target, those two HOWs are rated with a Negative or Strong Negative relationship.

Further Reading

Columbia University CALM References:
Anderson, R. and A. Boulanger,, 2003. Lean Energy Management I: Lean Energy Management Required for Economic Ultra Deepwater Development, Oil & Gas Journal, March 17, 2003,
Anderson, R. and A. Boulanger, Lean Energy Management II: Ultradeepwater Development: Designing Uncertainty into the Enterprise, Oil & Gas Journal, May 19, 2003.
Anderson, R. and A. Boulanger, Lean Energy Management III: How to Realize LEM Benefits in Ultradeepwater Oil and Gas, Oil & Gas Journal, June 30, 2003.
Anderson, R. and A. Boulanger, Lean Energy Management IV: Flexible Manufacturing Techniques make Ultradeepwater Attractive to Independents, Oil & Gas Journal, August, 25, 2003.
Anderson, R. and A. Boulanger, Lean Energy Management V: Enterprise-wide Systems Integration Needed in Ultradeepwater Operations, Oil & Gas Journal, November 24, 2003.
Anderson, R. and A. Boulanger, Lean Energy Management VI: Ultradeep Offshore Suitability Matrix for Estimating Value of Lean Processes, Oil & Gas Journal, June 28, 2004.
Anderson, R. and A. Boulanger, Lean Energy Management VII: Knowledge Management and Computational Learning for Lean Energy Management, Oil & Gas Journal, November 24, 2004.
Anderson, R. and A. Boulanger, Lean Energy Management VIII: Use of Matrices in Computer-Aided Lean Energy Management, Oil & Gas Journal, March 7, 2005.
Anderson, R. and A. Boulanger, Lean Energy Management IX: Boosting, Support Vector Machines and Reinforcement Learning in Lean Energy Management, Oil & Gas Journal, May 9, 2005.
Anderson, R. and A. Boulanger, Lean Energy Management X: How Martingale stochastic control navigates computer-aided lean energy management", Oil & Gas Journal, September 19, 2005.
Anderson, R., A. Boulanger, Wei He, U. Mello, L. Xu, Lean Energy Management XI: 4D Seismic Reservoir Monitoring with CALM's Martingale Controller, Oil & Gas Journal, March 31, 2007.
Anderson, R. A. Boulanger, Lean Energy Management XII: Real Options of Gas-to-Electricity in Ultra Deepwater Production, Oil & Gas Journal, July 23, 2007.
Anderson, R., Technical Innovation: An E&P Business Perspective, , SEG Leading Edge, June, 2000.
Anderson, R. et.al, Power Shortage, CIO Insight, July, 2001.
Anderson, R. and W. Esser, Energy Company as Advanced Digital Enterprise, American Oil & Gas Reporter, January, 2001.
Esser, W. and R.N. Anderson, Visualization of the Advanced Digital Enterprise, Offshore Technology paper OTC 13010, 2001.
Anderson, R., A. Boulanger, ThreatSim: Securing Wattage When Needed, Energy Pulse, Oct 14, 2002.

176. Anderson, R., A. Boulanger, J. Longbottom, R. Oligney, Future Natural Gas Supplies and the Ultra Deepwater Gulf of Mexico, Energy Pulse, 2003, online at: http://www.energypulse.net/centers/article/article_display.cfm?a_id=232
Anderson R. A. Boulanger, Smart Grids and the American Way, Power & Energy, American Society of Mechanical Engineers, March, 2004.
Anderson, R., The Distributed Storage-Generation "Smart" Electric Grid of the Future, in The 10-50 Solution: Technologies and Policies for a Low-Carbon Future, PEW Center for Global Climate Change and the National Energy Policy Council, July, 2004.
Gross, P. R. Anderson, et.al., Predicting Electricity Distribution Feeder Failures using Machine Learning Susceptibility Analysis, International Association of Artificial Intelligence, July, 2007.

Machine Learning References:

Rouse, W.B. (1977). Human-computer interaction in multitask situations. IEEE Transactions Systems, Man, and Cybernetics, SMC-7, 293-300.
Rouse, W.B. (1988). Adaptive aiding for human/computer control. Human Factors, 30, 431-443.
The Social Life of Information, with J. S. Brown, & Paul Duguid Boston, MA: Harvard Business School Press, 2000
Enabling Knowledge Creation: New Tools for Unlocking the Mysteries of Tacit Understanding by Ikujiro Nonaka, Georg Von Krogh, and Kazuo Ichijo, Oxford University Press, 2000.
Bellman, R. E. (1957a). *Dynamic Programming*. Princeton University Press, Princeton, NJ.
Bellman, R. E. (1957b). A Markov decision process. *Journal of Mathematical Mech.*, 6:679--684.
Sutton, R.S., Barto, A.G. (1998), *Reinforcement Learning: An Introduction*. MIT Press.

Matrices References:

Akao, Y., ed. Quality Function Deployment: Integrating Customer Requirements into Product Design, Productivity Press, Cambridge MA, 1990.
Griffin; A., John R. Hauser ,The Voice of the Customer Marketing Science, Vol. 12, No. 1. pp. 1-27, Winter, 1993.
Hales, R., D. Lyman, and R. Norman, QFD and the Expanded House of Quality, Quality Digest, February, 1994.
Hauser, J. and Don Clausing, "The house of quality", Harvard Business Review, vol. 32, no. 5, pp. 63-73, 1988.
Karsak, E. E., Sozer, S., Alptekin, S. E., "Product planning in quality function deployment using a combined analytic network process and goal programming approach", Computers & Industrial Engineering, 44, 171-190, 2002.
King, B., Better Designs in Half the Time: Implementing Quality Function Deployment in America, GOAL/QPC, Methuen, MA, 1989.
Lowe, A., Ridgway, K. UK user's guide to quality function deployment Engineering Management Journal, Vol.10, 3, 147-155, Jun 2000.

Pugh, S., Total Design: Integrated Methods for Successful Product Engineering, Wokingham, England: Addison-Wesley, 1991.
Mazur, G. and Andrew Bolt, "Jurassic QFD" Transactions of the 11th Symposium on Quality Function Deployment, Novi, MI, ISBN 1-889477-11-7, June, 1999.
Pugh, S., Clausing, D. ed., Andrade, R. ed. Creating Innovative Products Using Total Design: The Living Legacy of Stuart Pugh. Reading, Mass.: Addison-Wesley, 1996.
Werbos, P.J., Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Harvard University, 1974.
Werbos, P.J., The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting, Wiley, 1994.
Werbos, P.J., Generalization of backpropagation with application to a recurrent gas market model, Neural Networks, Vol. 1, pp. 339-365, 1988.
Werbos, P.J., Maximizing Long-Term Gas Industry Profits in two Minutes using Neural Network Methods, IEEE trans. On Systems, Man, and Cybernetics, Vol. 19, No. 2, 315-333, 1989.

Recommended Resources

- FredHarriman.com Jidoka Page
- ISO 9000 - An Introduction and Description
- Lean manufacturing glossary Gemba Research
- Lean manufacturing Profit Technologies Glossary
- NWLEAN The Northwest Lean Manufacturing Network
- NIST's Engineering Statistics Handbook section 5.5.6. What are Taguchi designs?
- PBS I, Cringely . Archived Column on Taguchi applied to marketing.
- Robust Design - Taguchi Quality Methods