Metadigritter
- forskjellige RBC-sfelle (AR-bjelke)
- har van del rettsjiste av liver andrene
- typisk avsett.
Vedlikeholde prosedyre

Må beholde vedlikehør PA og LFH

La bekræft vedligehører GA og LFH
Svaktuben med GA

1

Riget plan lægte en del gomme.

Blande speer aktive gomme.

VARIGEN
7. Put the basket into the freezer.
RUBBS lemon and par of i special condiments.

8. Make the sunrise in Japan
Implementing continuous improvement using genetic algorithms

Petter Øgland, PhD student
petterog@ifi.uio.no
Department of Informatics, University of Oslo
P.O.Box 1080 Blindern, 0316 Oslo, Norway

ABSTRACT
Purpose – On the metaphoric level, much as been written about complex adaptive systems (CAS) for implementing total quality management (TQM) and organizational learning (OL) in turbulent or unpredictable environments. The aim of this paper is to add practical insights on how a specific CAS-technique called genetic algorithms (GA) can be used for designing quality management systems for keeping the organization in a constant state of continuous improvement in unpredictable environments.
Methodology/Approach – The paper describes design, implementation and evaluation of a GA method for a TQM program in the context of the climate department of a Scandinavian meteorological institute. The author was at the time working as a quality engineer responsible for designing the genetic algorithm as a change intervention for improving inflow, quality control and computer-generated statistical use of meteorological observations.
Findings – In the given organizational context, a genetic algorithm was easy to implement using elementary quality management tools such as statistical process control (SPC), Pareto Analysis and Business Model Assessment. Rather than going through conventional change management steps of unfreeze-change-freeze, the double loop structure of the chosen GA method made it possible to maintain “edge of chaos” in a stasis of continuous change. As expected from CAS theory, however, the algorithm caused stable improvement in unpredictable environment at the cost of producing redundancy, complexity and less than optimal efficiency.
Originality/Value of paper – Although the idea of applying GA to TQM is not new, this paper appears to be the first attempt to go beyond metaphorical ideas and computer simulations in order to actually define, implement and evaluate a GA method for TQM implementation.

Keywords Continuous improvement, organizational change, complex adaptive systems, genetic algorithms

Type of paper Research paper
Introduction

The way Shewhart (1933; 1938) defined statistical process control (SPC) as a tool for improvement research, and how Deming (1986) popularized and expanded the application areas for this method, there are similarities with the with unfreeze-change-freeze process from organizational development (Lewin, 1950). Before starting the improvement interventions, enough data need to be collected to make sure that current SPC statistics give a predictable portrait of the current process. The process can then be unfrozen and changed, and the new process is frozen as data is collected for defining the control parameters for the new SPC diagrams. Juran (1964) referred to his version of distinct steps of planning, control and improvement as “the quality trilogy”, while in current improvement programs such as Six Sigma (Pande, 1998), the DMAC method is a slight variation of the same idea.

However, not in all organizational contexts is it obvious how to freeze and unfreeze processes in order to manage controlled change. The software development industry is an example of an industry where ideas about management of quality and change have been influenced by the ideas above (Watts, 1989), but where either the nature of the work, the people who typically work as engineers and managers in such places, or a combination of both, often makes it difficult to achieve levels of equilibrium and stability of a similar kind that is achieved in the mass production context where the SPC ideas originally evolved.

The aim of this paper is to present and evaluate a framework for keeping an organization in continual change by locking it into a stasis of everlasting small improvements at the “edge of chaos” rather than going back and forth between stages of freezing and unfreezing. The framework consists of sequencing elementary statistical quality control methods such as SPC and Pareto analysis into the structure of what is called a Genetic Algorithm (Holland, 1998). The resulting improvement process is a algorithmic description of continuous improvement of the type known as “kaizen” (Imai, 1986).

The framework was developed and tested in the context of software development in the small organization of the climate department of a Scandinavian meteorological institute. In order to make the ideas presentable for a wider context, some of the specific issues related to this narrow industry have been downplayed in the presentation, but are commented upon towards the end of the paper, in the context of general claims and ideas for further research.

The paper is structured as follows. First there is a review of related research, providing a context for how the framework was constructed. Then the framework itself is presented, followed by a section explaining how it was implemented as a part of the general research design. This is followed by results and analysis. The final section provides discussion and conclusion.

Related work

Dooley, Johnson and Bush (1995) state there is a need for investigating genetic algorithms in order to develop models of parallel learning. The way the usual PDCA
learning cycle is designed and implemented in organizations usually assume an
environment that is sufficiently simple and predictable to avoid ending up causing
problems by improving the wrong processes.

There have been many applications of GA in operations research and manufacturing (e.g.
Suresh & Key, 1997; Chambers, 2000). There is an extensive literature on designing and
implementing genetic algorithms on the computer (e.g. ref ref ref).

Less seems to have been written on the application of the ideas from GA in the context of
organizational change. Axelrod & Cohen (2000) present a general framework for
discussing organizational change through concepts like variation, interaction and
selection, but although they state that the approach is based on genetic algorithms (ibid,
xvii, 10-11) no explicit algorithms are presented in the book.

Bruderer and Singh (1996) come closer to the aim of this paper as they actually define a
genetic algorithm for organizational evolution. However, rather than implementing the
algorithm as a method for doing organizational change, they use it for doing computer
simulations.

Stuart Kaufman might be a good place to start, or what about Kevin Kelly, or should I
jump directly to Kevin Dooley? Axelrod & Cohen is another good reference, even
though they do not talk specifically about TQM. I should also check possible reference
on GA and Pareto Analysis as I believe I found something related to this on Wikipedia.
Has there been written anything on TQM and CAS in The TQM Journal? That might be
a good starting point, as this is the main target for publication after the conference.

**Continuous improvement**

In their overview on the literature on continuous improvement, Bhuiyan and Baghel
(2005) the term “continuous improvement” is interpreted more like “ongoing
improvement” rather than continuous improvement in a mathematical sense, thus making
them include both incremental improvement and breakthrough improvement as a part of
the concept.

While this expansion of concepts may be useful in their research, …there may be reasons
for The way continuous improvement is used in this paper, however, is based on and the
aim is to make no distinction between continuous improvement and discontinuous
improvement

In order to Process improvement by use of Shewhart’s Deming’s Kaizen or “continuous
improvement” has been much discussed in quality management literature (e.g. Imai,
1989; …). While conventional methods of quality control in a mass-production setting
have emphasized a project-by-project approach in process improvement (Croby, 1979;
Deming, 1986; Juram, 1988), the Japanese way of implementing ideas from industrial
engineering (Ishikawa, 1985) appears to be more focus on action and less inclined to
make clear distinction between theory and practice.
Some researchers investigating the possibilities of applying ideas from CAS in TQM (e.g. Dooley, …) emphasize how a kaizen approach is different from the Lewinian unfreeze-change-freeze in terms rather being in a constant mode of change, sometimes referred to as “edge of reason”.

While there is much literature on how to use information technology for designing QMS in a conventional sense of control and breakthrough-improvement (ref xxx). There is less literature on CAS (xx). Stuart kauffman (1985), however, suggests CAS as an approach for TQM. On the wikipedia page there is a link between pareto analysis and genetic algorithms.

In order to clarify the nature of CAS/GA management, I should contrast the genetic algorithm approach with the “project algorithm” approach. A project is often defined as a way of organizing work that is constrained by time, cost and functionality. The project algorithm thus consists of breaking down work, estimating time and cost, monitor progress, evaluate and make sure that whatever was planned but turned out not to be feasible within the scope of the project is handed over to be taken care of by later projects or by systematic improvement during maintenance cycles.

**Framework**

There is a large body of literature on introduction to genetic algorithms (e.g. Michell, 1996; Goldberg, 1989; xxx). The framework to be presented, however, is only based on the simplest and most elementary ideas from GA theory. A typical pseudo-code for genetic algorithms provides a starting point (Wikipedia, 2009):

1. Choose initial population
2. Evaluate the fitness of each individual in the population
3. Repeat until termination: (time limit or sufficient fitness achieved)
   3.1. Select best-ranking individuals to reproduce
   3.2. Breed new generation through crossover and/or mutation (genetic operations) and give birth to offspring
   3.3. Evaluate the individual fitnesses of the offspring
   3.4. Replace worst ranked part of population with offspring

For implementing this algorithm as a PDCA-like change management process in an organizational context rather than as an algorithm to be run on a computer, key concepts need to be defined in organizational terms.

The starting point is the idea that a process can be broken down into tasks. The population chosen in the first step is a list of all relevant tasks, both tasks that are being carried out at the moment and tasks that seem relevant in the future. The fitness function is defined by (a) how tasks and functions fit with the business model, and (b) by internal errors detected in each task, either as they are explicitly observed or as a result of discovery through use of SPC diagrams.
In step three, the test criterion is defined by the organization having reached organizational excellence, which in practice means that the loop will never stop.

The four steps inside the loop starts with identifying tasks that are both fit with the business model and are not marked by internal problems or immediate needs for quality and efficiency improvement. The ranking is done through the use of Pareto analysis both on a day to day basis and on more longterm interval basis, for example month by month.

The step 3.2 consists of constructing a new set of tasks that consists of new designs plus changes in the descriptions of some of the old ones. In step 3.3 each individual in this offspring population is given a number 0 or 1 depending on whether they have been sufficiently tested to be put into production or not. Only individuals that have been sufficiently tested (fitness = 1) are used as input for step 3.4 where they together with the surviving members of the original population of tasks define the new population.

The main difference between the standard PDCA cycle and this interpretation of the genetic algorithm for the purpose of process improvement is that the PDCA cycle is usually used for improving individual processes or issue while the purpose of the genetic algorithm is to focus on the population as the unit of analysis.

However, the genetic algorithm does not conflict with the PDCA idea. The planning ("plan") is integrated in the way the offspring is defined, the experiment ("do") corresponds with how tasks are being monitored through direct observation and SPC, while control ("check") corresponds with how the fitness function works. The return to each cycle with the corresponding evaluation of the cycle criterion corresponds with the decision issue ("act") from PDCA theory.

Implementation

The story is based on my own experience as a research scientist at DNMI 1991-99. The success of the model influenced further development and implementation in a different context while working as a quality manager at NTAX 2000-05 and an action researcher at NTAX 2006-present.

The starting point for developing a new climate database KLIBAS was in 1989. It followed conventional waterfall lifecycle project management until 1995 when it was put into production. The team consisted of six research scientists and two software engineers.

The period 1996-99 was a period of maintenance, continuous improvement and further development through prototyping. A long series of scientific documents were written during the whole period, and there were several research papers presented at conferences. There was also written a large amount of technical documents and administrative documents.
In 2008 and 2009 I have returned to interview people who are now running the organization, trying to get an impression of the current state and different versions of how the development period was experienced.

**Results and analysis**

The first period 1991-95 was a period of conventional project management. Even though it was important for the project and the department to work scientifically, and explain to others that they worked scientifically, I would in retrospect argue that this period was non-rational in several respects.

None of the research scientists or software engineers had a formal degree in systems development. The research scientists had meteorological background, while my background was a M.S. degree in industrial mathematics (numerical analysis). The two software engineers had no formal education.

There were lots of problems; prestige, envy, jealousy, fights, battles etc.

The second period 96-99 was concerned with making a KLIBAS that would actually work after having spent many years on requirements, functional analysis, and design, producing low quality documents that were of limited use. The best way seemed to be to prototype the system from scratch.

The GA model was inspired by studying Kelly (1994), a book on artificial intelligence, a book on statistical quality control (1993) and “The Deming Method” (a book on quality management).

The model worked nicely for developing in a chaotic environment, but it also produced a system that was (probably) difficult for others to understand, despite a long trail of scientific publications on quality control algorithm and technical documents explaining the programs and how they interrelated.

On current visits to the organization I have been told that very little, or probably nothing at all of the CAS/GA system remains. Some of the predictions I made at an EU conference in 1999 have come true however (the total automation of the QC of climate data), and I see researchers referring to some of my scientific reports on documents on quality control.

**Discussion**

The case study showed how a quality management system was designed as a genetic algorithm for bringing the organization into a stasis of continuous improvement despite the organization being unpredictable and complex.

As the basic ideas of genetic algorithms are old and have been debated within the community of organizational research for the past 10-15 years, it was surprising that it was so difficult to find related research on how to use GA for designing continuous
improvement. Pondering on explanations for this curious finding, the following list comes to mind:

1. There has been done a lot of research on this, although not under the name of "genetic algorithms".
2. People are not willing to have their lives run by a computer algorithm. They want to make their own decisions.
3. Using GA generates complexity thus making it difficult to hand over the responsibility for the QMS from one person to another.
4. Quality management professionals with a statistical or industrial engineering background have adopted the beliefs of unfreeze-change-freeze that is the underlying assumption of quality management gurus like Deming and Juran, and have little knowledge of CAS theory.
5. Quality management professionals with a human relations background may find the metaphorical aspects of CAS and GA interesting, but are skeptical of the literal interpretation of such theory in terms of defining and implementing algorithms in a strict sense.
6. There is limited prestige in applying the GA approach. The way of the method is to implement TQM with minimal management commitment. There are few quick wins, and no particular focus on creating "breakthrough improvement". There is little in the method that would predict that people responsible for designing and running the system would feel valued or acknowledge for their work.

The first point is a difficult one. The way Imai (1986) describes kaizen as ongoing improvement by developing standards, improving standards, focusing on details, applying the "logic of the crowd" through the use of quality circles etc., looks very much like genetic algorithms put in action. In fact, much literature on Japanese quality management seems highly consistent with understanding organizations and society from a CAS perspective including the formulating of solutions (management systems) as GA (e.g. Shingo, Ohno, xxx). However, there is no mention of genetic algorithms in this literature, something that seems to be reflected in the fact that quality management methods that work well in Japan and other Asian countries do not export well into other cultures based on different values and beliefs (ref ref ref). Even though there may be literature explaining the GA approach through different terms and words, such literature is of limited value until the practices are described in a universal language, such as the mathematical language of GA, that will make it possible to analyze and understand this type of quality management in a precise and scientific manner.

Point two is interesting from the perspective that the case study has been done in a Scandinavian context where organizational structures are relatively flat, authorities are often questioned, rules are often created by not always followed etc. (Hofstede, ..), meaning that "management by algorithm" does not seem to fit particularly well with the cultural context of "Scandinavian pragmatism" (Brandt, 1973). Nevertheless, in this particular case it worked well. During the past decade there has been a world-wide growing interest in self-management methods that are described through flowcharts and algorithms (e.g. Allen, 2001; 2003). Some have tried to explain this phenomenon
through the theory that self-control leads to “flow” and happiness (Csikszentmihalyi, 2000). In my personal experience, I would say that getting rid of trivial administration by applying a simple system had a positive mental effect. On the other hand, in interviews I’ve done with quality practitioners, few or close to none of them seem to share the idea that the quality management consultant company or the quality department should apply the same sort of quality control and process improvement tools that want others to use (Øgland, 2007).

Point three corresponds with my observations in the third part of the case study. When returning to the organization where the GA-based QMS had been implemented, little or none of the system remained despite the fact that it had been working well, was extensively documented and there was still need maintenance and development of more software. While there may be many reasons for why the QMS had been redesigned in different ways, it did not come as a surprise to me that the GA approach was no longer in use. No matter how well the GA-based system had been documented and no matter how well it was working, the main mechanism in the GA approach is adaptation and adaptation builds complexity. In the process of one person taking over the responsibility for a system developed by another person, there is always the question of whether one should invest time in figuring out what the other person had done and continue along that path or rather redesign and follow their own experiences and beliefs in how things should be done.

Concerning point four, while there may be researchers that have studied the implications of CAS theory in the context of TQM (e.g. Dooley, 1995; 2001; Wheeler, xxx), I do not believe these insights have yet been fully adopted by the quality management community in a way that leads to practical implications like designing TQM by GA. As can be observed from the online discussion forum on the American Quality Society, even the moderator of the discussion thread on complex adaptive systems seems to have limited knowledge of CAS beyond the metaphorical level (ASQ, 2009).

Point five fits with my personal experience of social isolation and a certain level of alienation as a consequence of formalizing and computerizing management issues, thus reducing the need for meetings and discussions as the machine was more or less fully capable or doing the administrative work of planning, monitoring and evaluating work. The fact that concepts like CAS and GA had little to do with meteorological or climate issues did not help much either. On the other hand, although technical skill was necessary for implementing the system, the idea was to eliminate the “waste” of trivial administrative work in order to be able to focus on creative work. Although creative work in this particular case was to a large degree focused on software development, there was also a monthly discussion with the technical department for discussing automatic weather stations. If properly understood and simply implemented, there is no reason why the GA approach should not be used to TQM experts with a HR focus.

The final point about power, pay, prestige and social issues seems relevant on a first approach. As none of my co-workers understood anything about CAS and GA, and had little interest in such issues, my peer network consisted of a handful of TQM researchers
in other Scandinavian geophysical institutes. Implementing TQM as a long-term, noiseless and invisible process did not correlate well with aspirations of power, prestige, pay or social acknowledgement within the organization, but, on the other hand, I believe this has more to do with the nature of working in the field of quality management and organizational development than anything particular to do with the GA approach. What makes the GA approach different from other ways of cultivating TQM is that it is designed to handle an unpredictable organizational environment through applying a robust approach by trading off on efficiency and visibility, at least this was the way the approach was implemented in this case study.

**Conclusion**

Many TQM and OD projects fail due to lack of management commitment, unpredictable environments, lack of motivation, technical incompetence or due to the fact that a rational analysis of a complex organization may be faulty and cause the improvement projects to follow wrong trajectories. Literature suggests that complex adaptive systems (CAS) might be a good way for describing such organizations and use CAS theory for developing TQM and OD strategies. In natural science, these insights have led to practical solutions such as genetic algorithms (GA), but in organizational theory little seem to have been achieved beyond using GA for simulating organizational development on computers or using metaphors for stimulating change.

In this paper, however, a framework is presented, showing a simple way of designing a continuous improvement using GA. The algorithm consists of a few simple steps, it only uses the most basic quality management tools, and a case study describes how it was designed, implemented and evaluated in a real-world setting.

The fact that the approach is claimed to be both simple to design and adequate for the given context makes one wonder why it is so difficult to find similar research results. Six explanations are suggested, but when investigated, none of them are capable of answering the question in a satisfactory way.

As the problem of managing quality in unpredictable environments is becoming more and more important, and as the GA is a highly successful solution for non-social problems of similar kind, more theoretical insights on the issue of using GA for designing continuous improvement in unpredictable environments should be of significant importance for the TQM and OD community.

**References**