DSM – WHERE IT'S BEEN – WHERE IT NEEDS TO GO

Donald V. Steward
Steward@Problematics.com

DSM has come a long ways since its origin in the early 1960s. And many of you have been very involved in its growth. But it still has a long way to go.

DSM had a difficult birth. Initially three journals turned it down and over a decade went by before it was finally published for the first time in 1981. Essentially that same paper was published by IEEE Transactions on Engineering Management and a book was published based on my Ph.D. thesis from the University of Wisconsin. Then Steve Eppinger, Dan Whitney and others at MIT picked it up. This resulted in many DSM theses, my being asked to be a visiting scholar at MIT in 1992, and eventually led to this series of International Workshops. Earlier workshops were hosted at MIT, Cambridge University and the Boeing Company. We are now here at the 9th workshop at the University of Munich.

Why did it have such a difficult birth? For thousands of years our principal method for solving problems was based on breaking them down in hierarchies so that we could solve the smaller problems, and then build their solutions into the solution of the original problem. This hierarchical approach dominated the structures of all of our organizations, which tend to reflect how we solved problems.

But over the years, the problems became more complex. Symptoms started to arise indicating that the hierarchical approach may no longer be adequate. People started to escape the hierarchy in order to get problems solved, not through the hierarchy, but despite it. Initially some were even punished for doing so. But eventually it became a way of life when it was recognized that escaping the hierarchy was necessary. The role of circuits and the need for assumptions had not yet been fully recognized. This 'outside the hierarchy' activity tended to be undisciplined; resulting in much wasted effort and overlooked assumptions that caused projects to end in disappointment or even catastrophe.

The DSM concept was an acknowledgement that complex problems did not have nice hierarchical structures. Each problem had its own structure that typically contained circuits and couldn't be solved without using assumptions. But this required that people had to change the way they thought and how they organized their work. This was a paradigm shift. And paradigm shifts don't fare well in their early years.

DSM came out of an environment where many people were working with matrices to show how things were connected. DSM was different in its interpretation of how this structure could be used to solve problems.

It took me many years to gain an understanding of DSM's implications, and I am still working on that. It's more that just an engineering method. It's a problem solving method. Although its application began in engineering, it needs to be extended to business as a whole. After all, the business of business is solving problems; solving its own problems by solving its customer's problems. The business world is a problem solving network. I'll solve your problem if you solve mine.

To my knowledge DSM has been used primarily for planning projects. But I believe that the real payoff won't be fully realized until we also use it to manage the conduct of that plan. As the project proceeds, some assumptions get resolved, doing away with many of their risks. This allows us during

the project to focus on what assumptions and their risks remain. This should be used as an important measure of progress.

And now we have the Explainer. Given knowledge about something in the form of information and its associated relations, the Explainer is a computer program that can find the causes of that thing's behaviors. It can be used to diagnose why things don't behave as they should, or show how they can be made to behave in some desired new way.

Again, it's a new way of thinking and it's having its own birth pangs. Echoes of DSM. Apparently this capability had not been developed earlier because people had the mistaken belief that any such method for doing this would have to be NP-complete, meaning that the only way of finding all explanations would be to blindly generate all combinations of elementary causes and assumptions and test each one by deduction to see if it provided the needed explanation. But the computing times for this approach would increase exponentially with the size of the problem. This was perhaps good enough for explaining the behaviors of very small systems, but not good enough for the size of systems we would like to deal with.

The answer to how to solve these problems in polynomial computing times to make it practical became obvious, but only after years of thinking about it. It is not necessary to test all combinations generated blindly, but to find only those very few combinations for which there are paths from the behaviors to be explained to the hypotheses that could explain them. If the relations are transitive, then the appearance of intermediate events can be eliminated by substitution just as we would solve a system of simultaneous algebraic equations. Direction comes from solving for the behavior in terms of the hypotheses rather than vice versa. The only remaining problem is that substituting Boolean expressions can generate redundancies that balloon out of control if there are not eliminated quickly. The solution is fast, simple, and now perfectly obvious.

Why do I bring up the Explainer at a workshop for DSM? Because I believe that Explainer, like DSM, can be a design tool. We need to pursue this possibility further. I have shown how an outline of a design can be produced by asking for an explanation of how the requirements can be met given knowledge about the components that it would use. Further, I believe that an audience that has shown their support for the DSM concept could well be the first to understand the potential of the Explainer.

The method is extremely fundamental and thus applicable to many different types of problems. But as long as most people believe it couldn't be solved and are stunned that anyone would be so foolish as to claim he had done it, and as long as reviewers can't find references to hang their courage on, it probably won't be published. So the word will have to spread by word of mouth, or by workshops like this until people can publish not the just the concept, but successful applications that people can believe in.

The process generally occurs in this order: 1. Concept, 2. Awareness, 3. Resources, 4. Application, and 5. Publication. The Catch-22 is developing awareness before it's published.

The lesson is: When you think out of the cage, you still have to deal with the people inside the cage.

Incidentally, Harvard Business Review wouldn't publish the paper I submitted just recently on applying DSM to business. They say they like the concept, but won't publish until I have a demonstration that it has actually been done. If you can demonstrate that it has been done, I would like you to coauthor a paper with me that I think we can get published by Harvard Business Review.

Thank you for your continued excitement about and contributions to using DSM.

Now your work is cut out for you: extending DSM to business environments and to tracking a project as well as planning it, and demonstrating the application of the Explainer to practically everything. And oh yes. Continue thinking outside the cage. I'll be there outside with you.

9TH INTERNATIONAL DSM CONFERENCE

DSM & Explainer Overview

Where we've been. Where we're going.

Donald V Steward

Problematics LLC
Steward@Problematics.Com
http://www.problematics.com

IN COOPERATION WITH BMW GROUP
CAPITALIZE ON COMPLEXITY

Both DSM & Explainer began as audacious ideas

- Both were initially declared NP-complete.
- DSM is no longer audacious.
  - But its applications need to be extended.
    - Active Risk Management to track progress
    - Application to general businesses
- The Explainer is still audacious.
  - It still needs to be understood.
  - It's too simple, too widely applicable, after much work too obvious, and too impossible to be believed.

The DSM idea began in the 1960s

- I had found a way to flatten the power shape in a nuclear power plant and thus raise its power by starting with an optimal shape and computing the infeasible control distribution to produce it.
- Myron Tribus had a way of diagramming equations.
- DSM was an attempt to generalize these concepts.

Early events

- 1968: General Electric internal report *The Design Structure System*.
- 1973: Ph.D. Thesis: University of Wisconsin
- 1981: Book: *Systems Analysis and Management: Structure, Strategy and Design*
- 1981: “The Design Structure System: A Method for Managing the Design of Complex Systems” in *IEEE Transactions on Engineering Management*.
- 1983: Eppinger’s invitation brought me to MIT as a visiting scholar
- 1999: Beginning of DSM workshops

DSM Principles

- Every problem has a structure.
- The problem solving process must ultimately follow that structure.
  - We can thrash around until we stumble onto the structure.
  - Or we can first find the structure and manage the process by using that structure.
- This structure is usually not a hierarchy.

IN COOPERATION WITH BMW GROUP
CAPITALIZE ON COMPLEXITY

DSM Principles

- The structure of the problem usually involves circuits.
- Circuits are contained in blocks on diagonal.
- Circuits are broken by using assumptions.
- Assumptions are risks.
- Planning the approach shows what are the assumptions and where they:
  - Will be used.
  - Can be verified.
- Then track remaining risk by what assumptions still remain unresolved.

DSM Questions

- Tearing reorganizes the items within a block to develop an approach that defines what is assumed, where it's used, and where it can be confirmed.
- Who is using tearing?
  - If not, why not?
- Who is reducing their risks by tracking their risk progress?
  - If not, why not?

IN COOPERATION WITH BMW GROUP
CAPITALIZE ON COMPLEXITY

Now the Explainer Problem

- Given a system or situation described by a set of cause-and-effect statements, provide hypotheses to explain any observed or desired behavior of that system.
- Then use deduction on each hypothesis to describe how it was arrived at and generate side effects beyond what was to be explained.

Once this problem is solved, you should be able to:

- Diagnose medical symptoms or find the causes of difficulties with various machines and processes.
- Design by asking requirements specifications to be explained by cause-and-effect descriptions of likely components.
- Solve crimes.
- Explain and change situations.

But ‘everyone knows’ that there is no practical solution for this problem. Thus no one was foolish enough to try to solve it.

Then along comes a fool who finds they were looking at the wrong problem.

So now how do you override their prejudice to convince people that it can be solved?

By developing these applications!!