White Paper

SSM Stage 1 – Accepting the Challenge

WP0173 | January 2015

Ceri Williams

Ceri has thirty years in the IT industry, originally delivering complex control systems and subsequently broadening focus to Enterprise Architecture, Governance and transformation of the IT function. Working as a chief architect, consultant and coach, he enables FTSE 250 organizations to make medium and long term decisions on the shape of the Enterprise Architecture and positioning of the IT function.

He advocates putting people at the heart of technology and business change with focus on the human enablers and constraints. His work deals with the way in which rigorous engineering and architecture disciplines are integrated with the cognitive and behavioral capabilities of the people who practice them.

The previous white paper in this series [Ref 1] introduced the processes, steps and stages that the Soft Systems Methodology (SSM) provides to guide practitioners, and the way in which they afford well-formed integration points for blending with engineering disciplines such as INCOSE and TOGAF. It sets the scene for the more detailed exploration of what SSM can add to commonly encountered Architecture Methods to enrich them and make them more effective.

This white paper deep-dives into the first process step – Entering the problematic situation – and illustrates it through scenarios regularly encountered by the Enterprise Architect. It continues to explore the structured approach that the Soft Systems Methodology provides to guide practitioners, and the way in which this affords integration points for blending with engineering disciplines and frameworks.

There's no substitute for reading the papers themselves, but for readers short of time, the next section is an extract taken from Papers 1 and 5. It provides a very short outline of the Soft Systems Method - what it is, where it came from, and why it is significant. Readers wishing to deepen their background in the topic before embarking on this Paper can read the previous papers [Ref 1]. Readers already familiar with these papers can skip the next section.

A (very) Short History of Soft Systems

In a nutshell - the Soft Systems Methodology (SSM) is a systemic approach for tackling real-world problematic situations. Soft Systems provide a framework for users to deal with the kind of messy problem situations that lack a formal problem definition. Enterprise Architecture deals with “real-world problematic situations” and routinely encounters “messy problem situations that lack a formal problem definition” – this is why a re-imagining of Enterprise Architecture as a blend of Soft Systems and Systems Engineering disciplines is now needed, and provides us with a complete set of concepts and tools with which to operate in a complex, people-centric environment.

The SSM originally emerged in the 1960s in response to problems encountered in tackling management & organizational problems using a systems engineering approach. From Ref [3]: “…the pattern of activity found in Systems Engineering – namely, precisely define a need and then engineer a system to meet that need using various techniques – was simply not rich enough to deal with the buzzing complexity and confusion of management situations”. I would add that the Systems Engineering approach also makes a number of (usually unstated) assumptions. Specifically that:

1. The problem and solution space can be modeled as a single definitive version of ‘the truth’ that is common to all stakeholders.
2. A stable snapshot of the environment (people, process, material) can be base-lined and persists largely unchanged during engineering analysis and solution delivery.
3. The time taken to assemble the baseline and develop a solution is short enough that the solution is relevant and valuable at the time it is implemented.

Every movement has its gurus, and Soft Systems is no exception. The first mainstream work to encode and specialize the knowledge around Soft Systems centered around Lancaster University, UK in the mid-1960s pioneered by Professor Gwilym Jenkins and subsequently by Dr. Brian Wilson, before reaching the mass market through the work of Professor Peter Checkland. A number of useful references are included at the end of this White Paper.

Despite the name, the Soft Systems Method does not differentiate between ‘Soft’ and ‘Hard’ systems. It does not even treat ‘Hard’ and ‘Soft’ as features of the problem under consideration – they are features of the relationship between the problem and the person interested in it. They relate to the way in which the problem analyst perceives and interacts with the situation. For this reason it provides the best reference point for Enterprise Architecture and an inclusive, systematic framework for integrating Engineering and Soft Systems approaches. For the sake

of clarity in this series of papers, provided we accept that we construct our viewpoint to represent a ‘system’ and that ‘Hard’ and ‘Soft’ are not intrinsic to the system, we shall refer to ‘Hard’ and ‘Soft’ Systems.

For further reading and a very concise and complete account, see [Ref 2].

For the purpose of this series of White Papers and in line with the general consensus in the field, Soft Systems and Hard Systems are treated as views of a system, rather than features of the system itself. Hard Systems are generally well suited to treatment with a Systems Engineering approach, soft systems with Soft Systems Methods. These viewpoints can be differentiated as described in Figure 1. The following Table 1 considers the main distinctions between Hard and Soft systems.

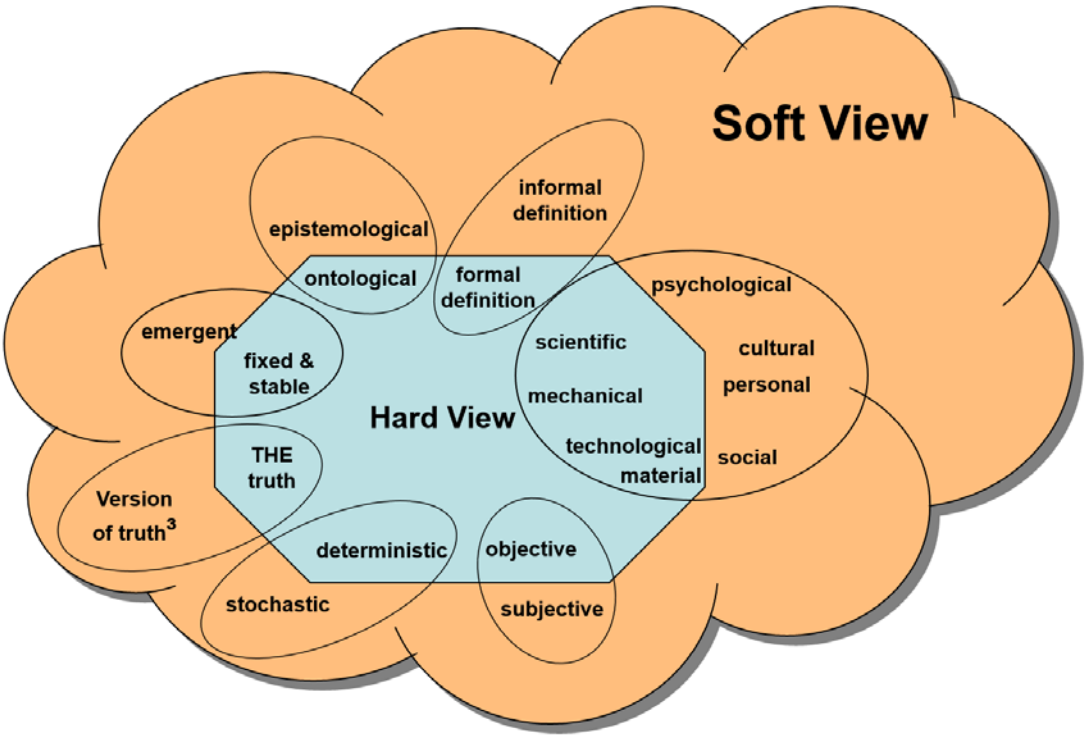


Figure 1 – The Relationship between Soft and Hard System viewpoints

Figure 2 provides an outline of the principal SSM 'Stages' that help the practitioner organize the work involved in following an SSM approach.

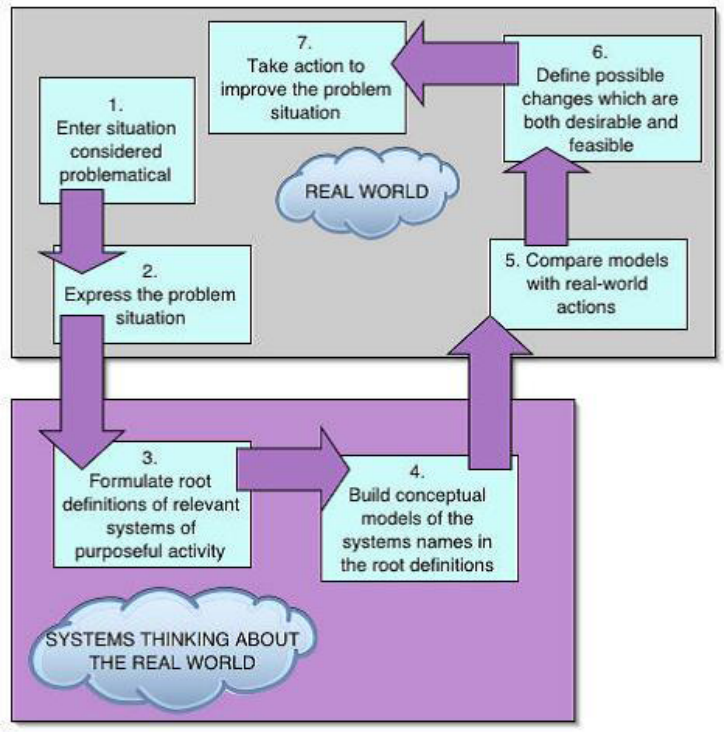


Figure 2 – SSM Process Stages

Configuring the Stages

This white paper focuses on the early Stages and places them in the context of the Enterprise Architect. First though, it is worth taking a quick look at the observations made by Don Koberg and Jim Bagnall in the All New Universal Traveller (Ref [3]). They adapted the standard SSM Stages and made them more suitable for design challenges, re-casting them as:

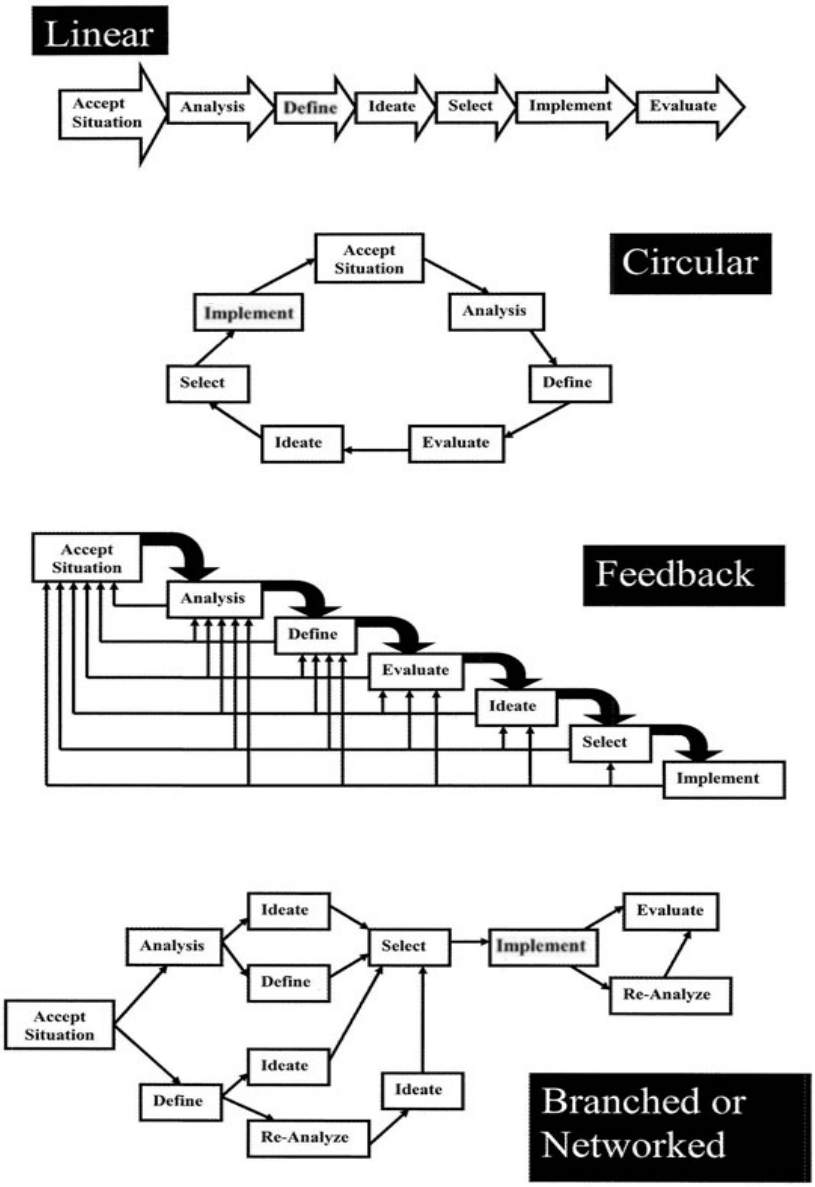
- **Accept situation:** pretty much a match for Stage 1: enter the problematic situation but with additional emphasis on the acceptance and commitment of the participant to addressing the challenge.
- **Analyze:** this is mainly about understanding the "world of the problem" and discovery of the relevant features of the situation – so not just Stage 2: express the problem situation, but doing so iteratively and intelligently, adjusting the targets for discovery depending on the feedback from initial analysis.
- **Define:** brings together two SSM Stages – Stage 3: Formulate root definitions of relevant systems of purposeful activity and Stage 4: Build conceptual models. It uses the information gathered during Analyze to structure the definition of the challenge in a way that facilitates problem solving. It provides the bridge between analysis and synthesis.
- **Ideate:** focuses on options generation – the 'art of the possible',

an adaptation of Stage 6: Define possible changes. This should be a very familiar stage to Systems Engineers, for whom 'Optioneering' is a critical process during which trade-offs are analysed and solutions optimized.

- **Select:** also focuses on Stage 6 as it involves choosing from the options, partly based on feasibility and desirability.
- **Implement:** all about taking the action needed to improve the situation – Stage 7.
- **Evaluate:** not present in SSM, this stage is focused on becoming aware of the results of the actions and closing the loop on cause and effect.

Koberg and Bagnall recognized that there were a number of ways in which these processes could be positioned in relation to each-other. A couple of these will look very familiar to the EA versed in the competing approaches of 'waterfall' (aka. 'Linear') vs. 'agile' (aka. 'Circular'), but the others also provide some useful ways of arranging the processes.

The way in which the processes are configured is as much of a decision for the SSM practitioner as the content they engage with through them – these arrangements represent significant choices for the Enterprise Architect and the participants in the process. As an example, the 'Branched or Networked' approach works well with a traditional engineering 'System of Systems (SoSA)' approach as it allows for a recursive approach to each Stage (e.g. reaching the 'Define' Stage for a System and then proceeding from the 'Analysis' Stage for each of the Subsystems in parallel.)



Stage 1: Enter the problematic situation [Accept Situation]: this involves acceptance by the participants; that they are prepared to assume responsibility for understanding a situation and working through improvements to it. It is important at this stage for participants to consider carefully what they can feasibly accept or must practically reject.

Consider a typical situation for an Enterprise Architect in a consulting role. He is commissioned by a visionary and enthusiastic CEO to engage her first line reports in an Enterprise Architecture work stream. She has read about the use of EA to define and implement business as well as technological strategy and is keen to try it out. However, for the Executive team, the EA initiative presents as just one more thing on the to-do list for which they don't have time. It is something that only pays off their time investment far in the future (and they may not be in post at that time) and is the latest of a series of pet projects of the enthusiastic CEO, few of which run to completion.

Without clear and visible acceptance on the part of the team, the EA will continually struggle to engage the key participants in the process as they drift away onto BAU and development of more pressing local strategies. It is important under these conditions for the EA to reflect back to the Exec team what he notices about their acceptance, focused on the congruence or dissonance between what is said (e.g. in whole team workshops) and what is done (e.g. between workshops). The EA has to recognize the common signs of acceptance and non-acceptance such as: turning up late (or not at all), lack of participation in group sessions, domination of sessions, lack of progress on actions, disruptive revisiting of decisions already made, 'learned helplessness' and passive-aggressive behavior.

Participants (and the EA) will need to make their time, skills, energy and commitment all available if the Enterprise approach is to be sustained and of lasting value. Koberg and Bagnall have some interesting suggestions for techniques to encourage Acceptance and encourage disclosure by participants of their real degree of Acceptance:

- **Ad valorem:** finding ways to increase interest by increasing personal importance – the EA can encourage the participants to explore how they can make the EA work stream more important for themselves.
- **Personal Priorities Matrix:** a graphical (and highly visible) way of showing individuals and their team-mates how the EA objectives stack up alongside others that are competing for your time and energy
- **'What's in it for me?'** – this is the 'benefit' side of the cost/benefit analysis, focused on the individual benefits. Collective benefits may also be useful provided the team has a history of motivation through collective achievements. One area to explore here is the synergy between the EA work stream and the high priority work that individuals are already tasked with – such as producing the budget forecast, business growth plan, product development plan.
- **Conformity:** this is the process of trying the EA approach out as an experiment, for a limited time, but for real. Sometimes it is better to just get the habit by doing and then question the value, rather than the other way round. This is often a good strategy for introducing EA as an approach in an environment where it is an alien concept, rather than trying to understand what it's all about in theory first. Some things are only really understood and taught by doing, rather than by telling or theorizing. This is the difference between something that is exoteric and something that is esoteric.
- **Give it Up:** leaning on Zen philosophy and Buddhism, this technique encourages something that is anathema to the CEO – that the participants consciously decide to give up focus

on the goal and immerse themselves in the process. This can be particularly helpful if the whole 'EA thing' seems rather overwhelming – encouraging them to just take the first step and only then the next.

- **Self-Hypnotism:** while on the surface, this sounds like a rather flippant technique, it boils down to visualizing the 'problem' being solved and how that feels, in order to create energy and commitment. The job of the EA here is to help the participants associate the effect of that feeling with Enterprise Architecture as the cause.
- **Who's in Charge:** not quite as suggested by the title, this technique focuses on reminding participants that there are only two choices – accept things as they are or take responsibility for changing them. This is useful if the participants do not feel empowered to follow through the commitments implied by the EA approach.
- **What's holding you back?:** this technique involves the participants explicitly disclosing and working through the main causes of non-acceptance. These are not unique to Enterprise Architecture – they are an integral part of everyday life:
    - It feels like punishment to do this
    - It is more beneficial to do something else
    - The relevance of doing this is not clear (or believed!)
    - There are other problems & obstacles in the way

The main aim of this approach is to turn participants from feeling like victims into individuals with a sense of agency.

- **I am responsible:** this rather mischievous tactic encourages the participants to consider some overwhelmingly difficult challenges, such as elimination of poverty or global warming. The idea is to put the EA challenge in perspective and illustrate that, by comparison, it is far from overwhelming. This realization can then put the team into a more resourceful state.
- **Tragic Scenario:** considers all the worst things that could happen if the team doesn't successfully embed EA as a routine working practice. A good starting point for this is to examine todays problems that can be traced back to the absence of coherent long range planning a couple of years in the past.
- **Contingency Management/ Habit Making:** these techniques are all about addressing the excuses for inertia as general excuses rather than reasons. It involves 'training' in new habits as a pre-requisite for deciding whether EA is worth committing to. This

ensures that if it is not followed up, it is for real material reasons rather than just entrenched habits.

Of course, it may well be that having gone through this process of Acceptance, it becomes visibly apparent that the team is not yet collectively ready, willing or able to accept the EA challenge. This has to be seen as a legitimate outcome of the process which the EA feeds back to the CEO, and one that has emerged through systematically working on Acceptance, rather than either just assuming it or ignoring it. Hopefully though, by working through some of the techniques described, at least some of the team feels ready, willing and able to move forward with the project and enthusiastically demand that they proceed to the next steps without delay.

White Paper #7:

White Paper #7 deep-dives into the next Steps:

- **Stage 2:** Express the problem situation (Analysis)
- **Stage 3:** Formulate root definitions of relevant systems of purposeful activity (Definition)
- **Stage 4:** Build conceptual models (also Definition)

I hope you have enjoyed this White Paper. Please get in touch if you have views to offer on the topic and feedback on the series, either direct to Orbus or via my eMail at: ceri.williams@theintegrationpractice.co.uk.

References:

[1] Enterprise Architecture meets Soft Systems Series, Papers 1-5. Orbus: www.orbussoftware.com/resources/authors/ceri-williams/

[2] Checkland, P & Poulter, J: learning for Action – A Short Definitive Account of Soft Systems Methodology and its use for Practitioners, Teachers and Students. ISBN: 9780470025543

[3] Koberg, D & Bagnall, J: the All New Universal Traveller – a Soft Systems guide to creativity, problem-solving and the process of reaching goals

Orbus Software
3rd Floor
111 Buckingham Palace Road
London
SW1W 0SR
United Kingdom

+44 (0) 870 991 1851
enquiries@orbussoftware.com
www.orbussoftware.com