White Paper

How to Systematically Incorporate Social and Cultural Factors into Enterprise Architecture Practice

WP0152 | June 2014

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 behavioural capabilities of the people who practice them.

The previous white paper in this series [Ref 1] considered the similarities and differences as a means of understanding where an Enterprise Architect can be more effective by switching between an engineering approach and the Soft Systems Methodology (SSM). It proposed that while the Enterprise is always complex, it is a long way from being an engineering object. This means that an exclusive focus on a 'hard systems' approach actually prevents the Enterprise Architect and associated stakeholders from understanding the Enterprise in a way that leads to effective and efficient improvement.

The original outline of this third paper in the series focused on how the Enterprise Architect can recognize the situations when Soft Systems methods are likely to be helpful and when an Engineering approach may be more appropriate. Based on feedback from the first two papers, I'm going to deviate slightly from this focus to continue with examination in more detail of a key area where the Soft Systems Methodology (SSM) complements the traditional engineering approach to Enterprise Architecture.

This paper drills deep in to a key area of difference and explores what it means in practice for the Enterprise Architect. It considers how SSM is inclusive of all areas of the situation/action space (i.e. scientific, technological, mechanical, material, psychological, social and cultural), while an engineering approach excludes psychological, social and cultural influences. This paper describes how an Enterprise Architect can appropriate elements of SSM and related social and cultural disciplines

and blend them in as a defined part of a holistic approach to Enterprise Architecture.

There's no substitute for reading the papers themselves, but for readers short of time, the next section is an extract taken from Paper 1. 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 two papers [Ref 1]

A (very) Short History of Soft Systems

In a nutshell - the Soft systems methodology (SSM) is a systemic approach for tackling real-world problematical 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 Soft Systems Methodology 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 baselined 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 & 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 3].

Key Concepts

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 and highlights those considered in the remainder of this paper in Bold.

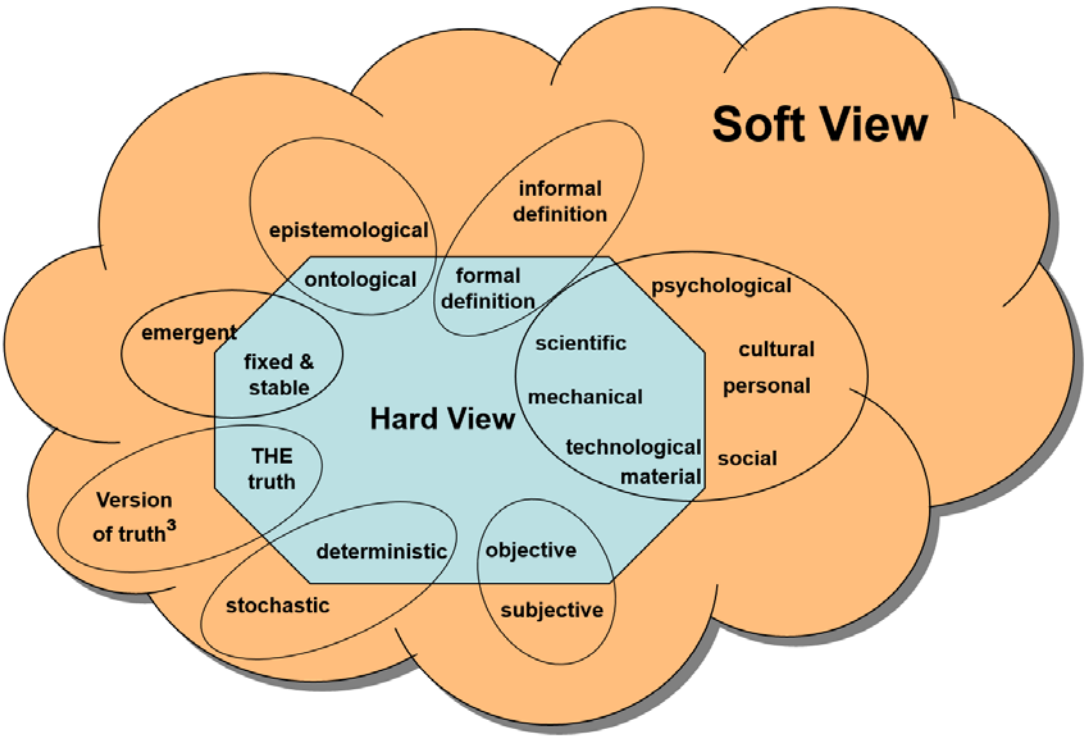

Figure 1 – The Relationship between Soft and Hard System viewpoints

#	Soft System View	Hard System View
1	Inclusive of scientific, technological, mechanical, material, psychological, social and cultural domains.	Inclusive of scientific, technological, mechanical, material domains. Exclusive of psychological, social and cultural domains.
2	Accepts that Systems develop emergent properties that cannot be foreseen at the outset. Provides concepts and tools to cater for this.	Assumes fixed and defined System and environment in which it operates. Unanticipated changes to either require re-entry into the Systems Engineering process at some point.
3	Provides the ability to integrate Systems that exhibit features and behavior that may be random, stochastic (i.e. statistical) and deterministic (i.e. individual cases predictable by analysis).	Deals effectively with deterministic systems and environments in which they exist. Has limited ability to deal with stochastic systems.
4	Tolerant and accepting of subjectivity and multiple 'versions of the truth'. Treats all models as viewpoints that express how stakeholders perceive the system. Accepting of dissonant and inconsistent viewpoints.	Considers multiple viewpoints as filtered views of a single, objective, canonical definition of a system or problem. Assumes and requires common agreement across all stakeholders, convergence and consistency of viewpoints.
5	Conceives of 'System' as an epistemological entity – i.e. as made up of conceptual and mental schemas & models that determine the perception of what the system is. Considers the perceptual schemas are an integral part of the 'system'.	Conceives of 'System' as made up of ontological entities – i.e. representation of, or actual entities physically existing or proposed to exist in the real world. The 'system' is independent of the way in which it is described.
6	Integrates Systems and problems that can and cannot be represented by formal definitions. Formal definition may not be possible either because of the nature of the System or because there is no suitable formal language with which to describe it.	Requires that problems and Systems can be represented by formal definitions (i.e. having conventionally recognized form, structure or set of rules). Assumes that they are structured, well-formed and logical.
7	Recognizes the significance of stakeholder values and world views (Weltanschauung) and their impact on the scope and shape of the System.	Recognizes stakeholder values and world views only to the extent that they filter the information that represents the system and separates stakeholder concerns.
8	Seeks problem and 'solution' definitions, actions and commitment to change that stakeholders can live with, rather than that they all agree on. SSM calls this 'Accommodation' between differing views.	Seeks consensus across stakeholders and requires that they believe the same 'truth'. Treats alternative views as incorrect and in need of change.
9	Inclusive of change to structures, processes and attitudes as a means of delivering improvement to a situation.	Inclusive of structures and processes, does not cater for attitudes.

Table 1 – Differences between Soft and Hard Systems Viewpoints

Items 4 and 6 were considered in some detail in Paper 2 in this series [Ref 1]. This Paper explores item #1.

Social & Cultural Features

Paper 1 [Ref 1] indicated that an engineering approach considers a System to be made up from scientific, technological, mechanical and material components. SSM adds psychological, social and cultural components to this mix. Recognition of 'World View' as a component of the system represents a major step toward inclusion of social and cultural features as an integral part of the system, rather than external to it.

If an engineering approach considers social and cultural features at all, it is typically very late in the change process and typically in reaction to the lack of adoption of a new piece of technology or a new process. Often accompanied by mystification and indignation around why the stakeholders can't 'just follow the defined process' or can't 'just live with the off-the-shelf data model'. In engineering, the challenges of adoption and adjustment of the way people operate are at best addressed following system design & build – during transition planning. At worst, they are not considered at all.

A soft systems approach would recognize that the seeds of adoption are sown much earlier in the process, as adoption of new ways of doing things is enabled and constrained primarily by social and cultural features. There is little point in designing a theoretically efficient and effective new process if the culture of an organization means that it is resistant to following any process. In this case, the best first intervention would be to experiment with the stakeholders to develop the capability to 'work to process'. Where this requires a shift from diverse individualistic behaviour to convergent collective behaviour, a cultural intervention is the only effective course of action. An engineering approach does not possess the concepts, language or structures to design such an intervention – it does not even possess the sensory apparatus or motivation to recognize such challenges.

SSM possesses the necessary concepts, language and structures to both recognize such challenges and design social and cultural interventions to deal with them. More than that, as it is inclusive (i.e. a superset) of the engineering approach, it facilitates their integration with more structured formal engineering activities. For example, while an engineering-centric approach would typically hand over (explicitly or by omission) responsibility for cultural and social change to the 'Management of Change (MoC)' team once the system is built, the SSM approach would not just integrate MoC with earlier (e.g. scoping, requirements) phases of an improvement initiative, it would lead with it. An engineering-based initiative may then be deemed an appropriate intervention, alongside others and be fully integrated into the overall plan.

For illustration let's use the fictitious organization Complex Systems Inc (CSI) that I introduced in Paper 2. CSI is a FTSE 250 mobile telecommunications provider delivering technologically-intensive products and services to the consumer and business markets. The development and delivery of these products and services is predominantly performed by a Product Development function. A separate Operational Support Systems (OSS) function provides the intelligent communications network that enables the products and services.

The Enterprise Architect with responsibility for OSS needs her Architects and delivery teams to better integrate with the Product Development team, specifically to deal with ongoing problems caused to the mobile bearer network by bandwidth-hungry products that are getting more and more sensitive to Differentiated Qualities of Service. The OSS people believe they already know the answer. Coming from an engineering based, rigorous and methodical culture, they believe that all that's needed is a gated governance process for 'accreditation' of new products and to not let anything through that might cause problems. However, this has already been recently tried, and the Product people just ignored it, bypassing many of the checkpoints, subverting the intent.

Leaning on SSM, the OSS EA decides to bring cultural and social factors and worldviews of the Product Development and OSS functions in as explicit parts of 'the system', and commissions the Business Support Systems (BSS) EA to facilitate. Not being an integral part of these two 'systems' means that the BSS EA is in a perfect position to become what the anthropologists call a 'participant-observer' (Jerry Weinberg gives a digestible account of this in [Ref 2]). The BSS EA starts the process by focusing on language and categories, risk appetite and the ability to work systematically to a defined process.

Adopting the role of 'participant-observer' means that the EA can facilitate a process by which Products and OSS stakeholders begin to recognize similarities and differences in their use of language and the way in which this exposes significant cultural differences. 'Ready for Release' meant very different things in each camp, as does the 'delivery process'. In Products, the delivery process is primarily a creative one, in OSS it is a rigorous engineering production endeavour. Relatively speaking, 'Low Risk' in Products meant 'High Risk' in OSS (and 'High Risk' in Products = 'Out of the Question' in OSS!). Facilitation by the BSS EA helps each camp recognize their 'ethnocentrism' – i.e. their tendency to identify the 'foreign' language of other parties as the source of misunderstanding and underlying belief that their own language is 'superior' and 'correct'.

Having identified these differences, the OSS and BSS EAs agree to run an experiment to raise mutual cultural awareness, by placing 'ambassadors' from each in the others space. They are briefed to reflect back to the foreign organization the unstated assumptions they are making and the implications for their own people. It becomes apparent

that the root of Products seeming inability to systematically follow process was fear of a loss of creativity and as a result, nothing short of their identity. It also becomes apparent that the gated governance process is over-prescriptive and unnecessarily risk-averse, ultimately slowing down the agile, iterative development needed to keep CSI ahead in the marketplace and rapidly refine the products to respond to emergent need.

This illustration shows how cultural and social factors can be brought in to view as explicit parts of the system. It shows that, when they are, these factors can be systematically engaged with in a way that integrates with engineering disciplines. Fortunately, there are a number of structured, analytical ways of defining and understanding 'culture' that have developed over the last few hundred years. Probably the best (and oldest) articulation of such a model comes from a reputable source – Jean Jacques Rousseau. He first proposed an analytical model to represent culture in the eighteenth century:

The next figure provides a good example of the working through of

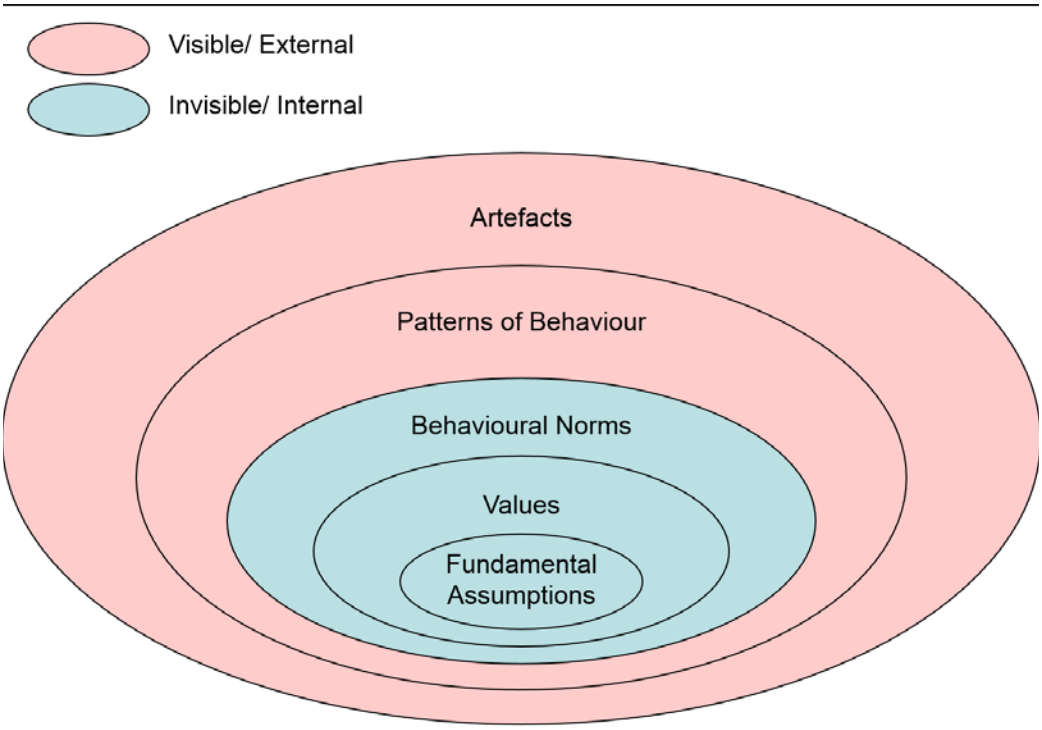

Figure 2 – Rousseau's Layers of Culture

this model, in combination with use of a 'Rich Picture'. This was put together as part of an initiative to capture the essence of a school as a basis for planning improvement. While not specifically part of a process consciously using SSM, it also demonstrates how SSM, like many methods, is formed through f the appropriation and 'packaging' of concepts and practices that people often use quite intuitively:

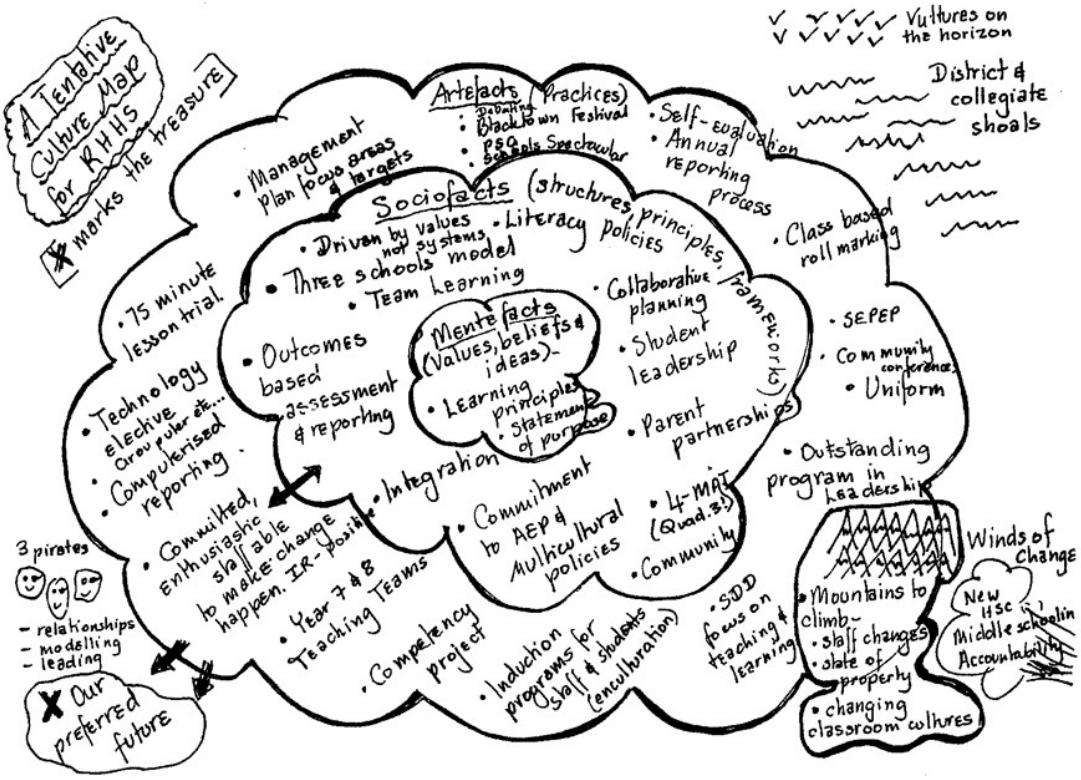

Figure 3 – Layers of Culture worked through in practice

Enterprise Architects would also find much transferrable wisdom in the book '101 Things I Learned at Architecture School' [Ref 5]. Particularly relevant to this Paper is the observation the author makes that "Engineers tend to be concerned with physical things in and of themselves. Architects are more directly concerned with the human relationship with physical things.". Enterprise Architects have to cover all these in two principal domains: the first, in relation to the 'systems' they are charged directly with changing, whether this is through target architectures, intermediate architectures, roadmaps, programmes or projects. The second, in relation to the 'systems' they are not usually charged directly with changing – themselves as individuals and the Enterprise Architecture function itself, including engagement with the social and cultural stakeholder environment in which it operates.

In both domains, adding the cultural dimension in as an identifiable part of the system under consideration enables the Enterprise Architect to make the invisible visible. Doing this provides a complementary extension of the mechanical 'hard systems' engineering approach in a way that appeals to an analytical mindset to bridge 'hard' and 'soft' worlds. Adding this ingredient to the mix catalyses a more organic problem solving approach, and amplifies the effectiveness of the Enterprise Architect.

White Paper number 4:

White Paper number 4 in this series responds again to feedback received on the previous paper to give more space for detailed consideration of two further major areas of difference between SSM and an engineering approach. These are:

- **Structures, Processes and Attitudes** – the way in which SSM is inclusive of change to structures, processes and attitudes as a means of delivering improvement to a situation, while an engineering approach excludes attitudes, or takes attitude change as given.
- **Accommodation and Consensus** - SSM seeks problem and 'solution' definitions, actions and commitment to change that stakeholders can live with, rather than that they all agree on. SSM calls this 'Accommodation' between differing views.

White paper number 5 will then start to explore the structure approach that the Soft Systems Methodology provides to guide practitioners, and the way in which this affords integration points for blending with engineering disciplines such as TOGAF.

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 directly to Orbus Software or via my email at: ceri.williams@theintegrationpractice.co.uk.

References:

[1] Enterprise Architecture meets Soft Systems Series, Papers 1 & 2. Orbus: http://www.orbussoftware.com/resources/authors/ceri-williams/

[2] Weinberg, Gerald M: An Introduction to General Systems Thinking. ISBN 0-932633-49-8

[3] 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

[5] Frederick, Matthew: 101 Things I Learned at Architecture School. ISBN 978-0-262-06266-4

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

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

orbus software