White Paper

How to include attitudes in Enterprise Architecture and why consensus is not important

WP0161 | August 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 how Soft Systems Methodology (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. It described 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.

This fourth paper in the series continues with examination in more detail of a key area where the Soft Systems Methodology (SSM) complements the traditional engineering approach to Enterprise Architecture. It considers two further major areas of difference between SSM and an engineering approach. These are:

- 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.
- 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.

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 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 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 and 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 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].

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.

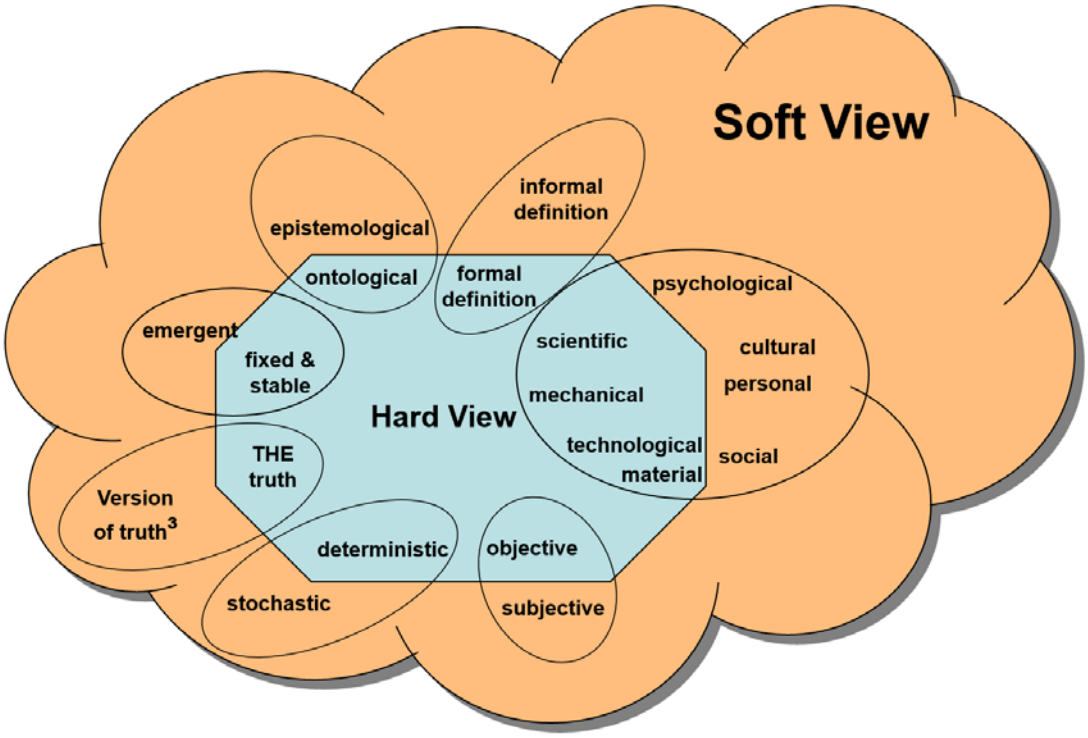

Figure 1 – The Relationship between Soft and Hard System viewpoints

#	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 and 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

Accommodation and Consensus

The engineering (or 'Hard Systems') approach considers multiple stakeholder viewpoints as filtered views of a single, objective, canonical definition of a system or problem. It assumes and requires common agreement across all stakeholders as well as convergence and consistency of viewpoints to establish and maintain coherence.

This view is embodied in the IEEE 1471 standard and adopted by most EA frameworks, including TOGAF and MODAF. This works well for systems that can be ontologically modeled and consist predominantly of inanimate objects that can be designed and directed. However, it is wholly inadequate for complex systems that are inclusive of people and the beliefs and values that drive their behavior in response to situations. There is some accommodation of this in the way that IEEE 1471 differentiates between 'Projected' views where all views are guaranteed consistent as they are projections of a single, canonical information set, and 'Constructed' views where the views are created in separate environments, without significant reference to each other, to meet the needs of a specific set of stakeholders. The clear suggestion though is that 'Constructed' views are not desirable (and should therefore be minimized) as they lack model coherence, but do have to be acknowledged as the way the world works and the way that models emerge from different areas at different times.

A direct consequence of this belief is that consensus has to be achieved across Stakeholders as a pre-requisite to meaningful action. SSM allows for consensus, but does not require it as a pre-requisite in the same way. It recognizes the significant challenges involved in reaching consensus and proposes a more real and feasible objective – that of reaching accommodation.

The dictionary definition of 'accommodation' is "to make room for" or "to make fit, suitable or congruous". In the SSM environment, this means that practitioners are better off seeking 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. This emphasis on tolerance means that it is easier to bring disparate stakeholders together in the first place, and then easier for them to get to the point where action can be agreed. 'Tolerance' in this context means the "willingness to accept feelings, habits or beliefs that are different to your own".

Tolerance also has a specific meaning in the engineering domain which is of value here in integrating the soft and hard worlds. Engineering has long worked with the concept of tolerance as the ability of the component parts of a system to work together without the need for a precise fit. Consider the difference between road vehicles and locomotives – roads are highly variable surfaces, rails are far more

consistent and predictable. This means that a car must be 'tolerant' to the variability of the road and does so through the use of rubber tires and steering. A locomotive on the other hand can assume a precise track, and focus more on speed rather than smoothing out track variation or steering.

The concept of tolerance is a significant connection between SSM and engineering approaches and provides an opportunity to bring the SSM concept of accommodation into the engineering domain for integration with the more precise methods on which engineering depends.

As an example based on our fictitious Telco (see previous papers) Complex Systems Inc., the Business Support Systems (BSS) Enterprise Architect finds that it is possible to unite diverse stakeholders around a high level concept of a 'Billing' capability or service. Working at the conceptual level means there is ambiguity, and therefore a high degree of tolerance. At that level of description, there can be a consensus, although practically it is only an apparent consensus because it is based on a variety of interpretations of what 'Billing' means. Operating at this level and the use of industry standard reference models (e.g. eTOM) is good for identifying common ground and bringing stakeholders together, however it is not enough to define action to improve such as rationalization of processes and systems. The more difficult step comes next, when the process of unpacking and defining Billing quickly shows that some stakeholders consider it to include the rating and monetizing of the metered usage and others believe it to be just about administering the bills and payments post-monetization. In addition, the first set of stakeholders are not of one view as some believe that rating and monetizing also includes management of the reference data that enables the calculations, while another subset believe that this is the job of 'Product Catalogue Management'.

I would argue that not only is the use of accommodation the most desirable objective of stakeholder integration – it is the only feasible approach. Consensus at best is a bonus, a beneficial side-effect of the type of exploration with stakeholders designed to develop accommodation. At worst it is an illusion – more an expression of hope than reality. Consensus cannot be 'driven through' and even if it could, would take too long for most situations the EA finds himself in. No matter how much stakeholders may appear either by active confirmation or a passive stance (interpreted as assent) to have reached a consensus, it quickly unravels through the passive-aggressive behavior that follows during implementation. There is always ambiguity in any apparent agreement that can then be exploited in the absence of accommodation. As an example, while it is likely that an organization can drive through conformant behavior on safety procedures, the same is not true of system design processes. In fact, many features of design methods deliberately facilitate ambiguity as a way of providing controlled flexibility

and resilience – differentiating between logical and physical architecture is a good example of this.

The EA is typically facilitating exploration and courses of action that need to persist over long timescales (years) and across broad business and technical domains. These courses of action have to be resilient to changes in the environment and churn of key personnel. This means that consensus cannot be a firm foundation for action as it is typically brittle and shatters on change of influential personnel who are advocates or champions. Accommodation however is naturally resilient and inclusive of new and emerging stakeholder viewpoints.

For this, and other reasons, the EA needs to develop awareness and techniques to promote the search for and identification of accommodations. These will be covered in more detail in later papers in this series. At this point however, it is worth highlighting a couple of key concepts based on the Layers of Culture model described in the previous white paper [1]. To provide some structuring of this pretty fuzzy area, the proposition here from the perspective of an Enterprise Architect is that accommodation and consensus can be specifically focused on these layers, where differences and similarities occur and can be actively sought.

Figure 2 provides a reminder of the layers that enable the EA to unpack 'Culture' to enable its components to be systematically worked with the key features of the next section: Structures and Processes (Visible/ External) and Attitudes (Invisible/Internal).

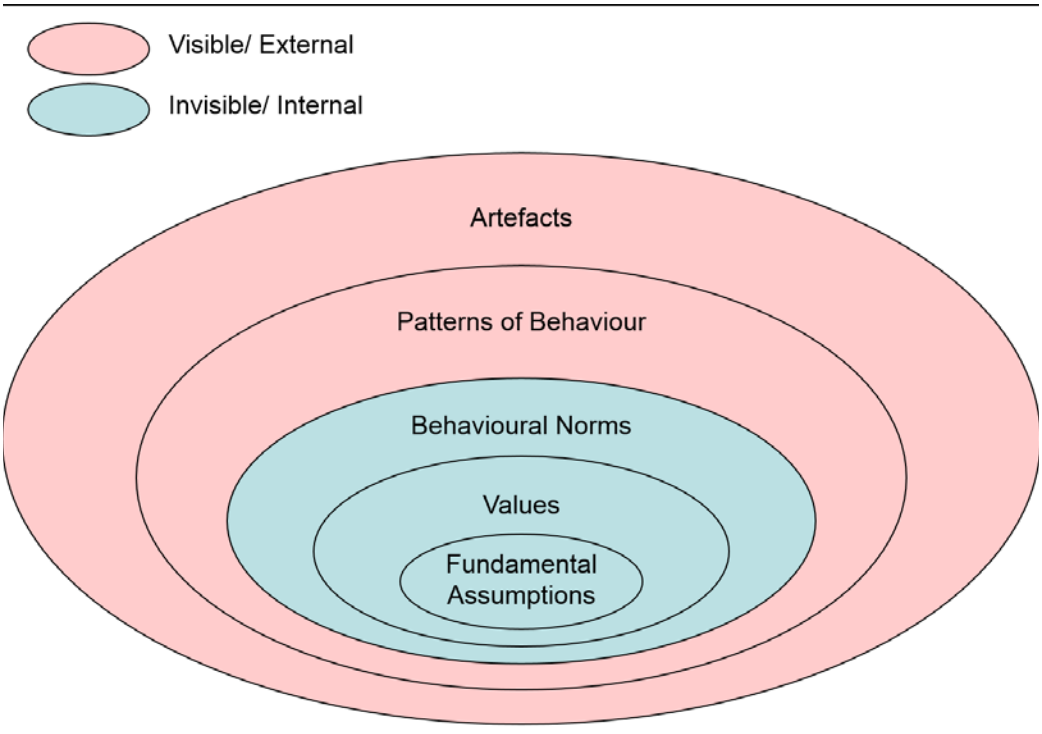

Figure 2 – Rousseau's Layers of Culture

Figure 3 suggests how these provide a structure within which to explore where accommodation and consensus operate. It proposes zones to enable the EA to make a conscious differentiation between accommodation and consensus as each requires a distinct and specialized response to develop. Seeking accommodation is a more realistic objective as well as being a pre-requisite for consensus, and is therefore the best starting point for an EA tasked with identifying and realizing synergies across business domains.

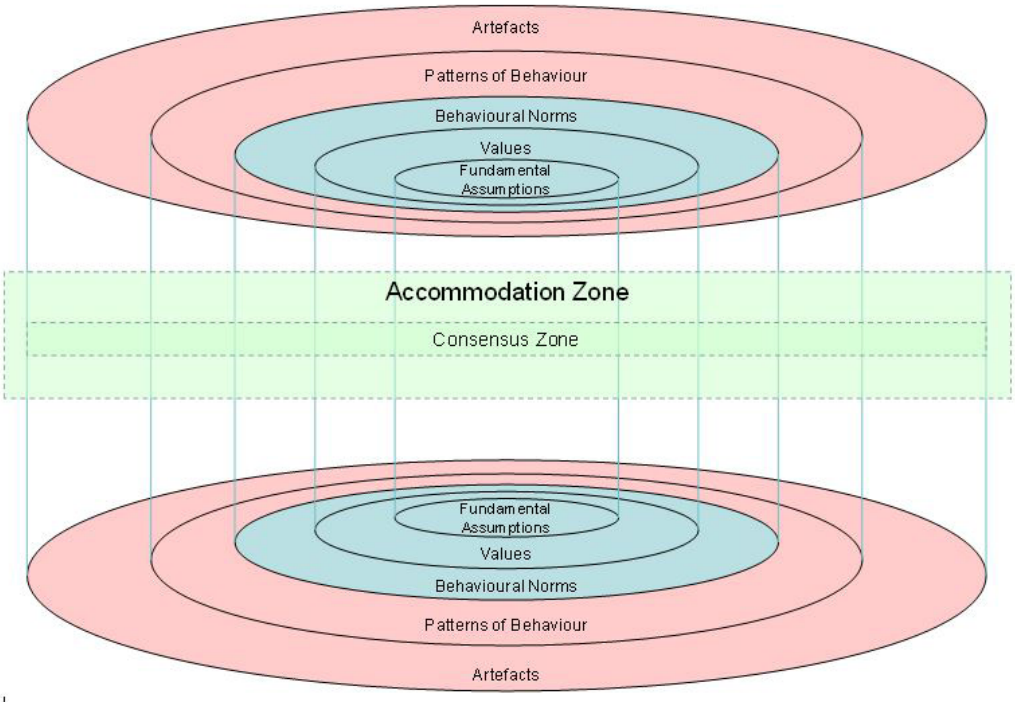

Figure 3 – Layers of Culture to structure accommodation and consensus building

Structures, Processes and Attitudes

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. To date, attitudes have not lent themselves easily to modeling either in themselves or in a way that integrates with the ontological modeling of real-world structures and processes. An implication of SSM is that EAs need to find a way to make the important able to be modeled, rather than making important the things that are just easy to model. Attitudes are important as they provide key enablers and fundamental constraints on the success of any attempts to plan and build flexibility into an enterprise's technology or to drive efficiencies through convergence.

The vast majority of organizational change (whether by design or as an unintended consequence) focus on the things that are easiest to model and change on paper. More often than not a new 'Target Operating Model' for any part of the business starts with a new organigram rather than considering the capabilities and services that the organization is there to deliver – least of all the attitudes needed to make it all work.

Structure can be changed (at least on paper) by decree, through the exercise of legitimate power and as a result, large organizations typically re-organize every 18 months – some IT functions every year. As an example, between its creation in 1948 and 1980 UK governments imposed structural change on the NHS more than 20 times.

These re-organizations are normally driven by the arrival of new senior managers (or governments) wanting to (and incentivized to) make rapid and significant improvement. This motivation, combined with the innate need to exercise power and make quick, visible change (although not necessarily improvement) often leads to a paper-based re-organization. The order in which change is planned is typically the reverse of what is needed as it progresses: structure, then process, then (if at all, usually not) attitudes. I propose that all these need to be dealt with concurrently and iteratively, and that considering them a sequence actually prevents delivery of the 'biggest bang per buck'. For example, if the re-organization draws functional boundaries that cut into naturally cohesive clusters of capabilities and services, the resulting inter-functional complexity and close coupling actually prevents the organization from functioning.

This focus on the map rather than the territory delivers poor results as well as a being poor basis for future decision making. Polish-American scientist and philosopher Alfred Korzybski remarked that "the map is not the territory", encapsulating his view that an abstraction derived from something, or a reaction to it, is not the thing itself. Korzybski held that many people do confuse maps with territories - that is, confuse models of reality with reality itself. This pitfall is particularly relevant for anyone practicing an approach that is based on an Engineering discipline – Enterprise Architecture, along with general business organization decision-making, is very prone to it.

The EA industry is rich in terms of concepts, techniques and tools to understand and plan change to structures and processes. In fact, it is over-rich in the competing options and variations available to the EA. These help to make visible and integrate models which can spot the type of problem highlighted above and help draw organizational boundaries in the best place. They can also help the organization shift attention quickly on from structure to process – that by itself is a significant benefit. However, more is needed to deal in parallel with the attitudes needed of the individuals to implement the new structures and processes as they were conceived.

Again, IEEE 1471 provides some opportunities to shape a holistic and inclusive approach to EA. An enhanced version of IEEE 1471 could include the means through which to capture and model the fundamental assumptions, values and behavioral norms from Figure 2. Doing this would make the SSM 'World Views' an explicit item of the Enterprise model, bringing it into the center ring rather than it being a side-show.

Only then can it be worked on systematically as the enabler of business improvement. Even then, it is still not feasible to 'design' or 'direct' attitudes, as people are complex and autonomous, however just making attitudes visible and legitimizing their consideration is a major step in the right direction.

White Paper #5:

White Paper #5 starts 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 such as INCOSE and 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 direct to Orbus or via my email at: ceri.williams@theintegrationpractice.co.uk.

References:

[1] Enterprise Architecture meets Soft Systems Series, Papers 1-3. Orbus: http://www.orbussoftware.com/resources/authors/ceri-williams/

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

[3] Checkland, P: Soft Systems Methodology www.yhcsleadership.co.uk/download-file/43

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