

# White Paper

## Crafting Root Definitions and Purposeful Activity Models

WP0185 | April 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 [\[Ref 1\]](#) consider the analyses associated with the first couple of steps in the Soft Systems Methodology:

- **Analysis One:** shaping the SSM intervention and mobilizing the stakeholders
- **Analysis Two:** understanding and defining the ‘social texture’ including roles, norms and values
- **Analysis Three:** exploring the politics and understanding the disposition, expression and management of power

This white paper builds on these analyses to focus on the crafting of Root Definitions of the problematical situation and the development of conceptual models of the systems concerned. It examines the building of Purposeful Activity Models, taking the Enterprise Architect into more familiar modeling territory, but with a few interesting extensions in preparation for the designing and implementation of feasible and valuable change in the enterprise environment. It also considers in more detail the ways in which standard EA frameworks could be adapted and extended to incorporate key SSM features.

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.

Access our **free**, extensive library at  
[www.orbussoftware.com/community](http://www.orbussoftware.com/community)

# 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 still relevant, the best option 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 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 convenience 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.

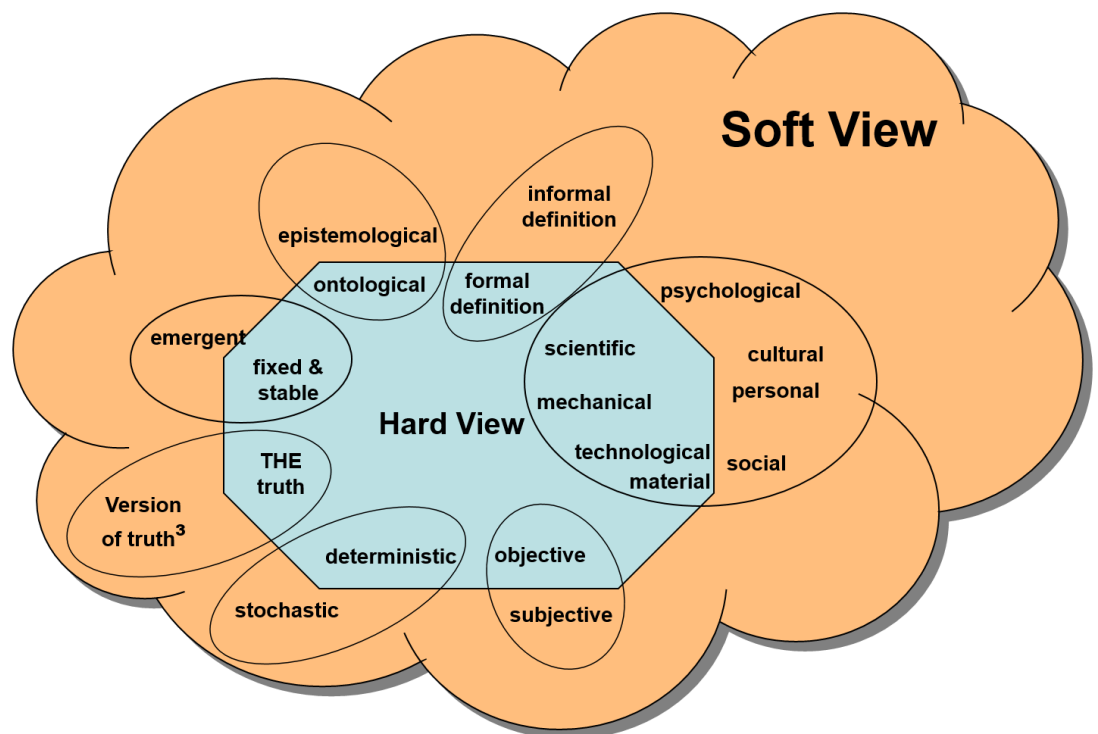
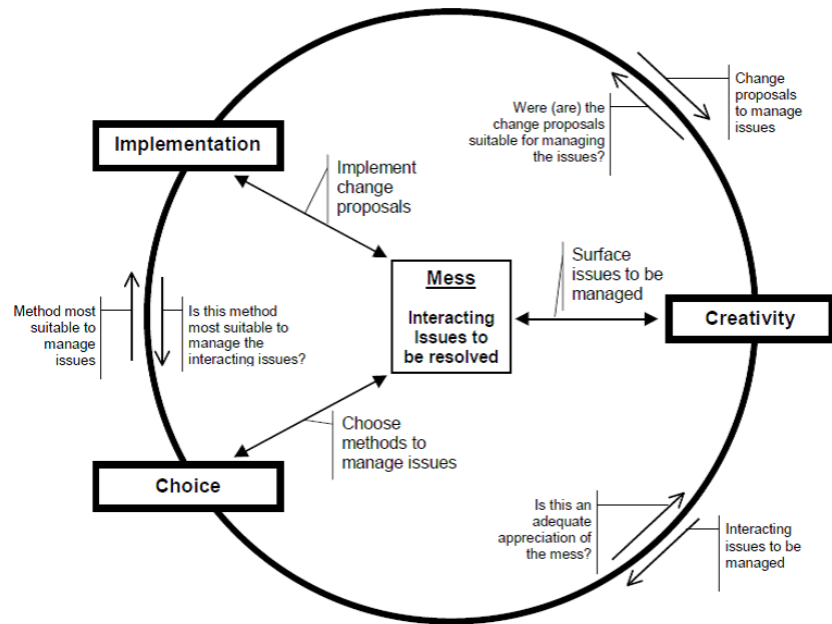


Figure 1 – The Relationship between Soft and Hard System viewpoints

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**

This white paper focuses on the crafting of Root Definitions of the problematical situation and the building of Purposeful Activity Models as the basis for Steps 3 and 4.

Before diving in to SSM specifics, it is worth zooming out and putting SSM in context. SSM is an example of a Problem Structuring Method (PSM) – an entire field of study in itself. For an Enterprise Architect conversant with the concept of 'Meta', the field of PSMs can be considered the next Meta level up from SSM. The significance of this is that it is as important that the EA (and increasingly most close stakeholders) is skilled and has the capability to be a Methodologist – as in, he is conversant with the variety of frameworks and methods for practicing Enterprise Architecture. To be effective, the EA has to integrate multiple, dissimilar, special-purpose frameworks and methods to practice the creation of Enterprise Architectures and realize them through delivery programmes and projects. Previous white papers in this series have explored the value of SSM working itself at a 'Meta' level to facilitate the integration of approaches optimized for 'soft' challenges with those optimized for 'hard' (i.e. engineering) challenges. The field of PSMs provides for even more capability to position SSM alongside other holistic and specialized approaches such as Strategic Options Development and Analysis (SODA) typically associated with the field of Operational

Research (OR). [Ref 3] provides a critical and comparative account of some mainstream approaches.

One advantage for the EA of positioning as a methodologist, is that it weakens the grip of the kind of orthodoxy and zealotry surrounding competing frameworks and methods that often alienates stakeholders and prevents the EA from delivering value. It does this by encouraging the EA and stakeholders to acknowledge and accept that their particular 'world views' are equally legitimate, and that the methods they employ are good at some challenges but not at others. It promotes acceptance that one size does not fit all and that the integration of methods enables systematic collective effort and better understanding of each-other's worlds. For readers with enough academic concentration and patience, [Ref 4] considers the challenges of 'Multi-Methodology' environments in great detail and in a rather mind-bending fashion, starts to create a form of methodology for analysis and exploitation of methods.

While recently the 'System of Systems' approach has been appropriated by the engineering community, it was originally conceived by Jackson & Keys in the 1980s to include many of the challenges and features of SSM. Specifically the way in which complexity is a product of perspective rather than inherent in the subject – inclusion of views and interests. Linking these ideas with Multi-methodology provides some useful concepts for the Enterprise Architect when positioning his purpose and practice in the context of other stakeholders. Figure 3 illustrates these concepts as a set of dimensions that can help categorize the methods, suggesting where the challenges of method integration may lie – also useful for characterizing the context of the problematic situation to facilitate a matching of method to context:

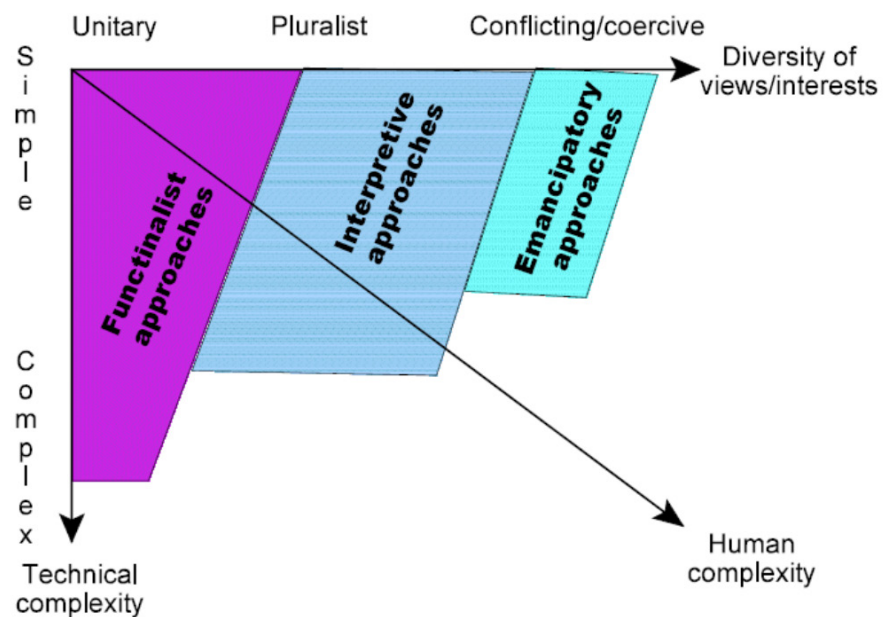
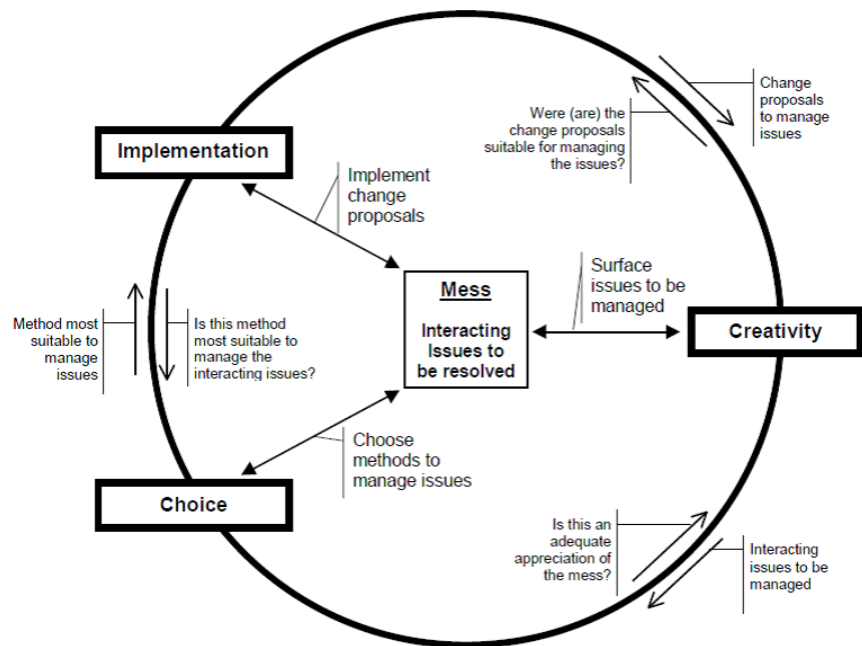


Figure 3 – Problem & Method Categorisation Dimensions

Total Systems Intervention (Flood & Jackson) emerged in the 1990s, building on System of Systems concepts as a framework for systematically understanding the problematical situation in a way that facilitated not only the design of the intervention, but selection of the Method(s) by which the intervention can be shaped. In fact, while this is presented as a progression from SSM, it is more of a representation of SSM concepts – in particular Analysis One which designs the SSM intervention itself. That said, TSI concepts are more easily assimilated into the analytical/engineering arena through positioning the selection of methods as something that itself can be addressed by a method. Figure 4 illustrates the main TSI concepts, continuing the SSM theme of dealing with ‘messy’ situations:



**Figure 4 – Total Systems Intervention**

The significance of framing of this white paper in this way is that the crafting of Root Definitions and development of Purposeful Activity Models within SSM share a number of features of approach and concept with other more familiar and traditional disciplines such as requirements analysis and the Model Driven Architecture (MDA). The sharing of these features is helpful in recognizing existing capabilities that the Enterprise Architect may possess that can be drawn on to understand and practice SSM as well as sensitizing the EA to notice similar capabilities in the stakeholder community (e.g. Project Delivery, Solution Architecture) to enable the meaningful integration of EA with other disciplines (as well as one EA framework with another).

It is always helpful to understand a new concept in terms of an old one. To assist with understanding, it is helpful to conceive of the crafting of Root Definitions as a form of requirements capture, analysis and definition and Purposeful Activity Models as a form of model building in general – familiar to anyone who has built models using MDA, MoDAF, UML or ArchiMate as the base languages.

In a nutshell:

- **Root Definitions** are used to identify what the system's purpose is, who the interested parties are. It concisely captures what the system is, how it will work and why it is needed.
- **Purposeful Activity Models** are used to say how the system would function, i.e. what activities are necessary for it to take place. The situation is examined by defining systems (inputs-transformations-outputs) which are relevant to it and working out the activities that will be necessary for such a system to function.

Much of the difference between the two arises from the degree of detail and precision, rather than fundamental differences of concept. From this perspective, a useful enhancement to SSM is to conceive of Purposeful Activity Models as the first iteration of a recursive unpacking of the Root Definition that contain the same elements, but just in greater resolution.

## Crafting Root Definitions

To reverse the usual format for expressing a theory, first, some examples of Root Definitions are included below, with the theory to follow:

1. An Enterprise Architecture function is a system owned by the Chief Information Officer operating with a systematic and repeatable culture to deliver coherent direction and governance through delivery programmes and projects by means of engagement and decision support within people and financial resource constraints in order to achieve medium and long term business efficiency, flexibility and effectiveness
2. A university is a government owned and operated system to award degrees and diplomas to suitably qualified candidates, by means of suitable assessment, (in conformance with national standards), in order to demonstrate the capabilities of candidates to potential employers.
3. A university is a privately owned and operated system to implement a quality service, by devising and operating procedures to delight its customers and control its suppliers, in order to improve its educational products.
4. A drug procurement function is a hospital owned system which provides records of spending on drugs so that control actions, by administrators and doctors to meet defined budgets, can be taken jointly.

Crafting of the Root Definition is the first (or at least an early) step in the creation of perceived order to facilitate understanding of the problematical situation and the interventions to disturb it. Root Definitions set the direction of what happens next (Purposeful Activity Models) and are critical devices for scoping. Consider how different the Purposeful



Activity Models would be that emerge from a Root Definition of a prison as a) a rehabilitation system; b) a punishment system; c) a system to protect society; d) a system to train criminals; or e) a people warehouse.

The structure and syntax of a Root Definition follows a recurring pattern of the form: **What, How and Why**. For example: A System to do X, by (means of) Y, in order to achieve Z.

X – What the System does

Y – How it does it

Z – Why it is being done

The 'What' is the immediate aim of the system, the 'How' is the means of achieving that aim, and the 'Why' is the longer term aim of the purposeful activity. This is enriched by the CATWOE extension that provides memory-joggers for the SSM practitioner to consider Customers, Actors, Transformations, Worldview, Owners and Environment.

CATWOE refines the basic Root Definition to become: A System owned by O with a culture of W to do X through A by means of T given the constraints of E in order to achieve X for C. Any Enterprise Architect that has had the pleasure of being involved in crafting mission or vision statements will recognize this type of syntax. Root Definitions, like mission and vision statements suffer greatly when written by a committee. They risk ending up as long winded supersets of the incompatible viewpoints of all participants, rather than concise, memorable phrases that result from forcing participants to make hard choices and accommodations. This process is needed to bring the diversity of views to within a tolerable range that provides a sound basis for development of Purposeful Activity Models and subsequent action.

At this point, it is worth noticing that SSM suffers in the same way as any other concept that uses the interrogatives WWHWWW (i.e. Why, What, How, When, Where, Who) as a basis for definition. While Who, When, Where do possess some (near) absolute defining qualities, Why is actually a way of putting What and How in the context of a Who in order to express benefits or drawbacks, while What and How are very much not absolute in their definition – a practitioners How is in fact a methodologists What. Provided the participants are not too wedded to the absolutes here, the concepts work well enough. The point here, which is also recognized by SSM, is that Root Definitions are always written from a particular perspective. They can be Primary Task based relating to basic tasks and structures (like examples 1 and 2 above) or Issue based relating temporary or qualitative concerns or concerns of judgment (like example 3 above).



While requirements analysis has been a significant component of systems development for decades, rapid iterative methods have brought it into sharp focus over the last 20 years. DSDM Atern is a good and mature case which is used here to represent requirements capture/analysis/definition in general. DSDM Atern adopts a similar syntax approach to defining requirements as SSM does for the crafting of Root Definitions (see [Ref 5]). DSDM Atern defines a requirement as "...a service, feature or function that the user wishes the solution to perform or exhibit. Different users may have diverse, even conflicting, perspectives on which requirements should be included and their relative priorities". Acknowledgement of the presence of multiple perspectives represents another similarity between DSDM Atern and SSM – and that they are often apparently in conflict.

This is an area where SSM can learn something from DSDM Atern on the crafting of requirements (aka small-scale Root Definitions) in the way in which it proposes a series of iterative activities as the backbone for the crafting: Elicitation, Analysis, Validation and Management. Elicitation identifies the requirement through stakeholders; Analysis refines the requirement to become realistic, unambiguous and consistent with others; Validation seeks confirmation from stakeholders; Management attends to the business of managing the life-cycle of the requirement. DSDM Atern uses User Stories in a similar fashion to SSM to provide reminders to the practitioners, provide structure for stakeholder conversations and shape for the outputs. A typical format would be:

As a < role>...I need <requirement or feature>...So that <goal / value>

*For example:*

As a Marketing Director...I need to improve customer service...So that we retain our customers.

As an Investor...I need to see a summary of my investment accounts... So that I can decide where to focus my attention

SSM Root Definitions unpack the <role>, <requirement or feature> and <goal> to be more comprehensive and provide a richer perspective, but the concepts are very similar. For the Enterprise Architect, understanding the similarities between what stakeholders may not recognize (e.g. SSM) and what they might recognize (e.g. DSDM Atern) is a key capability that enables them to effectively practice the sort of Multi-methodology described earlier in this paper. The classic English essay topic that starts "Compare and contrast..." is trying to develop just these skills – the ability to spot similarity and difference along with the ability to make sense of it and communicate to others.

# Building Purposeful Activity Models

Purposeful Activity Models are about verbs. They are anchored by the Root Definitions. This is more familiar territory for the Enterprise Architect as there are many useful similarities with Business Process modeling. Other similarities emerge quickly on consideration of this modeling: for example, the need to define the system boundary will be familiar to anyone familiar with INCOSE Systems Engineering frameworks and Yourdon modeling. Another example is the way in which, to be intelligible and form the basis for stepwise refinement, the model will need to be built hierarchically – a concept familiar to practitioners of System of Systems approaches. Note that this hierarchical decomposition to capture refinement of detail is not to be confused with the making of design decisions that address implementation constraints – [Ref 6] gives a fine account of this.

As mentioned above, part of the job of Purposeful Activity Models is to unpack and refine the detail of the Root Definition. The link between these is illustrated in Figure 5 for the archetypal task of Gor-Tonking. The point here is to illustrate the process - even though we know nothing of Gor-Tonking, we can model the logical machine necessary to carry it out according to the Root Definition provided.

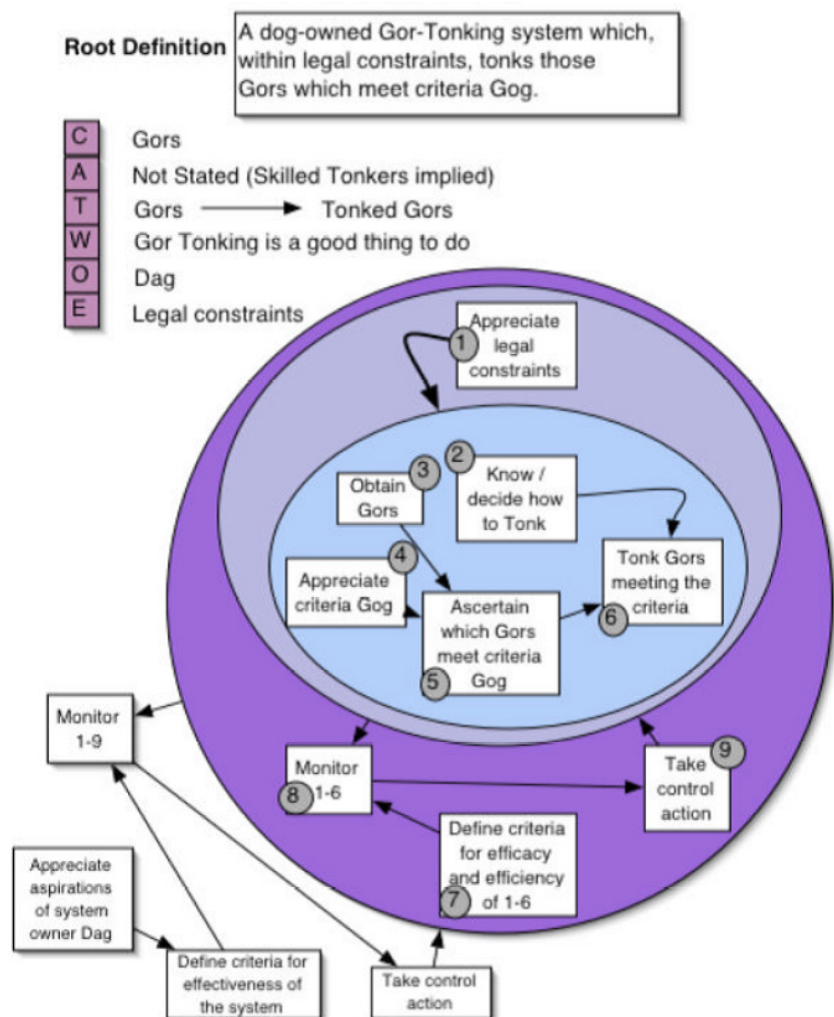


Figure 5 – The Gor-Tonking Example

In conducting this SSM process, moving from a Root Definition to a well formed Purposeful Activity Model is a giant leap. Typically the model would first be explored in an unconstrained fashion through the use of Rich Pictures. This is rather like the difference between brainstorming (unconstrained) and synthesizing of an orderly set of prioritized brainstorming outputs. The aim is to be constrained only by the Root Definition.

The Rich Picture can take any form and is not subject to the language constraints familiar to the Enterprise Architect such as MoDAF, ArchiMate and UML. It can contain any type of information such as processes, structures, values, materials and represents an attempt to understand the relationships between things from a particular perspective, or 'world view'. Figure 6 is a famous rich picture developed by the Pentagon in 2010 to explore and understand the complex situation in Afghanistan early in the insurgency in order to shape a counterinsurgency (COIN) strategy. The diagram was initially criticized for being incomprehensible, but came into frequent use as a means of providing big-picture context for more focused areas.

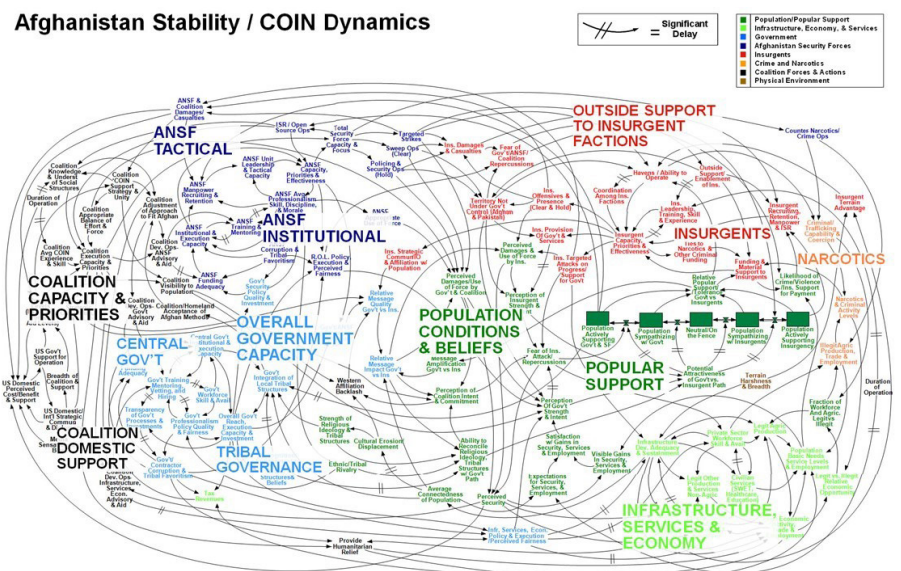


Figure 6 – Rich Picture Example

This line of criticism highlights a couple of key features of these SSM models – one is that they do not claim to be the definitive view of a situation, they are a view of a situation. Another is that they are qualified by a declared world view – i.e. that the world view (see earlier papers [Ref 1] for a discussion of world views, or Weltanschauung) is an integral part of the model. The models can only describe a range of interpretations which are relevant to debating the real-world – they only need to be defensible and are not put forward as definitive representations of the real world. This positioning is often a difficult for stakeholders to accept, especially anyone with an engineering background more used to working with definitive models from which to build Information Systems.

I hope you have enjoyed this white paper and the EA meets Soft Systems series. 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](mailto:ceri.williams@theintegrationpractice.co.uk).

## References:

- [1] Enterprise Architecture meets Soft Systems Series, Papers 1-7.  
Orbus: [www.orbussoftware.com/resources/authors/ceri-williams/](http://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] Rational Analysis for a Problematic World Revisited: Problem Structuring Methods for Complexity, Uncertainty and Conflict, 2nd Edition. Wiley; Jonathan Rosenhead (Editor), John Mingers (Editor); ISBN: 978-0-471-49523-9
- [4] Multimethodology: Towards Theory and Practice and Mixing and Matching Methodologies; John Mingers (Editor), Anthony Gill (Editor); ISBN: 978-0-471-97490-1
- [5] DSDM Atern: Requirements - <http://dsdm.org/dig-deeper/book/15-requirements-and-user-stories>
- [6] Parnas, D L: On the criteria to be used for decomposing systems into modules. Association of Computing Machinery Inc, 1972

© Copyright 2015 Orbus Software. All rights reserved.

No part of this publication may be reproduced, resold, stored in a retrieval system, or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Such requests for permission or any other comments relating to the material contained in this document may be submitted to: [marketing@orbussoftware.com](mailto:marketing@orbussoftware.com)

### Orbus Software

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

+44 (0) 870 991 1851  
[enquiries@orbussoftware.com](mailto:enquiries@orbussoftware.com)  
[www.orbussoftware.com](http://www.orbussoftware.com)

