Documents as Information Artefacts in a Model Based Systems Engineering Methodology

Paul W. Logan[1*], Dr David Harvey[2]

[1]Empel Solutions Pty Ltd
Eltham Australia

[2]Aerospace Concepts Pty Ltd
Fyshwick Australia

[*] E-mail: pwlogan@gmail.com

ABSTRACT

Much has been said and written recently about the emerging model-based methodologies replacing those that have been developed and applied within the traditional document-based paradigm. The commentary is often cast in disparaging terms, giving a listener or reader the impression that, with deference to George Orwell [1], 'documents bad, models good'[1]

This is an unfortunate and unnecessary position advanced in advocacy of model-based methodologies, because documents remain an essential part of model-based system definition, design, implementation and maintenance. While use of models is demonstrably advantageous throughout a system's lifecycle, with those models being the 'source of truth', documents remain the primary means of examining, distributing and confirming that truth.

The meaning of 'document', that is, the significance of an artefact that is electronic rather than paper, may be changing. Nonetheless, documents remain the principal means for most stakeholders to view and interpret the data contained within the model.

This paper examines the need for and use of documents within a model-based systems engineering methodology. The nature of documents in an automated system analysis, definition and design environment is described together with how document templates are used as inputs to structure the data model schema, and how the documents are auto-generated as complex views on the model. The essential need for documents as information artefacts in gathering, maintaining and reporting the 'truth' within a model-based paradigm is demonstrated.

KEYWORDS: document; model; template; artefact; source.

1. INTRODUCTION

'Document' in the English language is grammatically both a noun and a verb. Until the digital age, the dictionary definition of the noun form of document was (or words similar): *a written or printed paper furnishing information or evidence, a legal or official paper* [2].

For much of history, a document consisted of its content – information, or, to be precise, data of some kind – appended to a physical medium. The data was written or printed using some form of ink and the medium to which the ink was applied was most often some form of paper. Thus *document* and *paper* were and continues to be used synonymously as it was rare that a document was not on a form of paper. So colloquially: "The immigration official wanted to see my papers."

Paper, the medium, does not constitute a document until data, the content, is applied. But the nature of the medium shapes the presentation and persistence of the document content. Historically documents have been managed based on the characteristics of the medium: document handling and storage facilities, processes and

[1] Orwell, George, *Animal Farm*, "Four legs good, two legs bad"

techniques are concerned with preservation of the medium, that is, the paper and the ink.

The enduring persistence of paper and ink has lead to legal acceptance of a document as being evidence and fact. Despite knowing that paper documents can be forged or altered, provided the providence of the document is sound, people accept a document as factual – it is an unchanging presentation of data concerning a subject of interest. This acceptance of the veracity of the content of paper-based documents underpins configuration management policy and practice in project and most system management activities [3].

While the content of both hand-written documents, that is, manuscripts, and printed documents (sometimes referred to as the product of "Gutenberg" technology [4, 5]) is accepted as being a true record, it is usually 'signed' documents that are regarded as 'evidence'. Equally, while copies of a signed document may be useful and relevant, it the document to which the signature was attached that is regarded as the 'original' and 'one true record'.

Gutenberg's technology may have revolutionised production of documents and texts ('books'), the digital computer revolutionised production of content. With Gutenberg technology also supplanted by page-image techniques, now paper-based documents are readily produced.

Nonetheless the "one true record 'remains the paper copy to which a manuscript signature has been attached. The current situation appears to be one of '*digital* documents, *analogue* endorsement' in which digital development of the content of a document is a temporary means to the paper document as the permanent end. While this state of affairs may be model-based, the situation perpetuates the "document-centricity" so decried by model-based methodology advocates.

If systems engineering practice is to move from document-centricity to model-centricity, not just model-based, then the evidentiary qualities of paper-based documents, must be equally attributable to the model itself. However, regardless of whether this is achieved or not, the fact remains that the model *per se* can not be viewed directly by human readers and documents, as human readable reports on the model, will remain a key part of model-based systems engineering. While these information artefacts may be presented visually on screen – "pixel-based", rather than paper-based – they remain an essential part of model-based systems engineering.

2. SYSTEMS ENGINEERING PROCESS

2.1 Document-centric

The system engineering process is depicted in

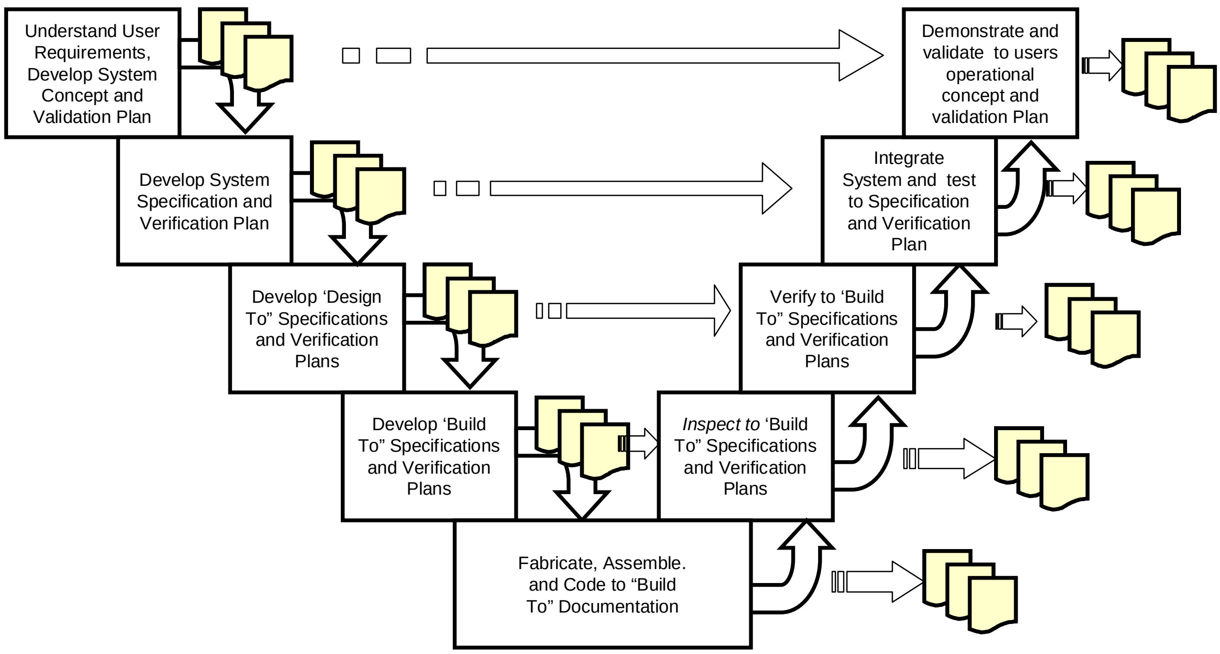

Figure 1: Systems Engineering Process (Vee Model) – Document Driven

Figure 1 using the Vee Model as an example to demonstrate the use and flow of data throughout the process. This view of the process shows that the initial "design" arm of the Vee involves the progressive refinement of system definition data and associated validation and verification plans

In this traditional method of system definition and development, the outcome of each stage of refinement of the system description is a set of documents. These documents are then the input for the next level of system definition. The documents when finalised are the source of record for the output of activity at that level and are the baseline for the next activity in system definition and this "drive" the system definition. The documents provide the criteria against which the system elements in the build phase are verified and validated.

Each set of documents is the product of the associated analysis and design activity at each level of system definition. Methods and techniques involved in developing document content can be very different, and may involve some model-based method or technique. However the models at each level of definition are essentially independent and often use different techniques and tools. The validity of document content is determined primarily by isolated review of the document content and the only connection to previous work is the traceability of requirements statements to the previous product. There may be little record of the analytic or design activity other than the documents produced.

2.2 Model-centric

A model-based and model-centric system development process is shown in Figure 2. Each of the analytic tasks, while remaining consistent in procedure and product with those involved in the document driven method, contributes to a common model repository – a relational database that captures the system elements and their relationships in a progressively detailed information model of the system of interest. Traceability between levels of detail is maintained as relationships in the model.

Documents are produced when required as reports on the data in the model repository. The documents are used to support review and further development of the system model. Model-centricity is maintained when the documents are a transitory means to validate and progress the refinement of the system model.

3. DOCUMENT USE

Documents continue to be used widely in communicating with stakeholders. They allow information gathering and information sharing.

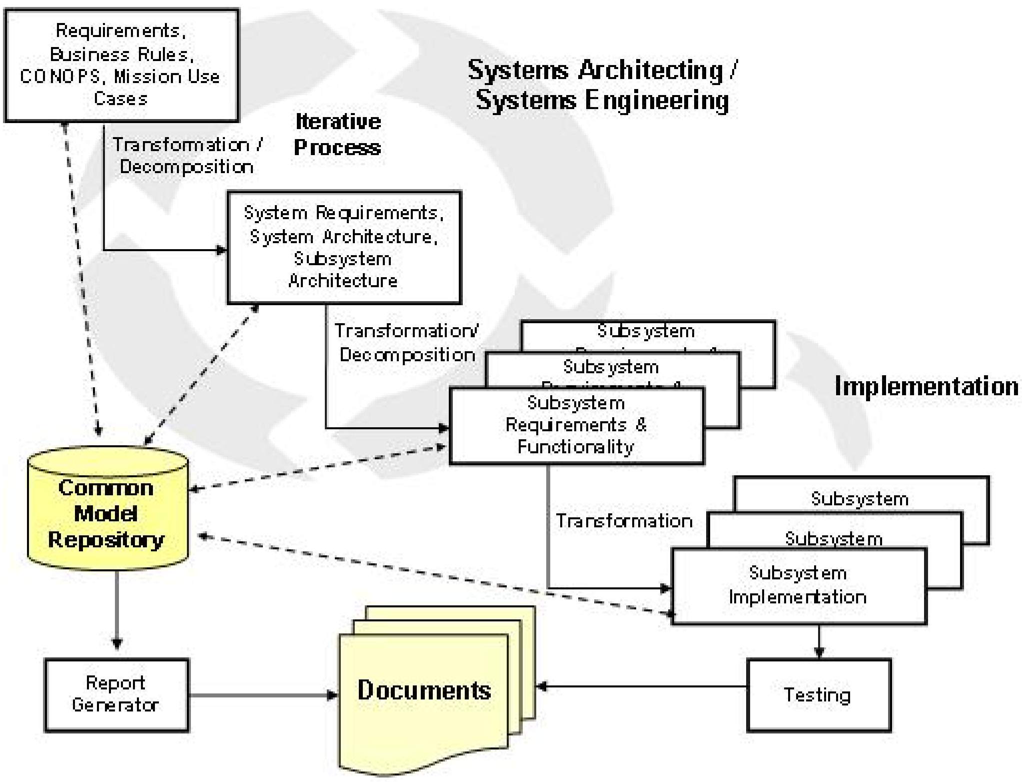

Figure 2 Model-centric systems engineering – iterative and data driven (modified from Estafan [6])

Documents may well be the only way stakeholders interact with system development process, through viewing or creating. The creation, or editing, of documents allows the stakeholder to provide information for and about the system in a way with which they are generally very comfortable. Viewing of documents may be purely for information, or may be as part of a review and approval process. The approved and signed documents (be they physical or electronic) are usually the only artefacts of the system development review process.

The use of electronic documents is on the rise and they are often mandated to be produced as artefacts in the course of model-centric systems engineering processes. There are some inherent advantages to electronic documentation in the baselining and configuration control of these artefacts. There are also further advantages to be found in storage of these electronic documents, with less space required, environmental conditions allowable more relaxed and redundancy (of identical copies) more easily achievable. Another major advantage is gained through the use of digital signatures and time stamping, which has improved over the last decade to become safer and more accurate than can be achieved with physical documents.

The move from physical to electronic documentation of engineering processes (where the two contain essentially the same information) is analogous to the rise of model-centric design in other engineering disciplines. One such example is in the field of mechanical engineering with its Computer Aided Design (CAD) [7] drawings replacing the hand-drawn (draftsperson-drawn) product drawings. In this case, the three-dimensional model of the product becomes the focus of development, but the various two-dimensional views of the part are still used for review and particularly approval purposes.

4. ARCHITECTURE DESCRIPTION

Architecture is defined as the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution [8]. The description of software intensive systems has been standardised over the last decade as the scale and complexity of systems has grown with the intent that standardised descriptions will add integration and interoperability of systems so that larger and more complex systems can be satisfactorily created and managed.

An architecture, that is, a system description, since the emergence of the seminal work of Zachmann [9] is standardised, for a particular community of stakeholders by an architecture framework, such as TOGAF [10] and DODAF [11]. These frameworks themselves have a founding structure described by the emerging international standard ISO/IEC/IEEE 42010. This standard is in final draft stage and it is the international version of IEEE Std 1471:2000 – Recommended Practice for Architectural Description of Software-Intensive Systems.[2]

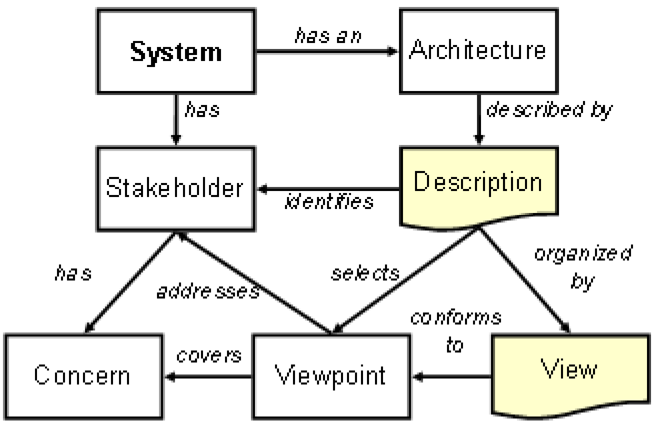

Figure 3 Architecture Description Key Elements

The essence of an architecture description created in accordance with IEEE1471 is shown in Figure 3. The purpose of the architecture description of the system of interest is to indentify both the stakeholders and their concerns and the viewpoint that addresses those concerns. A viewpoint is a way of looking at the system form the perspective of the stakeholder, for example, the engineer sees the components, their functions and interfaces while the finance manager see the cost and life of type. Each of the stakeholders has a different interest in the system and most importantly a different way of describing that interest, that is, they each have their own terms and definitions and at times a seemingly different language.

[2] As at 14 Jul 2011, *ISO/IEC/IEEE FDIS 42010, Systems and software engineering — Architecture description* has passed the final ballot cycle (Final Draft International Standard) with 21 approvals and 0 disapprovals from member bodies.

Each stakeholder's viewpoint determines the information required to address the concerns of that stakeholder. The information is packaged a view from the respective of the stakeholder. Multiple views may be required to fully cover the viewpoint that is the stakeholder concerns covered by that viewpoint.

IEEE 1471 provides the information model of an architecture description while the various architecture frameworks detail standard viewpoints and sometimes standard views to ensure that communication within a community of interest is achieved. IEEE1471 provides a metamodel construct which the frameworks further define so that information models of systems of interest can be constructed and reported in a standard fashion.

The various architecture frameworks structure and standardise the content of documents that collectively describe a system of interest. These frameworks can be applied in both document- and model-centric system engineering processes.

The effect in a model-centric methodology is standardisation of reports generated form the data repository. The various view descriptions in the frameworks become templates for the report generator in a modelling application. In essence, the frameworks make documents an essential part of the model-based paradigm.

5. AN EXAMPLE

Many systems engineers are comfortable interacting with information about a project in the form of a model or series of diagrams. However, the majority of stakeholders involved in a project are unlikely to be systems engineers. It is far more likely that these stakeholders will be more comfortable interacting with the project information using documents. These documents may be physical, or in electronic format, such as Portable Document Format (pdf) files.

Documents are capable of being utilised to structure input data for a model and as structured output from a model, providing a snapshot of a chosen sub-set of data in the model, and how it is related at a given time. Given the ease with which most stakeholders deal with documents, it is logical to use this comfort to advantage and utilise documents both for input to the model and as a means of viewing the current status of a model for review and approval.

Documents can be used as direct inputs to a model. They can be structured in such a way that a user can "fill in" the blank document as a means of eliciting initial information for a model, or for updating the information already in a model. The data contained in the model after this input will can then be automatically related to data already in the model, or other input data and the development can go ahead with all the associated benefits of traceability and robustness associated with the model-based approach.

The use of documents to structure inputs to a model has been explored by the authors and their colleagues in recent work. A blank template model has been created, which can be queried to produce a template document. This template document also serves as a map describing the model. The text in each of the sections of this document describes what information needs to developed to complete the section, how it relates to other information in the model and where it is stored in the model. This document map, or indeed targeted sections of the complete document, can be completed by stakeholders either in isolation or through interaction with the project's systems engineering team. The information is then in a format that is easy to process into the model, either automatically (through an input script or other program depending upon the Model Based Systems Engineering, MBSE, software being used) or via the systems engineer.

The document map method also has the benefit of showing the systems engineer exactly where the input information should go into the template model, as well as how it needs to be related to other information in the model. This increases the efficiency of the engineer in working with the information, providing a handy reference for those who work with the model on a day-to-day basis as well as a map for engineers who may be new to the project, or bought in as surge capacity for busy periods of work.

The other benefit of the use of documents in Systems Engineering is to allow review of the model content. Due to the nature of the MBSE approach any number of tailored documents (or views on the information in the model) can be created. This document production is automated, allowing regular querying of the current state of the model and rapid, if not quite instantaneous, production. These tailored views of the information in the model can then be shown to system and project stakeholders and they can review the validity of the information. Information presented in this familiar way to the non-specialist reviewer removes some of the mysticism surrounding the model-based approach. It allows the information in the model to be exposed to those people best-placed to review it. Once the output document generation mechanism has been created and set up, it has the added benefit of reducing the impost on the systems engineer. It allows the non-specialist stakeholders to review the contents of the model and amend, add and verify information without the direct assistance of the systems engineer (as would often be required to "walk through" the model without the document as an interface).

With careful management (such as tracking changes on the document), the user can review the information in the model and the systems engineer can then process any changes back into the model with little effort. This approach has been successfully implemented by the authors with non-systems engineering background, Defence specialists in a number of Defence capability definition projects.

Another major benefit of documents in the systems engineering process is for approval of various aspects and artefacts of a project. Documents produced from a model represent a snapshot of a subset of the information in the model at any time. The repeatable, known manner in which the information in a document is related to that in the model can be used to full advantage. Reviewers can confirm that the information in the document is correct, and this can be combined with validation that the document information content reflects that in the model, to indicate that information in the model is correct.

With the increase in complexity of MBSE models, it is becoming less likely that a non-specialist reviewer will be able to navigate the model, check the information within and certify that the model is approved. However, the linked approach of approving a document from a model, combined with certification that this is a true representation of information in the model, will continue to allow the approval of models into the future.

6. CONCLUSION

Model-based methodologies are not new – just the commentary on strengths and weakness of "model-based" versus "document-based". Sound practice of systems engineering, in fact any engineering discipline, has always had at its core a model of the system – albeit conceptual and in the minds of the system builder! The commentary on "model-based" versus "document-based" is often cast in disparaging terms. This is an unfortunate and unnecessary position advanced in advocacy of model-based methodologies, because documents remain an essential part of model-based system definition, design, implementation and maintenance. While use of models is demonstrably advantageous throughout a system's lifecycle, with those models being the 'source of truth', documents remain the primary means of examining, distributing and confirming that truth.

A key feature of the discussion on documents versus models stems from the misnomer: the issue is one of 'centricity', not 'basing'. Traditional systems engineering is 'model-based', or at least 'model-supported', while new software tools now allow the management of systems engineering to be model-centric rather than document-centric.

The meaning of 'document', that is, the significance of an artefact that is electronic rather than paper, may be changing. Nonetheless, documents remain the principal means for most stakeholders to view and interpret the data contained within the model. Documents are an essential part of "model-based" (that is, "model-centric") systems engineering.

REFERENCES

1. Orwell, George, "Animal Farm: A Fairy Story", Secker and Warburg, London, 1945
2. Macquarie Dictionary [Online, accessed 20 July 2011].
   URL: http://www.macquariedictionary.com.au/

3. Defense Acquisition University, "Chapter 10: Configuration Management" in System Engineering Fundamentals, https://acc.dau.mil/adl/en-US/38453/file/9330/SEF_3%5B1%5D.10_Configuration-Management.pdf Accessed: August 16, 2011
4. Singer, C.; Holmyard, E.; Hall, A.; Williams, T. 1958. "*A History of Technology*", vol.3. Oxford University Press
5. Nash, Paul W. "The 'first' type of Gutenberg: a note on recent research" in *The Private Library*, Summer 2004, pp. 86-96
6. Estafan "*Survey of Model-Based Systems Engineering MBSE Methodologies*" INCOSE MBSE Initiative, Figure 5.2, p55) http://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf Accessed 16 Aug 2011
7. CAD software - history of CAD CAM, http://www.cadazz.com/cad-software-history.htm Accessed August 16. 2011
8. Recommended Practice for Architectural Description of Software-Intensive Systems ISO/IEC 42010 :: IEEE Std 1471 http://www.iso-architecture.org/ieee-1471/faq.html#wh1471
9. Zachman, J. A.; , *"A framework for information systems architecture"*, IBM Systems Journal , vol.26, no.3, pp.276-292, 1987
10. TOGAF®9,URL:http://pubs.opengroup.org/architecture/togaf9-doc/arch/ Accessed: August 16, 2011
11. Department of Defense, "*DoD Architecture Framework Version 2.0 (DoDAF V2.0)*", Washington, US, 2009