Manage Changes in the Requirements Definition through a Collaborative Effort

Suja Joseph-Malherbe
CSIR
Defence, Peace, Safety and Security
P. O. Box 1177
Stellenbosch, 7599
084 909 0163
sjoseph@csir.co.za

Daniël Malherbe
DCM Consulting CC
Postnet Suite 219
Private Bag x5061
Stellenbosch, 7599
083 302 6528
malherbe@gmail.com

Abstract. Updating or changing the requirements statement during the systems engineering process may impact adversely on project parameters such as sequence, dependencies, effort, and duration of tasks, usually with an increase in development time and cost. Changes in the requirements statement that results in conflicting requirements may also have a negative impact on the system solution. Reasons for changes to the requirements statement during the systems engineering process and the origins of these changes will be discussed. Finally, this paper looks at how to manage changes to the requirements statement by using an integrated data repository in a model based systems engineering environment with a communication protocol that allows for real time collaboration where all stakeholders and the development team can contribute throughout the systems engineering process.

Introduction

Often, the systems engineering approach relies heavily on documents in configuration management distributed in hard copy to communicate information both as input to and output from the engineering effort. This results in slow communication and sharing of information. In such an environment the contribution of external stakeholders is limited by the opportunity to review documentation, usually close to a project milestone or a design review. Once a milestone has been reached or a design approved at a design review, a baseline is fixed and placed in configuration management. Subsequent changes to the requirements statement will be to the benefit of the system solution, but at an increase in project costs and resources. The systems engineering effort should ensure that the impact of the changes can be determined by upward and downward traceability in the system definition. Changes to the requirements statement can be brought about more effectively and efficiently when the development team shares the system definition with all stakeholders from an integrated data repository. All stakeholders can then evaluate the impact of the proposed changes. The paper shows how development teams in South Africa can implement a collaborative development methodology throughout the entire systems engineering (SE) process with computer aided systems engineering (CASE) tools and a communication protocol, allowing for real time collaboration.

(Carson 2001) argues that stakeholders have an interest in the system and therefore they have a need and even a right to be considered during the systems development process. Everyone who is influenced by the system's behaviour or, on their part, influences the system's behaviour are stakeholders. Stakeholders have an interest in the need and are therefore a valuable source of requirements.

This paper argues that all stakeholders, including the development team, should have an active participation in requirements formulation and refinement throughout the systems development process. We recognize that the tacit and explicit knowledge of stakeholders should be included in the requirements statement to ensure requirements are complete and can successfully form the basis of an acceptable solution. The SE process can then align stakeholders' needs with the proposed solution by ensuring the solution will perform within the requirements statement.

Requirements Change, What to do?

Requirements stem from people and people follow a specific thought process to identify a solution to address their need. Figure 1 describes the logical steps in the act of thought as defined by (Dewey 1910).

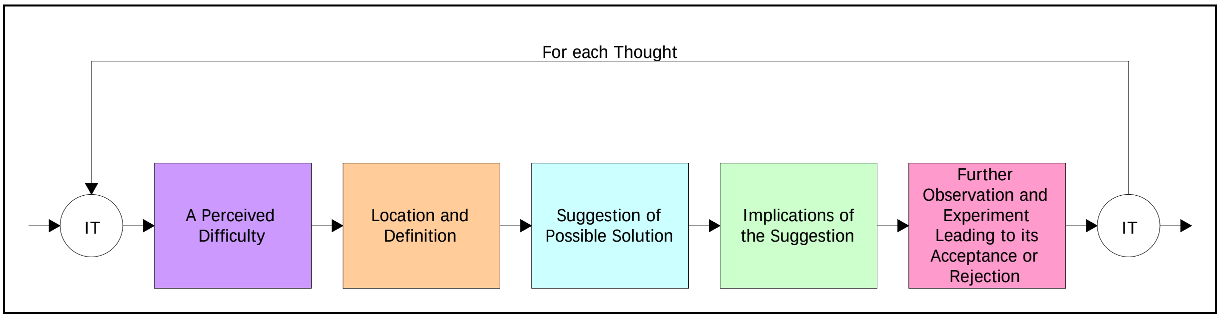

Figure 1. Reflective Thought Process

Instead of using selection to decide upon a solution, people would rather reject solutions until a suitable one is accepted. Therefore, people often do not know what their needs are, but from a number of possible solutions they will know what they don't need.

(Gilmour 2002) draws a parallel between the SE process and the reflective human thought process as shown in Figure 2. He states that systems engineering, indeed any form of human activity, has the same starting point as that of thinking. Therefore, "If the origin of thinking stems from some perplexity, confusion, doubt or perceived difficulty then it seems quite natural to take the same definition for the origin of system engineering. For if there is no problem to solve or need to be fulfilled why then perform any sort of engineering?" (Gilmour 2002)

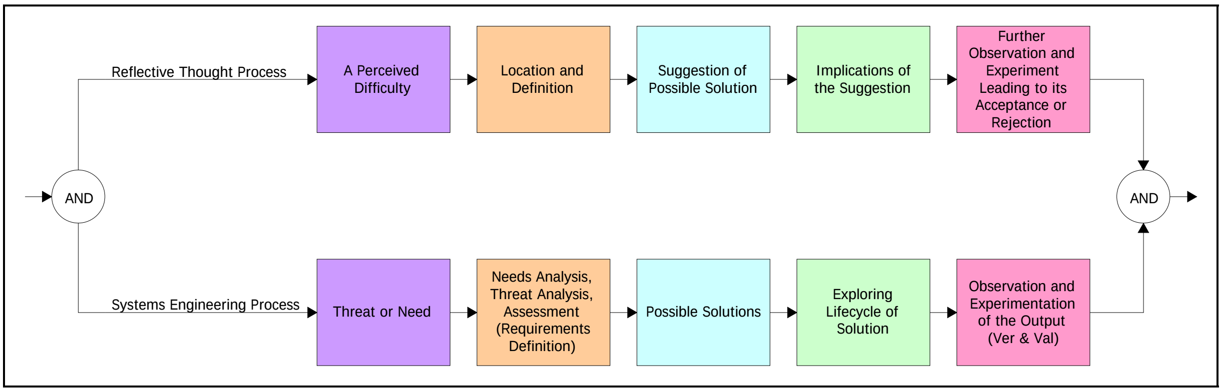

Figure 2. Comparison between the Reflective Thought Process and the Systems Engineering Process

He continues to say that, "If individuals perform disciplined reflective thinking in a systematic manner then as a collective whole they are more naturally inclined to follow the general principles of good systems engineering practice." (Gilmour 2002)

During requirements formulation, stakeholders follow the reflective thought process to define the need and scope of the project. Figure 3 below shows a simplified model of how the SE process transforms a need defined by all stakeholders into a solution that satisfies the stakeholders.

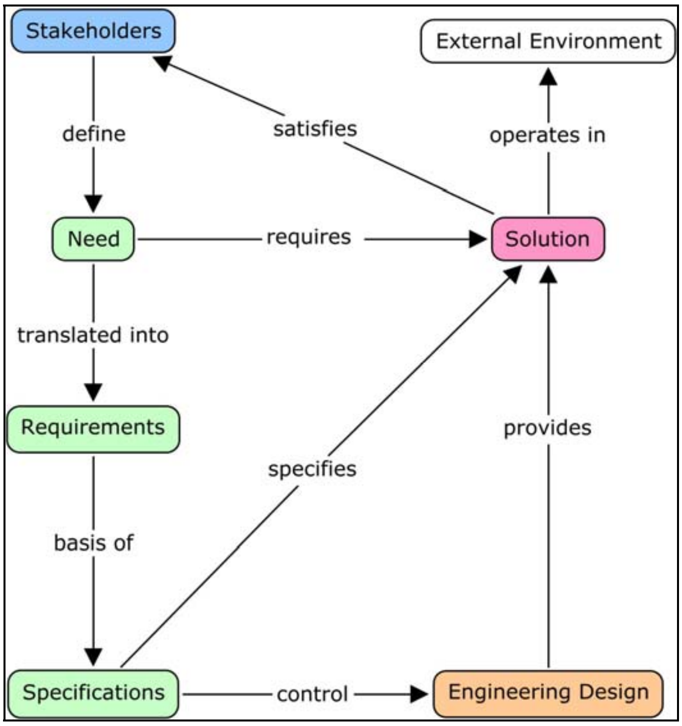

Figure 3. Simplified Model of the Systems Engineering Process

In this process the need of stakeholders are translated into a requirements statement that will form the basis of the SE process and the development of specifications.

During requirements analysis and formulation, stakeholders of a development project should be able to contribute their knowledge into an integrated data repository. A collaborative process of elimination and acceptance of requirements can then be the driving force of the requirements statement. If not, the development team may formulate requirements that are incomplete or that does not form a definition of the complete problem (Carson 2001).

Specifications are the control mechanism of the engineering and design effort which in turn provides a solution to the stated need. If the solution addresses the need statement adequately, all stakeholders will be satisfied. We identify three reasons that may prevent the engineering effort to develop a suitable solution, namely:

- the developed solution is inadequate,
- the initial need statement does not fully address the actual needs of all stakeholders, or
- the external environment in which the solution is to operate has changed.

(Childers et al. 1994) refer to an incremental SE process that provides complete interim solutions. Figure 4 shows the entry point for the needs definition (source documents) of stakeholders. A concurrent engineering process of requirements analysis, behaviour analysis, synthesis and architecture, and design verification and validation is executed for each layer to ensure a complete solution at increasing levels of detail. The development team, with input from stakeholders, reviews the solution at each layer during the SE process. When approved, the solution of a layer often becomes a baseline that goes into configuration management. The baselines at each layer serve as control mechanisms for the engineering effort of subsequent layers.

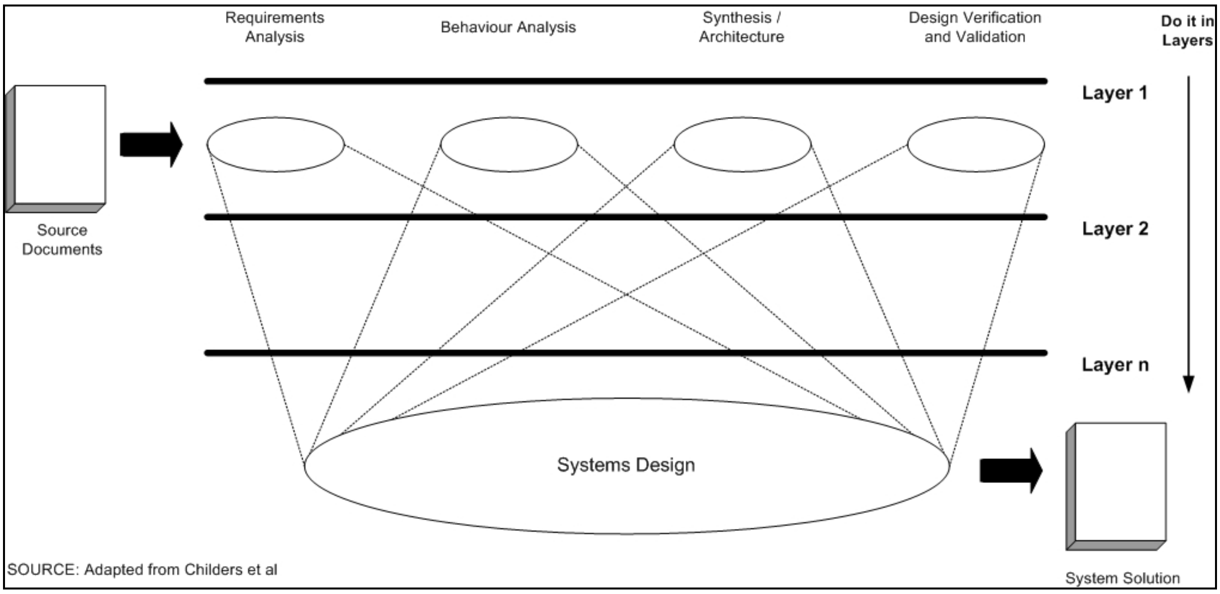

Figure 4. Incremental Systems Engineering Process

By controlling the engineering effort with baselines, the solution of a previous layer forms the basis of the requirements statement for the following layer. It often happens that, through the engineering effort, no solution can be found at a subsequent layer due any of the three reasons mentioned.

When one or more of the reasons prevents a feasible solution at a specific layer, the requirements statement of that layer may need adjustment. Changes to this requirements statement may be in contradiction with existing baselines at previous layers. Through good requirements traceability practice, the impact of changes can be traced through previous layers right up to the initial requirements statement, or to changes in the external environment.

If changes are necessary, the development team has to inform all relevant stakeholders, since each stakeholder has an interest in the behavior and performance of the solution. The knowledge of stakeholders is a very important input when revisiting requirements. They will have to contribute to and approve necessary updates to ensure a good solution. It can be very time consuming and expensive for all stakeholders to reach agreement on required changes, especially when the reflective thought processes of each stakeholder happens in isolation.

This paper proposes enhancements to the current SE process, with the support of new technology, to allow continuous contribution from all stakeholders during the engineering effort. The proposed model will allow all stakeholders to contribute their knowledge and coordinate the effort to reach agreement on necessary changes with little delay.

Collaborative Methodology

Coordination of the engineering effort is necessary to ensure the needs of all stakeholders are considered. (Gonçalves 2008) states that coordination of effort is based on the agreement between stakeholders of a project. Agreement is shared knowledge that follows from mutual adjustment. He continues to say that mutual adjustment is a coordination mechanism of communication between all stakeholders. This allows for feedback on the respective thought processes of all stakeholders and ensures that knowledge of each stakeholder is accessed in order to build agreement on how to proceed.

The interaction between the development team and the stakeholders is currently document driven. We propose a model, as shown in Figure 5, which allows for dynamic interaction between stakeholders and the development team. We will explore how existing information technology can support this dynamic and collaborative model.

In the proposed model an integrated data repository contains all the information relevant to the engineering effort. A history of changes made to the data repository should be kept throughout the SE process. The software development community refers to such a history of changes as version control. This will enable the systems engineer to generate a list of changes at any point during the development process, showing the time of the change and by whom the change was introduced. By sharing the change history with stakeholders, they can see how the model evolved and what the rationale was for each change. This enables fidelity between all interim solutions during the SE process and the stated need.

Mutual adjustment is necessary when a change is proposed on which stakeholders must reach agreement. An example of this is when a change in the external environment necessitates a change of the system definition and this change impacts on the interest of stakeholders. After agreement has been reached through mutual adjustment, the data repository can be updated and a new version can be created.

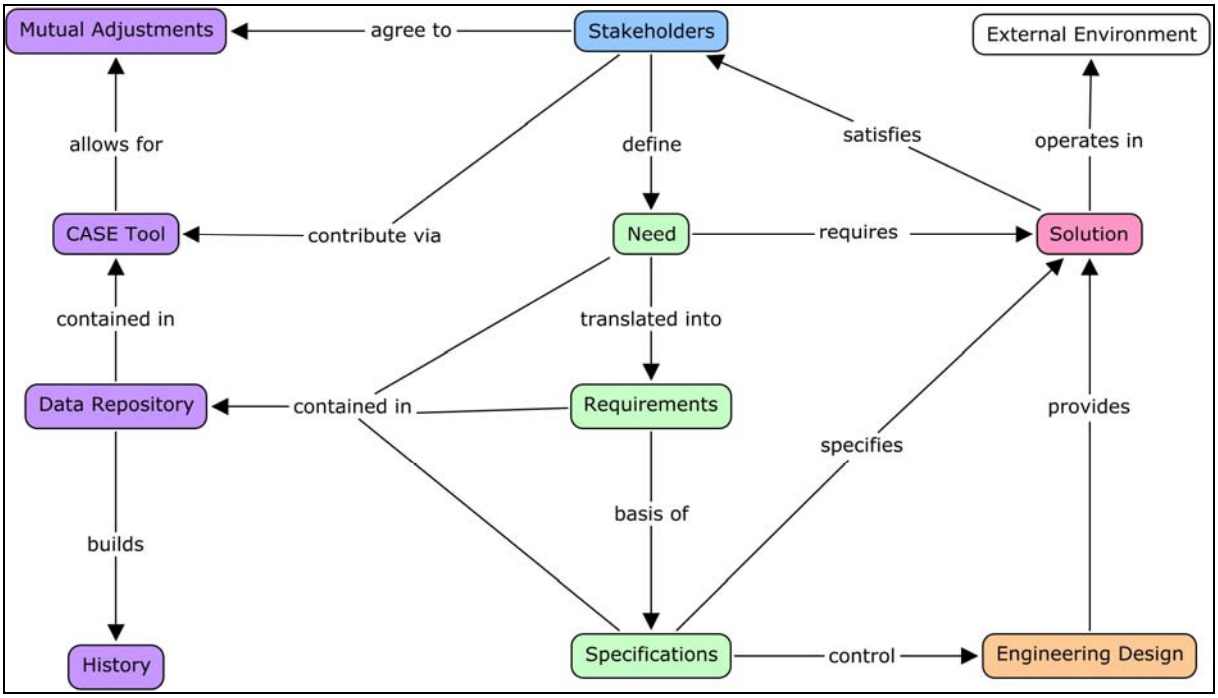

Figure 5: Proposed Model of the Systems Engineering Process

The proposed model encourages contribution from stakeholders, not only during reviews of interim solutions at each layer, but also during solution development of each layer and eventually throughout the entire SE process. The body of knowledge available to and information updated for the engineering effort are the result of a collective endeavour between the development team and other stakeholders. With support of network technologies, the advantages of stakeholder contribution and evaluation of the system definition can be summarized as follow:

- The risk of developing a system definition and ultimately a solution that does not address the specific needs of the stakeholders is reduced.
- Stakeholders can, through the reflective thought process, increase the understanding of their own needs as the engineering effort progresses through more detailed levels of abstraction.
- Relevant, accurate and timely capturing of information at all layers of the SE process will reduce the effort to reach mutual agreement at design reviews.
- The tacit knowledge stakeholders have of the external environment can be captured as soon as it becomes relevant.
- Costs and time associated with geographical dispersion of stakeholders such as travelling, distribution of printed documents, and opportunity costs associated with preparation for and attendance of design reviews can be reduced.

(Rettberg 2005) states that whenever people are collaborating on a project of knowledge sharing or creative production, they are collaborating not only with other people, but with a system which they, the other participants and the communicative environment help to create. He continues by commenting on Wikipedia, the collaborative created and maintained encyclopedia, as follows, “By making the distribution of power clear, by establishing collective responsibility, and by empowering stakeholders to not only opine, but act in the formation of the knowledge base, the [project] can manage to avoid bureaucratic bottlenecks that have plague similar endeavours in the past.” (Rettberg 2005)

The principle limitation of such a collaborative effort is that subsequent layers or levels of abstraction rely on an inclusive relationship with previous layers of the system definition.

Therefore, the success of the system definition depends on continuity and causality. This can be ensured by building rules of the prevailing systems engineering methodology into the model and by adjusting the system definition in previous layers through mutual adjustment. If necessary, the system definition can revert back to a previous version if proposed changes are not relevant.

We argue that by allowing stakeholders access to the integrated data repository, they can identify errors and anomalies and propose corrections through their contributions. When two or more elements of the system definition are in contradiction, changes and updates can occur seamlessly through mutual adjustment.

Technologies Supporting the Methodology

Various CASE tools are available for complex engineering development such as CORE, DOORS and Cradle. These tools allow for a concurrent engineering effort where the development team's contributions can be captured and evaluated in an integrated data repository. Current tools do not allow all stakeholders to effortlessly create and edit content during the SE process. The functionality of available CASE tools should be combined with a collaborative software capability. Technology should support accurate capturing of all stakeholder contributions when it becomes apparent during the SE process. This can be achieved when all stakeholders have the ability to view, create and edit content in the integrated data repository and collaboratively approve solutions at each layer of the system definition through mutual adjustment.

In our proposed model, the integrated data repository will contain all the information relevant to the engineering effort and a history of the changes made to data elements in the data repository. Therefore, the current approved baseline and the previous approved baseline with interim changes can be easily extracted. Figure 6 shows that in the concurrent engineering process, data elements can have a number of changes between two baselines. Version control for each data element should be implemented according to an approval rule, for example a data element can be updated to a new version when consensus is reached on proposed changes through mutual adjustment by all relevant stakeholders. This will prevent erroneous information from individual stakeholders to influence the feasibility of the solution. A baseline can be fixed for the system definition when a feasible solution is achieved at a specific layer.

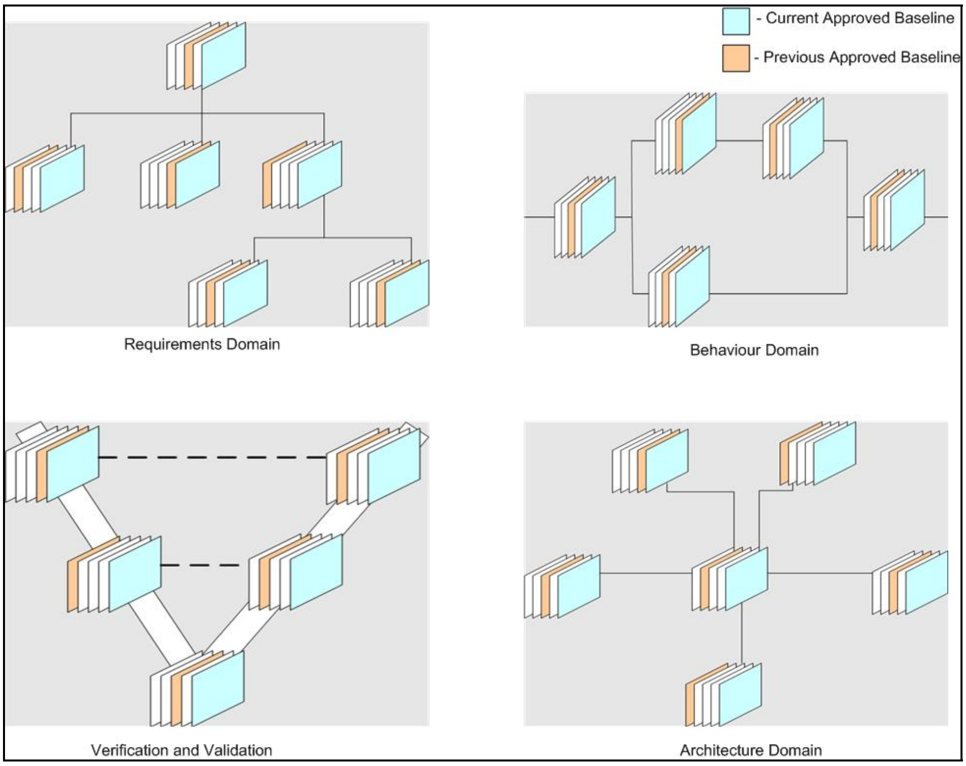

Figure 6: Three Dimensional View

Design decisions are often made during meetings and work sessions with relevant stakeholders present. All relevant contributions are necessary to augment data elements of the system definition into a feasible solution. This collaborative interaction between stakeholders can be supported with software technologies that will reduce the need for meetings and work sessions.

Wikipedia describes collaboration interactions and the technologies supporting it as follows. “In collaborative interactions the main function of the participants' relationship is to alter a collaboration entity. The collaboration entity is in a relatively unstable form. Examples include the development of an idea, the creation of a design, the achievement of a shared goal. Therefore, real collaboration technologies deliver the functionality for many participants to augment a common deliverable. Record or document management, threaded discussions, audit history, and other mechanisms designed to capture the efforts of many into a managed content environment are typical of collaboration technologies.” (Wikipedia 2009)

Google Wave is a communication protocol that supports conversations over a computer network with multiple participants. By using the Google Wave protocol stakeholders can now make all contributions from meetings and work sessions through online collaboration. Google Wave is currently in development and is intended for release towards the end of 2009.

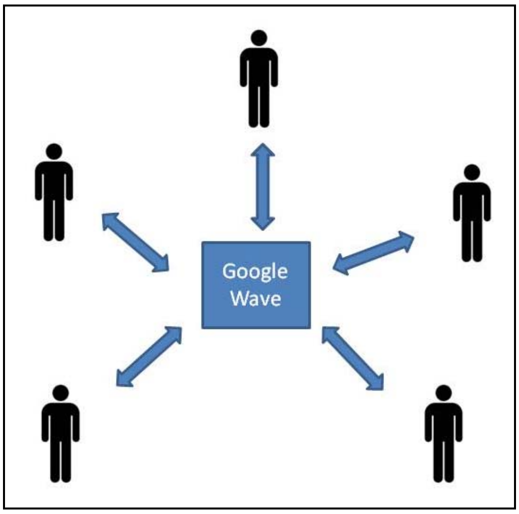

Figure 7: Google Wave Communication Protocol

A wave in the Google Wave protocol is defined as follows:

- **A "wave" is equal parts conversation and document.** Users can almost instantly communicate and work together with richly formatted text, images, and other multimedia.
- **A wave is shared.** Any participant can reply anywhere in the message, edit the content and add participants at any point in the process. Then playback lets anyone rewind the wave to see who contributed what and when.
- **A wave is live.** With live transmission as you type, participants on a wave can have faster conversations, see edits and interact with extensions in real-time (Google 2009).

By utilizing the Google Wave protocol, the proposed model can achieve its goal of dynamic interaction between the development team and stakeholders. The combination of the Google Wave protocol and systems engineering principles embedded in existing CASE tools can enable seamless contribution of all stakeholders in creating, updating and development of a system definition. A collaborative effort can be achieved through harnessing the capability of this technology in the field of systems engineering.

Conclusions and Future Work

Through collaboration the needs of all stakeholders can be considered, resulting in a more complete need statement. It also allows for dynamic changes at different stages of the system definition to ensure that a more robust, efficient and effective solution can be achieved. Collaborative contributions to the system definition should be tracked by building a history of changes. The ability to revert to previous versions of the system definition allows the systems engineer to have control and authority over the proposed changes.

As technology advances and systems become more complex, the systems engineering process should adapt accordingly to take advantage of new capabilities. With current advances in information technology and the expected increase in bandwidth for South Africa, there is an opportunity to encourage a collaborative endeavour of all stakeholders during the systems engineering process.

Upon formal release of the Google Wave protocol later in 2009, we aim to integrate necessary model based systems engineering rules into the collaborative functionality of Google Wave to test the feasibility of our proposed model. As an afterthought, this feasibility study can be done in collaboration with practicing systems engineers in South Africa and the rest of the world using Google Wave as the collaboration tool.

Acknowledgements

We would like to thank Alwyn Smit for his mentorship and his preparedness to share his vast experience in the field of systems engineering with us.

We thank Sven Woermann for eagerly sharing his knowledge about new and upcoming technologies with us.

References

Carson, Ronald, "Keeping the Focus During Requirements Analysis", INCOSE Proceedings, 2001.

Childers, S.R. and Long, J.E., "A Concurrent Methodology for the System Engineering Design Process", INCOSE Proceedings, 1994.

Dewey, John, *How We Think.* Prometheus books, 1991, first published 1910.

Gilmour, Graeme, "Let's Just Think About That for a Minute", INCOSE Proceedings, 2002.

Google, http://wave.google.com, 2009.

Gonçalves, Duarte, "Processes and Beyond", INCOSE SA Proceedings, 2008.

Rettberg, Scott, "All Together Now: Collective Knowledge, Collective Narratives, and Architectures of Participation", *DAC Conference Proceedings*, 2005. http://retts.net/.

Wikipedia, "Collaborative Software", http://en.wikipedia.org/wiki/Collaborative_software, 2009

Biography

Daniël Malherbe. In 2003 he graduated with a B.Sc degree and in 2007 with an M.Sc degree in Industrial Engineering from Stellenbosch University. The M.Sc dissertation focused on improving the competitiveness of the South African tool and die manufacturing industry by benchmarking the industry against international counterparts. In 2006 he joined a project team at the CSIR where he took the responsibility as database administrator of the Computer Aided Systems Engineering tool, CORE. He is the founder of DCM Consulting CC, a growing systems engineering and development consultancy. His current interest lies in optimizing the systems engineering process with support of emerging technologies.

Suja Joseph-Malherbe. She graduated with a B.Sc degree in Electrical Engineering from the University of the Witwatersrand in 2002. In 2003, after acquiring her degree, she joined CSIR - Defence, Peace, Safety and Security (DPSS). She has been involved in the application area of modeling and simulation, using tools such as MATLAB. She contributed, being part of the project team, in upgrading the simulation environment which provides decision support to the South African Air Force (SAAF) and implementing tracking algorithms for the purposes of visual tracking. Her interest currently lies in requirements analysis and in understanding and improving the interaction between the stakeholders and the development team.