International Journal of Cooperative Information Systems

A SITUATIONAL METHOD FOR CREATING SHARED UNDERSTANDING ON REQUIREMENTS FOR AN ENTERPRISE ARCHITECTURE

AGNES NAKAKAWA

Makerere University, P. O. Box 7062, Kampala, Uganda

PATRICK VAN BOMMEL

Radboud University Nijmegen P.O. BOX 9010, 6500 GL Nijmegen, The Netherlands

ERIK H.A. PROPER

Public Research Centre Henri Tudor L-1855 Luxembourg-Kirchberg, Luxembourg
Radboud University Nijmegen P.O. BOX 9010, 6500 GL Nijmegen, The Netherlands

HANS J.B.F. MULDER

University of Antwerp Management School, Sint-Jacobsmarkt 9-13, 2000 Antwerpen, Belgium

Received Day Month Day
Revised Day Month Day

Abstract. Lack of shared understanding among stakeholders is a commonly cited drawback in enterprise architecture development. Stakeholders need to have shared understanding of requirements and principles for an enterprise architecture, and the extent to which the resultant architecture addresses their concerns. However, existing approaches for enterprise architecture development lack adequate capabilities for managing aspects associated with creating shared understanding among stakeholders. Although such aspects can be largely managed by approaches for collaborative decision making and soft systems thinking, these approaches lack details on the enterprise architecture process and its products. Therefore, this paper explores ways of mutually diminishing these gaps through adopting situational method engineering, to guide the development of a situational method for enabling stakeholders to acquire shared understanding of requirements for an enterprise architecture. The situational method presented herein is a component of a broader method for supporting collaboration between stakeholders and architects during enterprise architecture creation. Although the latter was successfully evaluated in 6 enterprises, it exhibited highest performance scores in two enterprises after it was amended with the situational method. Therefore, this paper also presents key findings from evaluating the situational method in the two agencies that are located in Uganda.

Keywords: Requirements for Enterprise Architecture; Shared Understanding; Situational Method Engineering.

1. Introduction

The several benefits or functions of an enterprise architecture demand for a comprehensive collaborative engagement with key stakeholders during the creation of an enterprise architecture.[31, 41] This is because stakeholder engagement yields an enterprise architecture that accommodates concerns and requirements of stakeholders on the baseline and target contexts of the enterprise.[58] There are several enterprise architecture development approaches that specify the portfolio of products expected from an enterprise architecture effort, and/or the procedure and tools required to create those products.[8,58] However, architecture approaches lack adequate capabilities for managing aspects associated with creating shared understanding among actors. Such capabilities are available in existing approaches for collaborative decision making and soft systems thinking among others (see section 3), but these approaches lack specifics of the architecture process and products. Thus, challenges associated with the lack of shared understanding prevail in the architecture creation process[37] as depicted in Fig. 1.

Based on Nakakawa et al.[37], the bottom part of Fig. 1 shows challenges of collaborating with stakeholders during enterprise architecture creation, and arrows showing inter-linkages among these challenges are numbered 1 to 11. The top part of Fig. 1 shows specific challenges associated with creating shared understanding among stakeholders during enterprise architecture creation. These issues are plotted on the thick dashed vertical lines and are numbered using (a) to (f) for reference purposes in subsequent sections of this paper. The double arrows on the thick dashed lines indicate the intertwined nature of issues (a) to (f) and two collaboration challenges (i.e. ineffective communication and lack of shared understanding and shared vision).

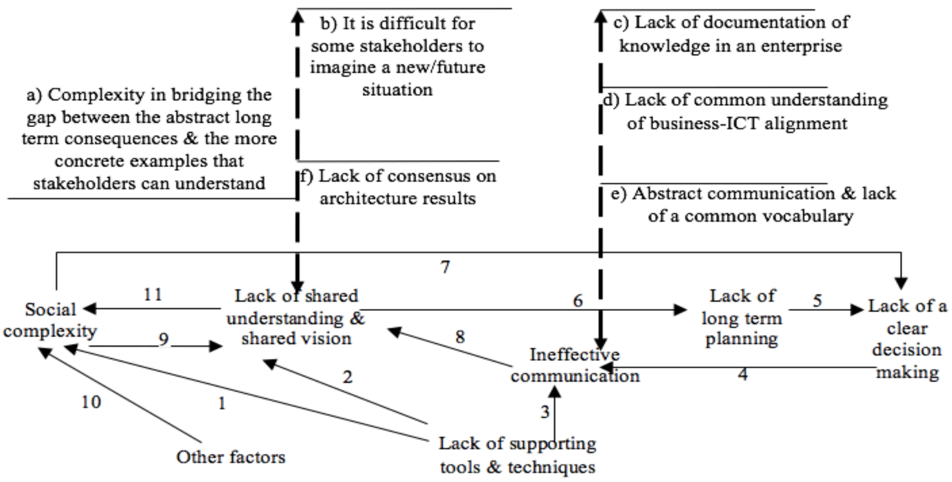


Fig. 1. Challenges of involving stakeholders in enterprise architecture (Based on Nakakawa et al.[37])

The challenges (and their inter-linkages numbered 1 to 11) at the bottom of Fig. 1 were comprehensively presented in earlier work.[37] Thus, this paper focuses on presenting a deeper outlook on how to address challenges plotted on the dashed arrows numbered (a) to (f) at the top of Fig. 1. Issues in Fig. 1 not only reveal the multidisciplinary and systemic nature of the architecture creation process, but also a critical need to devise techniques for enabling effective communication and building of shared understanding among stakeholders during architecture creation. Earlier attempts to address this need involved developing a process that supports four aspects of collaborative decision making in architecture creation, as shown in the left side of Fig. 2.[37, 38] However, its field evaluation indicated scores that were below our target performance on support for aspects associated with creating shared understanding (section 3.1 elaborates this).

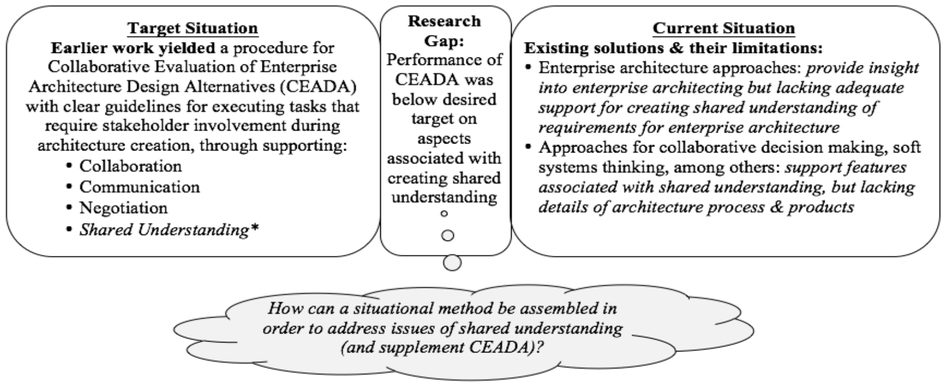


Fig. 2: Research Gap and Motivation

The right of Fig. 2 indicates that adopting and synthesizing techniques from existing approaches can addresses specific issues associated with shared understanding and thereby mutually diminishes limitations of these approaches with respect to addressing the research gap. *Situational Method Engineering* enables this through supporting the building of project or enterprise specific (situational) methods by assembling components or fragments of existing methods,[7] and reusing best practices, theories and tools.[50] Thus, it is adopted herein to develop a situational method as an orchestration of techniques to address the research gap. As indicated at the bottom of Fig. 2, this paper attempts to answer the question: *how can a situational method be assembled in order to mutually diminish limitations of existing approaches so as to address issues associated with creating shared understanding of requirements for an enterprise architecture, and what should constitute such a method?* To answer this, the remainder of this paper is structured as follows. Section 2 presents the research approach. Section 3 discusses the concept of shared understanding in enterprise architecture creation, related work, gap analysis, and possible solutions. It also discusses situational method engineering and how it was adopted in this research. Section 4 discusses the design of the situational method for creating Shared Understanding of Requirements of Enterprise Architecture during its Creation (SUREAC). Section 5 discusses the evaluation of SUREAC, and section 6 concludes the paper.

2. Research Approach

This research was guided by Design Science and Action Research. Design Science is a prescriptive and utility-oriented approach that involves using existing knowledge to devise innovative artifacts that improve system performance or address enterprise and societal problems.[33, 29, 26, 27] On the other hand, Action Research is a social-oriented approach that enables researchers to observe and closely interact with research subjects, so as to gain deeper understanding into the complexity of enterprise and societal problems.[4] The joint adoption of these two research approaches was motivated by following two reasons.

- First, Design Science sources[27, 26, 69, 68] articulate that at the evaluation stage of a Design Science Research effort, other approaches (such as simulations, experiments, case study, survey, field study, and action research) can be invoked and used to provide insight into rigorous evaluation of a designed artifact.
- Second, from the above list of candidate approaches for evaluating Design Science artifacts, Action Research was considered most relevant in the evaluation stage of this research. This is because: (a) Action Research informs the delivery of artifacts that attempt to address socio-technical and human related problems,[68] and can guide naturalistic evaluation of the usability of Design Science artifacts;[68, 69, 61] and (b) Design Science and Action Research are not *mutually exclusive* because the latter can guide the evaluation of artifacts designed using the former.[29]

Fig. 3 shows an instantiation of the complementary adoption of Design Science and Action Research (based on Hevner et al.[26, 27]) and the resultant four cycles in this research.

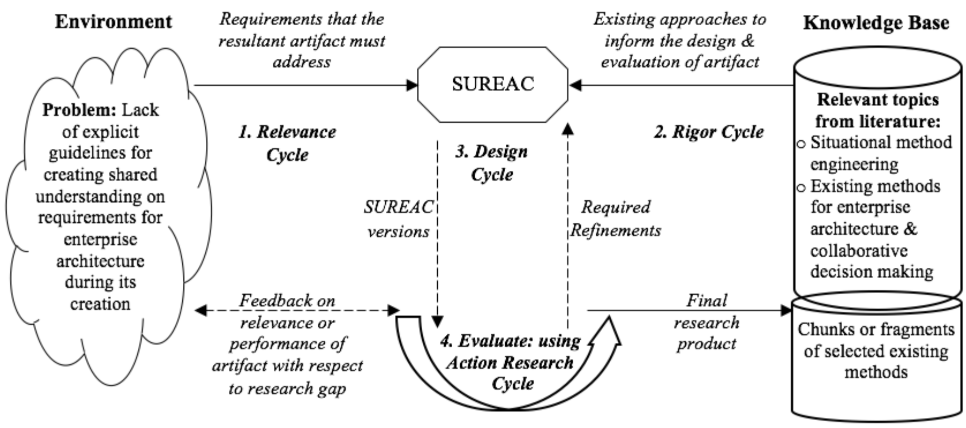


Fig. 3. Joint Adoption of Design Science and Action Research in Developing SUREAC

The *relevance cycle* indicates that requirements to be addressed by a research artifact are derived from the research problem, and resultant versions of an evolving artifact must be subjected to the problem environment so as to gain feedback on artifact relevance and performance.[26, 27] Thus, the left of Fig. 3 shows the research problem (which is further discussed in section 3.1), while requirements that the artifact must address are presented in section 4.

The *rigor cycle* highlights the need to continuously and skillfully adopt or use existing knowledge when building the artifact.[26, 27] Thus, the right of Fig. 3 shows the core paradigm adopted herein, while other approaches adopted are discussed in sections 3 and 4.

The *design cycle* indicates that the design of the artifact is an iterative process that involves building the artifact and refining it until it fulfils its purpose.[26, 27] Thus, the center of Fig. 3 shows SUREAC as the resultant artifact, while its design details are presented in section 4.

The *evaluation cycle* in Fig. 3 represents the actual instance when Action Research is invoked to evaluate SUREAC (based on the justification that precedes Fig. 3). Sections 3 and 4 further highlight how this research subscribes to guidelines of Design Science, while section 5 presents details on how Action Research was conducted.

3. Shared Understanding in Enterprise Architecture Creation

Design Science guides the development of technology-oriented innovations that can solve relevant organizational and societal problems.[26] Therefore, prior to discussing the technology orientation nature of research artifact in section 4, this section argues the relevance of the problem that motivates this research.

Stakeholder involvement is a key pillar in enterprise architecture creation and implementation.[58, 57, 47, 41, 31] However, it results in problematic issues that are silently fueled by ineffective communication and lack of shared understanding among actors, as indicated in Fig. 1. *Shared understanding* is a state when two or more actors ably explain a given situation in a similar way, predict similar consequences and outcomes of the situation, and specify similar justifications for specific courses of action needed to address the situation.[62] Also, shared understanding is a dynamic state where several actors in a group attain *mutual* knowledge on content of facts and structuring of facts, norms, and views about a situation.[63] While all individuals can not attain the same breadth and depth of understanding a given context, there is a level of reasonable understanding that they can mutually attain on specific aspects of a circumstance.[62]

Basing on the above definitions, shared understanding in enterprise architecture creation is a state when s*takeholders involved in the initiative reach a level of comprehension of the requirements for the enterprise architecture, that enables them to ably and similarly justify the extent to which each of their major concerns/needs are directly and indirectly accommodated in the requirements and by the resultant architecture*. However, this does not imply that all stakeholders understand the requirements and corresponding architecture in the same way in terms of breadth and depth. Instead, it implies that although some of their specific concerns/needs may not have been satisfied, the stakeholders are aware and can ably and similarly explain the reasons why and the tradeoffs made to accommodate major concerns/needs of key stakeholders. Creating shared understanding in heterogeneous groups helps them to gain efficiency and increased productivity in their operations.[63] To attain shared understanding, issues associated with communication and social complexity need to also be addressed (as indicated in Fig. 1 – lines numbered 8 and 9). Thus, section 3.1 highlights the research gap, and section 3.2 discusses potential intervention.

3.1. *Related Work, Gap Analysis, and Potential Solutions*

Related work herein is represented using three broad categories as discussed below.

First category comprises *enterprise architecture frameworks,* which delineate specifics of the process and products of an architecture effort. Buckl and Schweda[8] and Schekkerman[55] comprehensively assess the strengths and weaknesses of these frameworks. Enterprise architecture frameworks have shaped the architecture practice by articulating aspects that underlie it as a holistic approach for designing and guiding the implementation of enterprise transformations. A typical design approach articulates four dimensions that inform its mode of use, i.e.: *way of thinking* – specifies the philosophical notions on which the approach is premised; *way of modelling* – specifies types of models obtained using the approach; *way of organizing* – specifies procedures involved in the approach, their alignment and management or quality control aspects of the approach; *way of supporting* – specifies techniques and tools associated with the approach.[73] Thus, in the architecture practice, enterprise architecture frameworks specify: way of thinking (such as architecture domains to consider), way of modelling (such as modelling approaches to use and types of resultant models), way of organizing (such as procedures or guidelines to follow and architecture governance issues to consider), and way of supporting (such as recommended techniques and tools). However, two critical aspects are not covered comprehensively by the architecture frameworks, i.e.: (a) mode of stakeholder engagement, which falls under the way of organizing; and (b) mode of documentation, which lies at the intersection of the way of thinking and the way of modelling. Since aspects on stakeholder engagement are explored in our earlier work,[37, 38] this paper attempts to explore aspects on documentation of deliberations towards creating shared understanding of requirements for an enterprise architecture.

The second category of related work comprises *architecture modeling approaches,* which provide syntax and semantics used to derive architecture views that communicate stakeholder requirements. Instances in this category include: ArchiMate,[31] and Design and Engineering Methodology for Organizations – DEMO.[13] Architecture modeling approaches inform the way of modelling and way of supporting in architecture development by providing mechanisms for standard representation and interpretation of content presented by architecture blueprints that specify a desired or target transformation. However, they are silent about mechanisms of eliciting, structuring, and documenting stakeholder insights on various contextual aspects that inform the formulation of requirements and constraints for the enterprise architecture.

For example, ArchiMate is an architecture description language that addresses various aspects of documenting enterprise architecture domains in a standardized way in order to create shared understanding thereof.[31] ArchiMate was extended using goal oriented requirements engineering to model business goals and stakeholder concerns so as to enable traceability of business goals in the enterprise architecture.[46,20] Accordingly, ArchiMate also supports representation of the following: (a) motivation elements of an enterprise transformation such as strategic drivers, contextual assessment aspects, goals, stakeholders, constraints, requirements, and principles; (b) and strategy implementation and migration elements such as course of action, capability, resources.[74] Yet there is still need to supplement these efforts with support for eliciting and documenting stakeholder insights that underlie the specific motivation and strategy implementation elements. This is because in some contexts, the stage of elicitation and documentation of information on baseline and target aspects of an enterprise is not trivial. Thus, the research gap at this stage is a lack of *"worksheets" or "templates"* for guiding reasoning and deliberations which yield insights that serve as prerequisites for the formal architecture descriptions that are supported by architecture description languages. Moreover, as indicated in the challenges plotted at the top of Fig. 1, some stakeholders hardly envision concepts in abstract form. This underlines the need to establish worksheets or templates that help stakeholders to understand the required reasoning pattern when specifying aspects that constitute the motivation and strategy implementation elements or requirements for an enterprise architecture. Section 4 elaborates how such worksheets supplement the motivation and strategy implementation elements in ArchiMate.

The third category of related work comprises *supporting approaches* (in form of guidelines, techniques, tools, or methods) that are developed to fix various gaps in the above two categories (i.e. architecture frameworks and architecture modelling approaches). The research presented herein also falls under this category. Existing work that is closely related to the research herein includes the following:

- Guidelines for formulating architecture principles in a variety of settings.[23, 42] These give detailed insights into systematic structuring of content on architecture principles towards increasing their understandability and adoption or actualization. Architecture principles inform the formulation of constraints that underlie requirements for an enterprise architecture.[58] Since this research aims at providing support for eliciting and specifying constraints for the architecture, existing work on formulating architecture principles is considered foundational because stakeholders deduce constraints from principles. However, the formulation of principles is beyond the research scope.
- Guidelines for communicating with stakeholders during architecture development.[45] These guidelines articulate key aspects in architecture development conversations and provide insight into modes of communication that can be used in such conversations so as to increase shared understanding among stakeholders. These guidelines have been adopted in our earlier work,[38, 70] to define the procedure of how templates or worksheets proposed herein can be used collaboratively with stakeholders.
- Business scenarios – a technique recommended by TOGAF to support development of business requirements for an enterprise.[58] This technique highlights specific operational aspects of an enterprise that are relevant in formulating business requirements, and guiding questions that prompt stakeholders to articulate required aspects. Since this technique is silent on detailed means for structuring stakeholder insights on operational aspects in a business environment, it is adopted herein along with other approaches so as to address the research gap. Section 4.2 elaborates this.

- Use of situational method engineering to create a knowledge base of building blocks (of enterprise architecture approaches) that architects can use to derive a customized architecture process or method for a specific enterprise.[10] The research focus was addressing enterprise-specific concerns in the entire architecture development process by providing a flexible mechanism. However, the requirements formulation stage in architecture development is affected by specific challenges that need extra attention, such as those plotted at the top of Fig. 1 and those highlighted above under the second and first category of related work. Such challenges motivate the research herein.

Summary of Gap Analysis. Efforts above provide insight into facts, processes, and expected outputs in formulating requirements for enterprise architectures. However, they hardly provide detailed guidance on how the facts can be elicited and documented or structured in order to enable stakeholders to appreciate the required reasoning pattern, *origin* of facts, and *justification* of the breadth (i.e. scope) and depth (i.e. specification) of the requirements for an enterprise architecture. In addition, managing the complexity that is associated with the breadth and depth of some enterprise contexts requires *soft systems thinking*, in order to enable formulation of coherent and consistent requirements for an enterprise architecture. Soft systems thinking is the use of insightful exploratory models that accommodate political and social world views on human-related situations in order to manage complexity, rather than attempting to solve such situations by thinking in terms of systems that need to be developed or maintained.[64] Therefore, there was need to explore approaches that support soft systems thinking and other related techniques as potential solutions for enriching the requirements formulation process. A taxonomy of these is provided in Fig. 4.

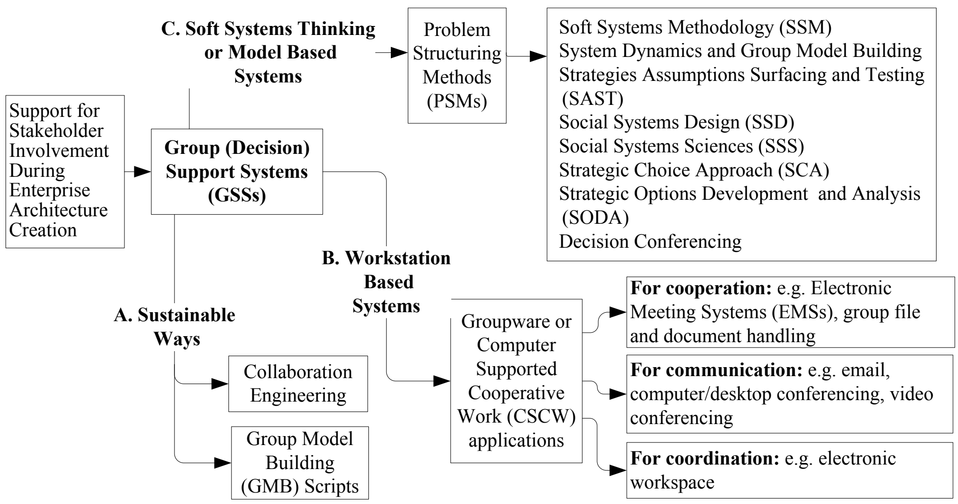


Fig. 4. Taxonomy of Existing and Potential Solutions[70]

Potential Solutions. The right of Fig. 4 shows that support for stakeholder involvement in architecture creation can be categorized into three main mechanisms labelled as A to C. It was informed by other taxonomies.[52, 17, 34, 19, 24, 15, 3, 43, 18, 44] Mechanisms A (Collaboration Engineering – CE) and B (Workstation Based Systems) were explored in our earlier efforts.[38] Collaboration Engineering (CE) was adopted to: (1) define activities involved in collaborative decision making during architecture creation; and (2) design a collaboration process comprising a set of thinking and working patterns or *thinkLets* that can be used to engage stakeholders during architecture creation. ThinkLets are pre-configured thinking and working patterns that groups undertaking critical tasks can use without hiring a professional facilitator.[5, 6] After evaluating the process that resulted from adopting CE and a Workstation Based System, findings indicated that the process performance could improve if support for shared understanding could be enhanced (see section 5 for details).

This motivated the need to also delve into soft systems thinking approaches, indicated as mechanism C in the top right corner of Fig. 4. Soft systems thinking approaches support rational and systemic deliberation on fuzzy or ill-structured and entangled enterprise problems.[12] In exploring soft systems thinking approaches, Soft Systems Methodology (SSM) was first used as shown at the bottom of Fig. 5.

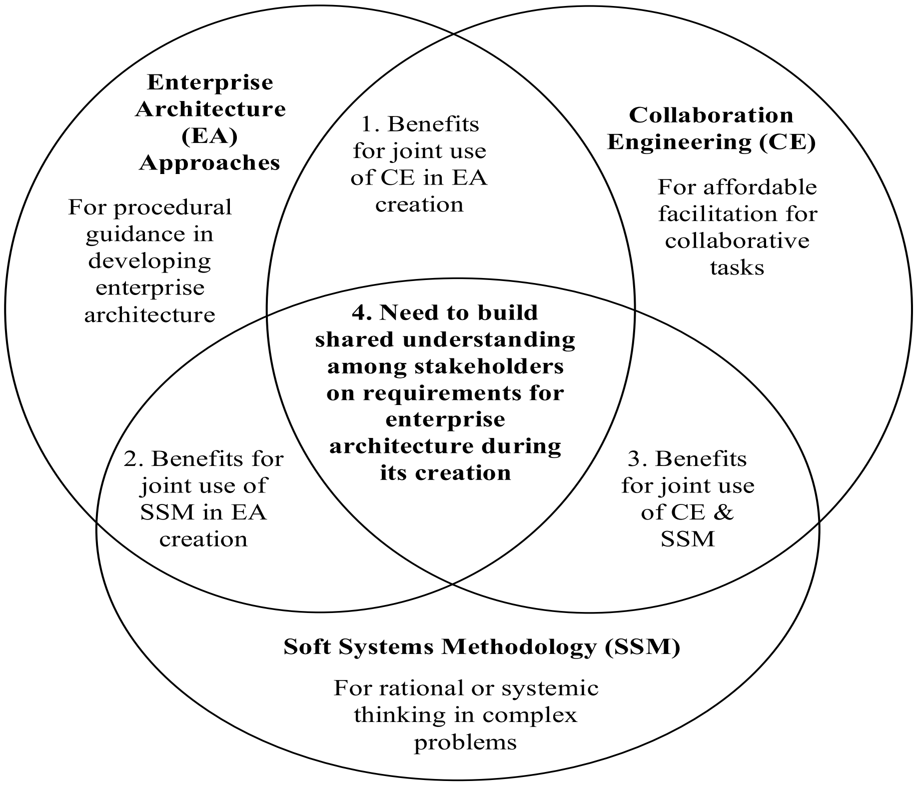


Fig. 5: Joint Adoption of Enterprise Architecture Development Approaches, CE, and SSM[70]

SSM is a sense making and context-learning approach to information systems development projects, that enables people in a problematic situation to explore and create the meaning of their context and then purposefully act with respect to that perceived meaning.[64] As indicated in the intersection parts of Fig. 5, the supplementary use of the three approaches mutually diminishes their specific limitations in the following three ways:

1. CE offers facilitation support during collaborative elicitation and analysis of requirements for Enterprise Architecture (EA), but inadequate support for visualization and structuring of facts to create a shared understanding and vision. Hence the use of SSM.
2. SSM supports systemic and rational thinking when investigating an enterprise problem situation and when assessing possible purposeful actions that the enterprise can undertake, which are key inputs in an architecture effort. Hence the joint use of SSM and EA development approaches.
3. CE offers facilitation guidance for SSM, while SSM supports visualizations and ways of structuring facts that trigger debate during tasks where actors need to process and analyze brainstormed ideas. Hence the joint use of CE and SSM.

The combo or joint adoption of approaches in Fig. 5 yielded a set of visual templates that were jointly used with thinkLets to enable execution of collaborative tasks during architecture creation.[37] This joint adoption approach was further evaluated (see section 5 for details). Although evaluation scores improved, the resultant qualitative feedback (see appendix B) and subsequent reflections indicated the need to improve the approach by addressing issues summarized in appendix B and issues associated with shared understanding that are plotted at the top of Fig. 1 (in section 1). However, addressing the latter implied the need to explore additional techniques/methods/theories that support structuring of facts with respect to paradigms of organizational structures and information management (e.g. stakeholder concerns and views, information needs, logical alignment of business needs and technology needs, performance indicators, among others). Such techniques would address the gap shown in the center of Fig. 5 and enable the creation of shared understanding on requirements for enterprise architecture during its creation. Consequently, the search for a coherent way of incorporating additional techniques into the justified combo in Fig. 5 motivated the swerve into Situational Method Engineering.

3.2. *Situational Method Engineering*

Method Engineering involves constructing and adapting methods, techniques and tools for developing information systems.[7] This paradigm was motivated by the *method-to-context* mapping dilemma, which occurred when different information systems engineering projects or enterprise situations were frequently encountered with peculiarities that could not be addressed by already existing rigid or standardized method descriptions.[25, 7] The research problem herein is an instance of such a dilemma. This is because the approach or mode of enterprise architecture creation depends on situational factors such as enterprise problem scope, purpose of the architecture effort, stakeholders involved and their concerns, architecture maturity level of an enterprise, and availability of required resources.[58] Yet, as discussed in section 3.1, there is no individual approach that can satisfactorily address all dimensions of problematic aspects or situational factors associated with architecture creation or with creating shared understanding (among stakeholders) on requirements for an enterprise architecture. Thus, architecture development can neither be a linear approach nor be supported by a one-size-fits-all approach.[41, 59]

Addressing a method-to-context mapping dilemma requires one to adapt existing methods or reuse components/fragments of existing methods to formally construct or assemble *situation-specific* methods, an approach referred to as *situational method engineering*.[50, 51, 48, 25] However, the success of situational method engineering depends on existence of several pre-existing method fragments which are assembled to form a new method.[28] These fragments are coherent parts of systems development or engineering methods that are identified to partially or fully address particular situational requirements.[14, 25, 7] Since Fig. 4 and preceding arguments echo that some issues associated with creating shared understanding during the formulation of architecture requirements can be addressed by jointly using existing approaches, situational method engineering is adopted herein. Section 4 discusses how this was done.

4. Development of SUREAC

In Design Science research, the design of an artifact is informed by scientifically sound foundational theories and methods or approaches.[26] Although there are several methods and techniques that are adopted to design the artifact presented in this section, the overarching approach is situational method engineering as justified in section 3.2. Therefore, this section discusses how the adoption of situational method engineering guides the systematic adoption of other approaches that are relevant to inform the design of the artifact.

Situational method engineering involves two key steps, i.e.: (A) defining the goal of a situational method engineering engagement, and (B) constructing a method to achieve the set goal.[50, 51, 48, 7] Steps A and B are shown in the center of Fig. 6, and the rest of Fig. 6 shows how these steps were adopted.

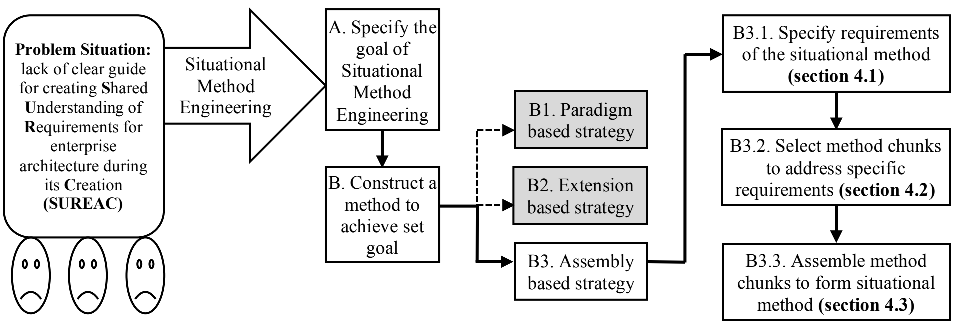


Fig. 6: Instantiation of Situational Method Engineering in Enterprise Architecture Creation

Under step A, as indicated on the left of Fig. 6, the goal of situational method engineering was to develop SUREAC. Under step B, Ralyte et al.[51] suggest three techniques of method construction, i.e.: (B1) Paradigm based strategy – abstracting an existing model or instantiating a meta-model; (B2) Extension based strategy – extending an existing method; (B3) Assembly based strategy – extracting method components from existing methods and reusing them to address specific situational requirements. Strategies B1 and B2 are shaded in grey and have connecting dashed lines to indicate that they were used in earlier efforts of this research, but evaluation results indicated a gap (see sections 3.1 and 1) that can be addressed by the mutual adoption of techniques that support specific aspects in enterprise architecture creation. Thus, herein strategy B3 was the most appropriate (as indicated by the thick arrow in center of Fig. 6).

Under B3, constructing a method using assembly based strategy involves three major tasks, i.e.: (B3.1) specifying method requirements; (B3.2) retrieving and selecting fragments or chunks of existing methods to satisfy specific requirements; and (B3.3) assembling the selected method chunks or fragments.[51, 35] A *method chunk or fragment* is a component of an already existing or standardized approach or technique or mechanism for fulfilling a given purpose or requirement, and can be treated as a basic building block in a new method so as to address organization-specific or project-specific or context-specific problems.[71, 72, 51] Examples of method chunks are discussed in subsequent sections. As shown on the right of Fig. 6, requirements for SUREAC are discussed in section 4.1, section 4.2 discusses method chunks selected to address the requirements, and section 4.3 presents the assembly of method chunks selected to constitute SUREAC.

4.1. *Specify Requirements for SUREAC*

Method requirements are elicited using either intention driven strategy – when the need is to adapt an existing method, or process driven strategy – when the need is to construct a new method.[51] Earlier efforts in this research explored the intention driven strategy by adapting existing approaches as specified in Fig. 2 (section 1) and Fig. 5 (section 3.1). Although success was attained, a need/gap was identified as indicated in sections 1 and 3.1. Therefore, this paper addresses the gap by adopting the process driven strategy to develop SUREAC as a new situational method that serves as a plug-in solution to earlier work in Nakakawa et al.[37, 38]

Process driven strategy involves eliciting requirements through defining principle activities of the required method.[48] To achieve this, the following three steps were undertaken:

1) Defining possible ways (such as activities and techniques) required to address issues associated with creating shared understanding, specifically those plotted at the top of Fig. 1 in section 1. The required sub activities and techniques have been presented in column 3 of table 1.
2) Extracting activities that are critical when formulating requirements for an enterprise architecture, from an earlier developed model of activities involved in collaborative decision making during enterprise architecture creation. Appendix A gives an overview of the earlier developed model, but the detailed sub activities that underlie the model in appendix A are presented in Nakakawa et al.[37, 38] The extraction of

activities crucial for formulating requirements for an enterprise architecture was based on the following two aspects. First, understanding requirements for the architecture implies the need to understand the following as key inputs: underlying assumptions, constraints, domain-specific principles, policies, standards, organization operational guidelines, specifications.[58] Second, stakeholders need to agree on structuring of problem facts about a situation, if they are to agree with the requirements drawn from those facts.[40] These insights were used as elaborated in step 3 below.

3) Updating the activities extracted in step (2) above using: findings from earlier field evaluations, insights from the requirements engineering spiral model in Sommerville[56], and output from step (1) above. Consequently, a model was derived (as shown in Fig. 7) that shows only activities involved in formulating requirements for an enterprise architecture. The granularity, composition, and sequencing of activities that were extracted from Nakakawa et al.[37, 38] have been modified (as specified and justified in appendix B) to form a refined set of activities that SUREAC must support. Therefore, the activities presented in Fig. 7 are pointers to the requirements that SUREAC must address.

Table 1: Possible solutions to address issues associated with creating shared understanding

#	**Shared understanding issues shown at the top of Fig. 1 (from Nakakawa et al.[37])**	**Possible ways (such as activities, techniques, and guidelines that can be adopted) to address the challenges**
a	Difficult to bridge gap between abstract long term consequences & concrete examples for stakeholders to understand the impact of specific courses of action in an enterprise[37]	1. Enable traceability-based deliberations on elements in the enterprise operational framework with respect to the short and long term consequences of requirements for the architecture *(by using the basic logic model and traceability analysis guidelines in steps 1.3, 2.4, 3.1 & 3.2 in Fig. 7).* 2. Enable formulation of detailed scenarios of desired solution strategies and enable multidimensional analysis thereof *(by using soft systems thinking and guidelines on inputs for formulating architecture requirements in steps 1.2, 2.1 to 2.4 in Fig. 7).*
b	It is difficult for some stakeholders to imagine the new or future/desired/target situation of an enterprise[37]	
c	Lack of documentation of existing practices and knowledge in an enterprise[37]	3. Formulate templates for eliciting and documenting enterprise functions, capabilities, and operational guidelines 4. Provide a checklist of required enterprise information resources for the architecture effort *(by using guidelines for developing an enterprise information repository in steps 1.1 to 2.4 in Fig. 7).*
d	Lack of common understanding of how business processes are to be aligned with ICT[37]	5. Formulate cohesion and traceability matrices or frameworks for supporting deliberations on business-ICT alignment with respect to the baseline and target enterprise operational guidelines *(by using guidelines on architecture building blocks and traceability analysis in steps 2.1 to 3.2 in Fig. 7).*
e	Stakeholders communicate in abstract terms and they lack a common vocabulary for clearly expressing relevant details in the baseline & target contexts of an enterprise[37]	6. Ensure that templates used help one to match terminologies in the enterprise operational framework with those used in developing the baseline and target enterprise architectures *for all steps.* 7. Create a mutual vocabulary of terminologies for interpreting or translating communication among stakeholders *for all steps.* 8. Develop a communication map along with stakeholders so that their communication requirements are correctly grasped *(by using guidelines on stakeholder mapping in step 1.3 in Fig. 7).*

f	Lack of consensus on architecture results and how specific stakeholder concerns have been addressed[37]	9. Build consensus gradually by seeking it at each stage *(by using the technique of single negotiating text for all steps)*.

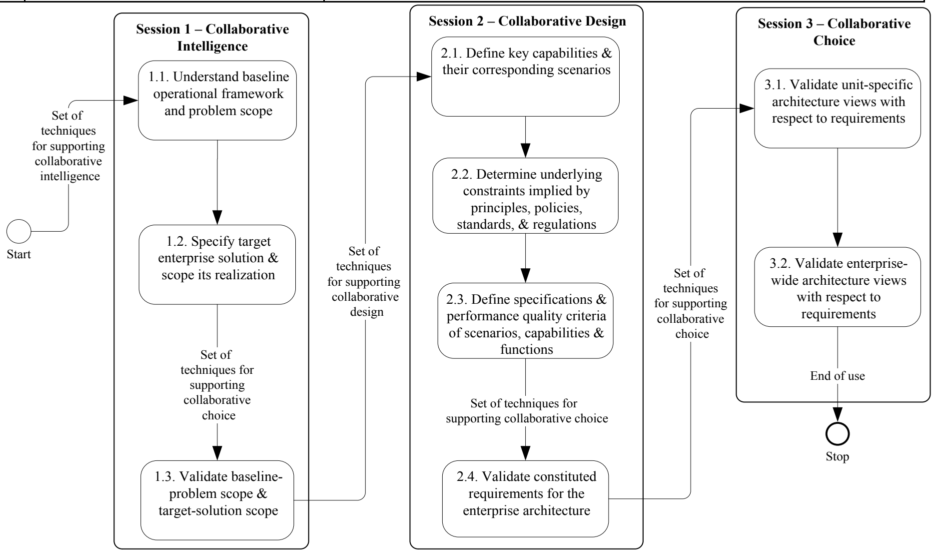


Fig. 7: Activities Involved in Formulating Requirements for an Enterprise Architecture (Concepts in the model have been extracted from activities in Nakakawa et al.[37, 38])

Table 1 and Fig. 7 are derived as described in steps (1) to (3) at the beginning of this section and represent requirements that must be addressed by SUREAC. After defining requirements for a situational method, suitable method chunks to address specific requirements are selected from a method base[51] and treated as unit building blocks for an assembly-based situational method.[49] Therefore, method chunks that were selected and adapted to address requirements in Fig. 7 are discussed in section 4.2, and are assembled to constitute SUREAC as discussed in section 4.3.

4.2. *Select and Adapt Method Chunks to address Requirements for SUREAC*

In the context of Design Science research, a feasible design of an artifact is attained through: exploring possible ways of addressing issues and laws that constitute a problem situation; and exposing the artifact to specialists in technology and management in order to accommodate insights from their experience.[26] The latter is discussed in section 5 while the former is discussed in this section. In this research, the possible ways of addressing issues in the problem domain can be perceived as method chunks that are selected and adapted as per the guidelines of situational method engineering. Accordingly, this section serves two purposes. First, it demonstrates avenues that have been explored in devising SUREAC (as a resultant artifact from this research). Second, it communicates the design of SUREAC to researchers and practitioners in enterprise architecture development. To achieve this, this section is structured as follows:

- Section 4.2.1 presents method chunks that were selected and adapted to address requirements or activities for the collaborative intelligence session (shown in Fig. 7). A summary of method chunks for collaborative intelligence session is provided in table 2, and a usage guide for the templates is provided in table 3.
- Section 4.2.2 presents method chunks that were selected and adapted to address requirements for the collaborative design and collaborative choice sessions (shown in Fig. 7). A summary of method chunks for these two sessions is provided in table 7, and a usage guide for the templates is provided in table 8.

Justifications for selecting and adapting specific method chunks have been provided at incidents in sections 4.2.1 and 4.2.2 where the purpose or application context of a chunk is introduced and discussed.

4.2.1. *Templates for Collaborative Intelligence and Justification for Selected Chunks*

Activities in the collaborative intelligence session that are presented in Fig. 7 were decomposed (as shown in column 1 of table 2), method chunks for supporting their execution were selected (as shown in column 2), and templates were derived to guide the customization or adaptation of selected method chunks into enterprise architecture creation in a coherent and traceable way. Techniques or method chunks in column 2 of table 2 were selected based on two criteria, i.e.:

a) the purpose and benefit of using a given technique or way of working. Column 2 shows the specific sources used to obtain information on purpose and benefit of each technique adopted.
b) the extent to which a technique supports the reasoning and deliberation required when executing sub activities for formulating requirements for an architecture (as specified in column 1). Column 3 shows that some techniques or method chunks were (jointly) used to design a specific template (see rows 1.2.1 and 1.2.2 in table 2).

Questions that guide analysis and deliberations when using the derived templates during the execution of sub activities for SUREAC are also presented in table 3. Thus, table 3 is a usage or questioning log that is used to guide discussions when populating templates presented in table 2. This implies that templates in table 2 are used to support the documentation of stakeholders' mutual views in response to questions specified in table 3.

Table 2: Selected method chunks and derived templates in collaborative intelligence session

Sub activities for in collaborative intelligence session (Nakakawa et al.[37])	**Selected method chunks to support reasoning and structuring of facts**	**Derived template to support documentation of facts in a consistent and traceable way**
1.1.1. Understand the enterprise operational framework to elicit inputs for requirements formulation[37]	• SSM's Rich Picture technique[12] • SSM's Analysis one-two-three[12] • TOGAF ADM's stakeholder map and power grid[58] • Management levels	• Set of symbols for Enterprise Rich Picture • Template for Stakeholder Categorization • Template for Specifying Stakeholder Roles and Constraints • Combo Template for Contextualizing Enterprise Situation

	pyramid[16] • TOGAF ADM's guidelines on inputs for describing enterprise baseline context[58] • SSM's activity model[12]	
1.1.2. Examine and scope enterprise problem[37]	• Ishikawa cause effect model[30] • Management levels pyramid[16]	• Template for problem analysis • *Template for stakeholder categorization & Template for specifying stakeholder roles & constraints (the one used in task 1.1.1)*
1.2.1. Determine most appropriate enterprise solution or venture and its corresponding implementation strategy or option[37]	• SSM's Root Definitions[12] • TOGAF ADM's on inputs of requirements formulation[58] • Traceability analysis[56]	• Template for elicitation and evaluation of solution alternatives and implementation options
1.2.2. Determine underlying internal and external constraints with respect to principles, policies and enterprise strategic goals[37]		
1.2.3. Determine purpose and scope of the architecture effort[37]	• Purposes of enterprise architecture[41] • TOGAF ADM's guidelines on scoping the architecture effort[58]	• Template for specifying purpose and scope of enterprise architecture
1.3.1. Determine key stakeholders to participate in formulating requirements for architecture[37]	• TOGAF ADM's guidelines on communication plan, stakeholder map and power grid[58]	• Template for enterprise architecture communication plan • *Templates for stakeholder categorization and Template for specifying stakeholder constraints (the one used in task 1.1.1)*
1.3.2. Validate baseline operational framework and problem scope, and the target enterprise solution or venture and its realization scope[37]	• Traceability analysis[56] • Committees and sub committees & single negotiating text of collaborative decision making[65]	• Governance based, specialization-based, and interest-task based division of stakeholders • Mutual scroll for traceability analysis and negotiation • *Template for enterprise architecture communication plan (the one used in task 1.3.1)*
1.3.3. Understand the enterprise-specific architecture creation and governance roadmap[37]		

Table 3. Questioning Log for Guiding Reasoning in Collaborative Intelligence Session

Tasks	**Underlying questions of exploration when formulating requirements for enterprise architecture**
1.1.1	What are the core enterprise business functions or capabilities, their key information exchanges, programmes and projects portfolio, products-services portfolio, key internal and external stakeholders, governing policies or principles and standards, strategic goals and drivers, key cultural-social-political values or constraints?
1.1.2	• What are the challenges or problems faced, operational concerns, possible causes and effects, possible inherent problem loops, weaknesses and threats, enterprise capabilities that are directly and indirectly affected and those likely to be affected, and internal and external stakeholders that are directly and indirectly affected and those likely to be affected? • Which problematic issues are being (or are likely to be) influenced by existing (or planned): enterprise internal principles or policies; or external regulations or policies from enterprise partnerships?
1.2.1	• What is the most appropriate enterprise solution or venture to address enterprise problem or need for change, and the enterprise strengths and opportunities for realizing it? • What is the most appropriate strategy/option for implementing the chosen enterprise solution or

	venture and the enterprise strengths and opportunities for adopting that strategy/option?
1.2.2	• What constraints do enterprise internal principles, policies, reports, plans, and strategic goals, external regulations, and policies from the enterprise partnerships impose on the chosen enterprise solution or venture? • Which principles or policies should be developed in order to properly achieve the chosen enterprise solution or venture?
1.2.3	• Given the chosen enterprise solution or venture and its implementation strategy, which architecture purposes best articulate how the enterprise would use the architecture deliverables? • Given the problem scope, the chosen enterprise solution or venture and its implementation strategy, architecture purpose, and available resources, specify: o Which enterprise functions and capabilities should be considered in architecture creation? o Which architecture domains (business, data, applications, technology, and security) should be considered and what is the appropriate level of detail to consider?
1.3.1	Given the problem scope, the chosen enterprise solution or venture and its implementation strategy, architecture purpose and scope, specify: • Which internal and external stakeholders are directly/indirectly/likely to be affected by the problem situation, what are their roles, and who will be their key decision maker? • Which internal and external stakeholders will directly/indirectly or are likely to benefit from the enterprise solution or venture, what are their roles, and who will be their key decision maker? • Which stakeholders will constitute the architecture board or steering committee that endorses deliverables from architecture creation? • What are the communication requirements for each stakeholder category?
1.3.2	• Which aspects are misrepresented, ambiguous, exaggerated, or underestimated in the baseline operational framework, problem scope, chosen enterprise solution or venture and its implementation option, underlying constraints, and scope of architecture creation?
1.3.3	• When will the roles of each stakeholder category be executed and deliverables communicated?

Details of selected method chunks (as listed in column 2 of table 2) and the derivation process of the templates in collaborative intelligence session are provided below. A justification for each selected method chunk is also provided.

Set of Symbols for Enterprise Rich Picture (Fig. 8): The lack of documentation of existing practices and knowledge in an enterprise often hinders stakeholders from having a shared conceptualization and understanding on baseline or target context of the enterprise.[37] Yet the syntax and semantics of formal baseline and target architecture views are not understandable by all categories of stakeholders. Therefore, there is need to first present baseline and target contexts of an enterprise in a format that all key stakeholder categories can understand, prior to presenting them in formal architecture views. To achieve this, the rich picture technique (as a method chunk of SSM) has been chosen. A rich picture helps to describe features of a situation in a holistic way to allow exploratory thinking thereof.[12] A rich picture can be formulated in various ways depending on the user, thus a general template for it can not be provided. However, a basic set of symbols to use in formulating it is provided in Fig. 8 to aid in holistically representing information or aspects that describe the baseline and target situations of an enterprise.

Fig. 8 extends the set of symbols presented in Nakakawa et al.[37] by adding aspects that had not been considered such as management structures or levels (especially for large, distributed, or networked enterprises), decision making boards, existing information systems among others. The set of symbols in Fig. 8 are drawn using shapes in Microsoft Office Visio. The symbols in Fig. 8 is used to represent stakeholder responses to questions listed in the usage log (see table 3, row 1.1.1), in a way that is understandable to all stakeholder categories. Examples of such responses could be on key business functions or capabilities, major information exchanges across business functions, existing programmes/projects, and key stakeholder categories or decision making committees and partners of the enterprise, major concerns or constraints in operations of a given stakeholder category.

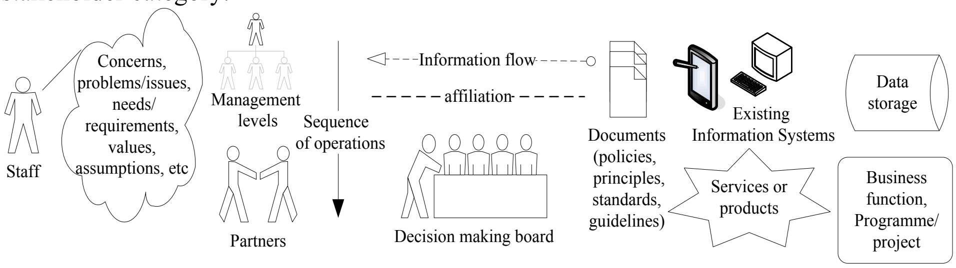


Fig. 8: Set of symbols for Enterprise Rich Picture

Template for Stakeholder Categorization (Fig. 9)**:** Identifying key stakeholders in a holistic way prevents forgetting to engage critical stakeholders in the foundational stages in enterprise architecture development. To support stakeholder identification, TOGAF ADM provides a generic stakeholder map and power grid. A stakeholder map specifies the various stakeholder categories of an enterprise, existing positions, and corresponding viewpoints; whereas a stakeholder power grid specifies the power, interest, influence levels of stakeholders.[58] The stakeholder map in TOGAF ADM comprehensively categorizes internal and external stakeholders of an enterprise, however it does not appropriately depict the entry points or levels of intervention for the different categories of external stakeholders. To do this, the management levels pyramid[16] was selected and adopted to help specify all enterprise key stakeholders. The management levels pyramid holistically and structurally depicts four general enterprise levels, three of which are managerial.[16] Besides, its structure directly implies or specifies the prioritization of stakeholders as specified by the stakeholder power grid in TOGAF ADM. The joint use of these resulted in the template shown in Fig. 9.

The template in Fig. 9 is used to represent stakeholder responses to the question of key internal and external stakeholders at all levels of the enterprise (see usage log in table 3, row 1.1.1). It shows that internal stakeholders are listed in the middle part of the template by specifying the positions of stakeholders under the respective categories represented on the left of Fig. 9. The priorities or rankings of each stakeholder, as adopted from TOGAF[58] are also indicated on the left of Fig.9. These priorities are captured because in subsequent stages of the process, trade-off analysis of stakeholder concerns and requirements has to be done. This implies that *sometimes* the concerns of a high ranking stakeholders may override those of low ranking stakeholders. External stakeholders of an enterprise (such as regulatory authorities, business partners, suppliers, corporate customers) are plotted in the block arrows that are positioned on the right part of Fig. 9. The position of the block arrows indicates priority level of each type of external stakeholder.

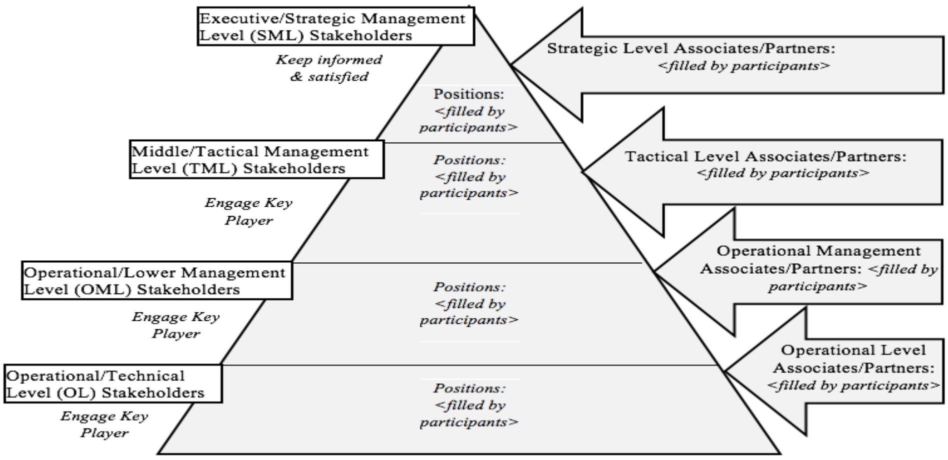


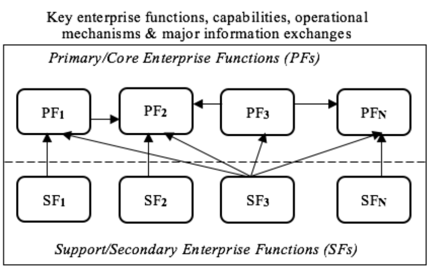
Fig. 9. Template for Stakeholder Categorization

Table 4: Template for Specifying Stakeholder Roles, Values, and Constraints

Stakeholder category	Specific stakeholder position within a category	Role of stakeholder (position and/or category)	Social, cultural, & political factors or values	Implied requirements and constraints for the enterprise architecture
Strategic Management level actors				
Strategic level partners or external actors				
Tactical Management level actors				
Tactical level partners or external actors				
Operational Management level actors				
Operational management level partners or external actors				
Operational level actors				
Operational level partners or external actors				

Template for Specifying Stakeholder Roles and Constraints (table 4): The template in Fig 9 is silent about probing for problem owners and the social, cultural, and political factors that influence their operations. Social, cultural, and political factors need to be comprehended during architecture development as they imply particular constraints that need to be accommodated in the enterprise architecture effort.[59, 58, 31] Therefore, to provide a mechanism of holistically eliciting and documenting these aspects during architecture creation, the analysis one-two-three (one of the techniques of SSM) was adopted and used to formulate the template in table 4. Analysis one-two-three supports the elicitation of social and political values of stakeholders.[12] Thus, the template in table 4 is used to represent stakeholder responses to the question on internal and external cultural, social, political, values or constraints that need to be considered in requirements formulation and validation (see usage log in table 3, row 1.1.1). Columns 1 and 2 in table 4 are filled using content from Fig. 9. Contents of columns 3 and 4 are elicited from stakeholders and offer insight into requirements and constraints that must be accommodated in the enterprise architecture. Accordingly, contents of column 5 are derived from columns 3 and 4.

Combo Template for Contextualizing Enterprise Situation (Fig. 10): TOGAF ADM specifies guidelines on relevant inputs or dimensions for describing an enterprise baseline context.[58] As indicated in the discussion of Fig. 8, prior to demonstration of baseline aspects using formal architecture views, there is need to document them in a way that is understandable by all stakeholder categories. Thus, the high level representations of enterprise aspects in Fig. 8 need to be further detailed using a format that captures and visualizes inputs or aspects of the enterprise baseline situation in an easily understandable way. To achieve this, the SSM technique of purposeful activity models was adopted and used to derive a template for contextualizing the enterprise situation (as shown in left part of Fig. 10). Purposeful activity models help to assemble transformation processes with respect to features that justify their establishment.[12]



Enterprise Function or Capability	Aim or strategic goals of capability/ function	Key inputs & source	Key outputs & destination	Titles of existing & planned governance & operational documents (policies, principles, standards)
Programme name & goal	Project name	Project goals	Project status	Programme & Project Key Performance Indicators (KPIs)

Fig. 10. Combo Template for Contextualizing Enterprise Situation

To secure stakeholder support and involvement, it is also vital that the enterprise architecture development initiative is established within the existing management and operational frameworks of an enterprise.[58] Thus, the right part of Fig. 10 shows a matrix for documenting other key aspects that constitute the enterprise baseline context (as specified by TOGAF and other architecture frameworks). Such aspects include the existing or planned strategic or management and operational frameworks (see right top part of the matrix in Fig. 10), as well as existing and planned programmes and projects that constitute them (see right lower part of the matrix in Fig. 10). The two templates in Fig. 10 are used to represent stakeholder responses to the question on core enterprise business functions, programme and project portfolio, and corresponding information exchanges that need to be considered in requirements formulation (see usage log in table 3, row 1.1.1).

Template for Problem Analysis (Fig. 11): Stakeholders need to understand the extent to which their concerns and requirements are accommodated in an enterprise architecture.[59, 58, 41] This implies the need to comprehensively and holistically explore all dimensions of the enterprise problem situation and extract a manageable set of core problems that must be directly addressed by the enterprise architecture, in order to indirectly accommodate other trivial problems. To provide guidance on how to achieve this, the Ishikawa diagram technique was adopted because its simplicity and understandability has caused it to be widely embraced by stakeholders from various disciplines. The Ishikawa diagram is a quality control mechanism that organizes factors that characterize a problem situation into mutual cause-effect relationships.[30] Thus, it was adopted herein to support elicitation, classification, and cause-effect analysis of facts about an enterprise problem situation. However, in a traditional Ishikawa diagram, similar causes are usually repeated, composite or nested causes are hard to illustrate, and causes that are somewhat minor are not adequately explored.[30] To address this gap, the Ishikawa diagram technique was adopted along with the management levels concept to derive a template (presented in Fig. 11) that can be used to identify, classify, and reason about enterprise challenges, their causes, and their inter-linkages with respect to the management levels.

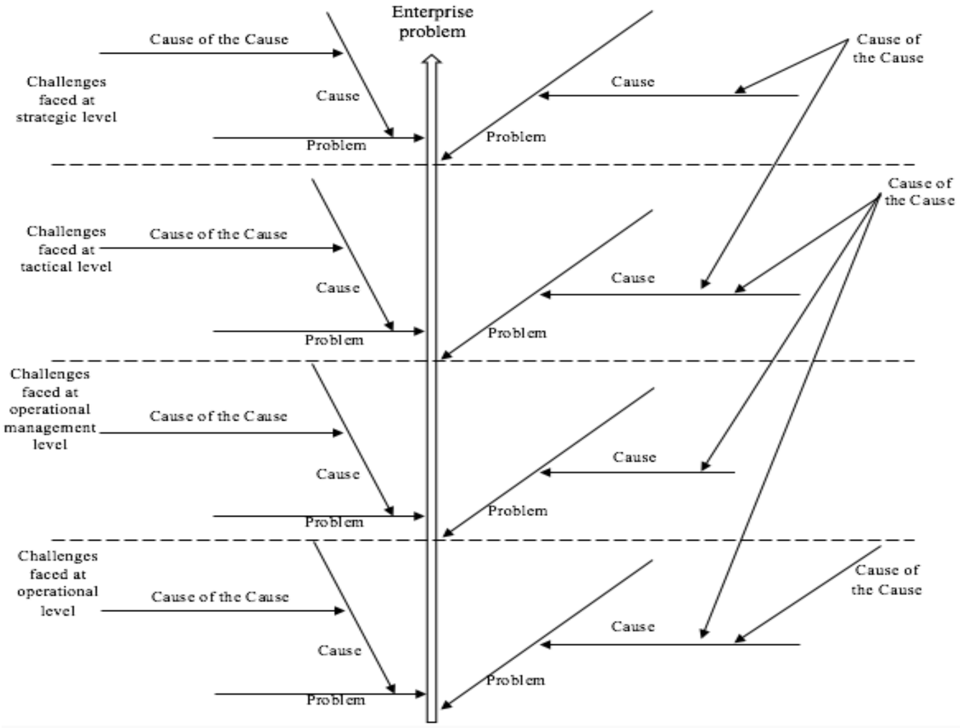


Fig. 11. Template for Problem Analysis

The template in Fig. 11 helps to determine the root causes that should be addressed by particular stakeholder categories. Thus, it is used to represent stakeholder responses to the question on challenges, concerns, causes, and effects that motivate the formulation of specific requirements (see usage log in table 3, row 1.1.2). Accordingly, the template in Fig. 11 supplement existing work on representing stakeholder concerns during architecture development. For example, in the context of ArchiMate as an architecture description language (as indicated at the start of section 3), the template in Fig. 11 can be perceived as a worksheet that prompts stakeholders to identify problems and their underlying causes such that these are represented formally and in a standardized format using ArchiMate motivation elements. Specifically, since a root cause of an enterprise problem or situation can be an internal or external directive or desired strategic intervention that was not well thought out prior to actualization, it can be represented using the *"driver"* motivation element in ArchiMate. On the left side of Fig. 11, the specific stakeholder category facing a given problem or concerned with a given problem or "driver" is indicated, and this directly translates into the *"stakeholder"* motivation element in ArchiMate. The causes and problems elicited from stakeholders using the template in Fig. 11 can be represented using the *"assessment"* element in ArchiMate.

Template for Elicitation of Solution Alternatives and Implementation Options (Fig. 12): Enterprise architecture development involves seeking stakeholder understanding on baseline and target capabilities of an enterprise, the extent to which the capabilities support achievement of business drivers and requirements, and implementation options for each capability.[58] To achieve this, there is need to first present these capabilities in a format that is understandable by stakeholders so as to elicit insights from various actors in a given transformation, prior to presenting these in formal architecture views. SSM's Root Definitions[12] and TOGAF ADM's guidelines on inputs of requirements formulation informed the design of the template for elicitation and evaluation of solution alternatives and implementation options (that is presented in Fig. 12). A Root Definition is phrase that uses a "solution strategy-to-realization option" format to suggest strategic possible courses of action in a problem situation. According to Checkland[12], a Root Definition is structured using the format: "Do X by Y in order to achieve Z", to influence reasoning towards findings answers to three vital questions in rational thinking – "*what to do* (=X)", "*how to do it* (=Y)", and "*why do it* (=Z)". The Root Definition technique was, therefore, adopted to formulate the diagram template in Fig. 12 so as to support classifying and synthesizing of brainstormed ideas on possible solutions to challenges that characterize a problem situation.

The template in Fig. 12 also helps to specify the main required course of action and possible ways of actualizing it as well as expected benefits. The WHAT component at the bottom of Fig. 12 prompts stakeholders to indicate a specific solution alternative to an enterprise problem, the HOW pillars prompt stakeholders to generate possible implementation options for a specified solution alternative, and the WHY at the top of each pillar prompts stakeholders to justify a given implementation option. This implies that Fig.

12 is instantiated for each major solution alternative to an enterprise problem.

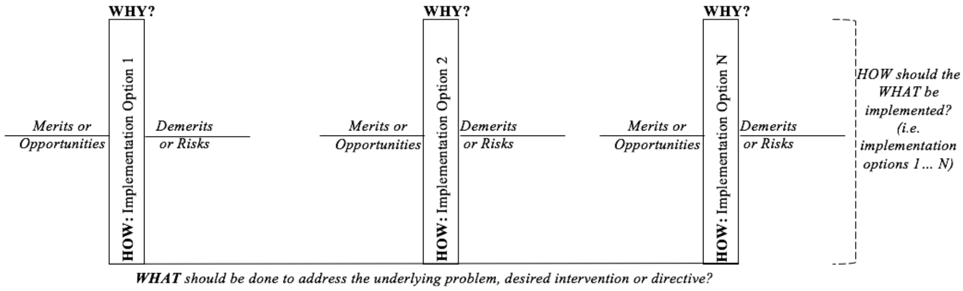


Fig. 12. Template for elicitation of solution alternatives and implementation options

Using an example in one of the agencies where SUREAC was evaluated (see details in section 5), Fig. 12 was interpreted as follows: the *main business solution alternative* was to develop an electronic enterprise-wide integrated information system that would support transaction processing at peripheral nodes affiliated with the enterprise and automatically send accurate aggregates or other statistics to regulatory authorities. *Implementation options* for realizing this were three, i.e.: (a) adopt in-house development of the desired integrated system so that an existing open source solution is customized to address enterprise-specific needs; (b) contract or procure consultancy services from service providers prequalified by the enterprise or its regulator; (c) advertise a "request for proposals or quotations or bids" to attract experts within the local and international community. These options can be represented using the 3 pillars in Fig. 12. After evaluating the merits and demerits (that should be plotted on the left and right of each pillar), option (a) was found most appropriate. Thus, Fig. 12 provides a layout for capturing information to guide the exploration and evaluation of possible courses of action to address an enterprise problem. Fig. 12 is used to represent stakeholder responses to the question on possible solution alternatives as key ingredients in requirements formulation (see usage log in table 3, row 1.2.1) and supplements existing work on representing such aspects.

For example, in the context of ArchiMate (as indicated at the start of section 3), the template in Fig. 12 serves as a worksheet that prompts stakeholders to specify aspects that can be represented in a standardized format using ArchiMate motivation and strategy implementation/migration elements. Specifically, since the WHAT in Fig. 12 represents the goal that should be achieved to address a problem, it can be represented using the into the "goal" motivation element in ArchiMate. The HOW pillars represent the possible mechanisms that can be implemented to achieve a given goal (i.e. the WHAT), and these directly translate into the "course of action" or "capability" migration elements in ArchiMate. The WHY in Fig. 12 represents the underlying motive under each possible mechanism of achieving a goal, thus it can be represented using the "driver" or "assessment" motivation elements in ArchiMate.

Template for Specifying Purpose and Scope of Enterprise Architecture Creation (Table 5): Scope management is one of the core factors in the success of technologyenabled enterprise transformations.[57] However, specifications regarding the scope or extent of coverage and detail in an enterprise architecture effort depend on the purpose of the architecture in a given transformation.[58] This implies the need for a matrix or reasoning pattern that guides stakeholders to concurrently deliberate about dimensions that constitute the scope of an enterprise architecture with respect to the purpose of (or intended benefits from) enterprise architecture.

In Op 't Land et al.,[41] benefits of enterprise architecture are classified into four main purposes of an enterprise architecture, i.e.: (A) support decision making during a desired transformation, (B) specify business requirements, (C) assess impact of a strategy prior to its implementation, and (D) inform and contract service providers. Purposes A to C echo the aspect of requirements for realizing a transformation (such as business, data, application, technology, security, human resources, financial resources) that can guide decision makers to assess the impact of a given transformation and make informed decisions. Thus, to consider the purposes of architecture during the formulation of the template in table 5, purposes A to C have been condensed into one and purpose D has been slightly amended as shown in column 2 of table 5.

Table 5. Template for Specifying Purpose and Scope of Enterprise Architecture Creation

Underlying or independent scope dimensions (i.e. resources & purpose)		Specifying dependent scope dimensions of an enterprise architecture initiative with respect to: *scope of enterprise problem (or desired change or regulatory directive), available resources for addressing it, and purpose of enterprise architecture*									
		1. Enterprise units that are affected by the problem or desired change		**2. Architecture domains that are critical in understanding/addressing problem or desired change**					**3. Level of detail required to understand/ address problem or desired change**		
#	**Resources available**	Specific units *(specify)*	All units	Business	Data	Application	Technology	Security	Vision	Intermediate	Detail
1	Timeline for addressing problem or desired change										
2	Human resource skill set										
3	Existence of enterprise information assets										
4	Financial support										
#	**Purpose of enterprise architecture**	Specific units *(specify)*	All units	Business	Data	Application	Technology	Security	Vision	Intermediate	Detail
1	Specify requirements to guide impact assessment & decision making during a transformation										
2	Contract and inform service providers in a coordinated and coherent manner										

In addition, TOGAF[58] presents four factors that are critical when scoping an enterprise architecture, i.e.: (1) the enterprise breadth or full extent in terms of functions or units and the required extent of coverage by the enterprise architecture; (2) a complete enterprise architecture description including all domains (i.e. business, data, application, technology, security) or only specific domains; (3) the level of detail required in each of the specified architecture domains (vision level or detailed level descriptions); (4) available resources in terms of enterprise information assets that can be leveraged, skill set, timelines of realizing required change, and financial resources. Dimensions 1 to 3 have been used to derive scope dimensions shown in columns 3 to 12 in table 5.

Dimension 4 – resource availability has been treated as an underlying or independent composite dimension that influences and justifies the scope specifications for the dependent dimensions 1 to 3. Other independent dimensions include the scope of the enterprise problem or desired change and the purpose of the enterprise architecture. These underlying/independent dimensions have been indicated in row 1 and columns 1 to 2 of table 5. Thus, the matrix in Fig. 5 is template developed to help specify and reason about the dependent dimensions of the scope of an enterprise architecture effort with respect to the independent dimensions such as intended purposes of an enterprise architecture, resources available, and enterprise problem or desired change. The template in table 5 is used to represent stakeholder responses to the questions on architecture scope and purpose (see usage log in table 3, row 1.2.3)

Template for Enterprise Architecture Communication Plan (table 6): Effective communication with stakeholders on their expectations during and after the architecture effort and resultant deliverables is a very critical aspect in an architecture development roadmap.[22, 47] Thus, a communication plan needs to be developed to articulate the target stakeholders grouped according to communication needs and risks, communication mechanisms, and information sharing avenues, and communication schedule.[58] TOGAF[58] provides a template of a stakeholder map, that specifies sets of architecture deliverables or artifacts that are relevant for specific stakeholder categories. However, it hardly provides a matrix that can allow mapping on communication strategies/techniques with stakeholder categories. Thus, the stakeholder map template in TOGAF[58] has been adapted to derive columns 1 to 3 of the template in table 6 in order to address the gap. Columns 4 and 5 have been added to enrich the matrix with dimensions of communication strategy and communication scheduling. The contents of columns 2 and 5 of table 6 vary with respect to the enterprise situation, and can only be populated after attaining an understanding of the core issues of the enterprise baseline and target contexts.

Table 6. Template for Enterprise Architecture Communication Plan

Stakeholder category & its constituents	Specific communication needs during and after enterprise architecture creation (based on stakeholder's role, goals, & concerns)	Relevant deliverables that address the communication needs	Appropriate mechanism of engaging a stakeholder category or disseminating relevant information	Timelines
[Internal & external SML actors]				
[Internal & external TML actors]				
[Internal & external OML actors]				
[Internal & external OL actors]				

Column 4 is populated using mechanisms that will create shared understanding among stakeholders. These mechanisms were adapted from two techniques, i.e.: *committees and subcommittees*; and *take-a-panel and share-a-panel*. The "committees and subcommittees" technique revolves around using group labor effectively by decomposing activities into modular tasks that are delegated to subgroups, that are formed and assigned to pursue specific sub goals and deliverables within a given timeframe.[65] On the other hand, take-a-panel technique involves creating sub groups or panels that learn new skills and use them to solve a problem in a short period, while share-a-panel technique involves allocating each participant a timeslot to explain individual views to panel-mates on a problem and devised solution.[11] The formulation of these sub groups is based on the nature of tasks and expected deliverables, and the expertise and interests of stakeholders.[65] Thus, the two techniques were adapted to form three modes of stakeholder engagement, i.e.: *expert/specialization* based (targeting only subject matter experts), *governance* based (targeting managers or supervisors at different management levels), and *interest-task* based (targeting teams willing to accomplish a specific task). The template in table 6 is used to represent stakeholder responses to the questions on key decision makers in a specific architecture development initiative and their communication needs (see usage log in table 3, row 1.3.1).

The Mutual Scroll for Traceability Analysis and Negotiation: For each of the templates discussed above (i.e. Fig. 8 to Fig. 12 and tables 4 to 6), there is need to engage stakeholders in a negotiation and comprehensive analysis so as to create shared understanding on responses or content for each specified templates. To enable this, two mechanisms were further adopted, i.e.: traceability analysis and single negotiating texts. Traceability analysis is a reasoning mechanism used in requirements management to assess the direct and indirect relationships between requirements, their origins, and their implications on the design or structure of an artifact.[56] Single Negotiating Text is a preliminary concept or model that serves as a mutual frame for negotiating parties to continuously critique, such that the facilitator iteratively revises it until all conflicts are addressed and consensus is attained among negotiating parties.[65] These two mechanisms have been adopted and used to develop a *mutual scroll for traceability analysis and negotiation* as indicated in the last row of table 2.

The mutual scroll for traceability analysis and negotiation is a pool of all SUREAC templates and the two usage or questioning logs (presented using table 3 and table 8). The mutual scroll for traceability analysis and negotiation allows an individual member/stakeholder of a specific subgroup or panel to populate a template, then panel-mates critique and refine it until it addresses concerns and heterogeneous views about a given topic. Thus, the summary list of templates (in tables 2 and 7) and the usage or questioning logs (in tables 3 and 8) complement each other towards a mutual scroll, which can be perceived as a Single Negotiation Text that can allow elicitation and deliberation of requirements during enterprise architecture creation.

4.2.2. *Templates for Collaborative Design-Choice Sessions & Justifications*

Table 7 shows sub activities that constitute the collaborative design and choice sessions (originating from Fig. 7), method chunks selected to support execution of sub activities in the two sessions, and templates derived as a result of customizing selected chunks for use in architecture creation. Criteria used to select method chunks in table 7 were given at the start of section 4.2.1.

Table 7: Selected method chunks and templates in collaborative design and choice sessions

Sub activities in collaborative design and choice sessions (Nakakawa et al.[37])	**Selected method chunks to support reasoning and structuring of facts**	**Derived template to support documentation of facts in a consistent and traceable way**
2.1.1. Define key business capabilities and scenarios that constitute the target enterprise solution or venture[37]	• SSM's CATWOE and activity model techniques[12] • TOGAF ADM's guidelines on inputs for requirements	• Template for requirements and scenarios elicitation
2.1.2. Define key data capabilities for supporting the defined business capabilities and scenarios[37]	• Data-Information-Knowledge-Wisdom hierarchy framework[1, 54] • TOGAF ADM's guidelines on requirements for applications and technology architectures • Traceability analysis[56] • E-government four-stage model[32]	• Template for elicitation of enterprise data and information needs or requirements • Template for Synchronizing Application Requirements • Template for Specifying Interoperability of Application Capabilities • Template for Elicitation of Technology Requirements
2.1.3. Define key application capabilities for supporting specified business and data capabilities[37]		
2.1.4. Define key technology capabilities for supporting the specified application capabilities[37]		
2.2.1. Define constraints that the enterprise legal framework imposes on the defined capabilities[37]		
2.2.2. Define constraints that regulations & policies from external partners impose on the defined capabilities[37]		
2.3.1. Define inputs, outputs and inter linkages of scenarios[37]	• Business process management[60]	• Template for specifying and analyzing requirements and scenarios
2.3.2. Define performance indicators or quality criteria for the scenarios and capabilities[37]		
2.4. Validate constituted requirements for enterprise architecture[37]	• Traceability analysis[56] • Committees and sub committees & single negotiating text of collaborative decision making[65]	• Validation brief and guide for enterprise architecture views • *Governance based and specialization-based division of stakeholders* • *Mutual scroll for traceability analysis and negotiation* • *Template for enterprise architecture communication plan (the one used in task 1.3.1)*
3.1. Validate unit level architecture views[37]		
3.2. Validate enterprise level architecture views[37]		

To exhaustively undergo the analysis and deliberation required when executing sub activities in table 7, questions for guiding reasoning and analysis or deliberation are provided in the usage or questioning log presented in table 8. Stakeholders' mutual responses to questions in table 8 are documented using templates in table 7.

Table 8. Questioning Log for Guiding Reasoning in Collaborative Design and Choice Session

Tasks	Underlying questions of exploration when formulating requirements for enterprise architecture
2.1.1	• Which key business capabilities are required to achieve the target enterprise solution or venture or business functions, and which scenarios constitute these capabilities?
2.1.2	• Which data should be captured and processed by scenarios that constitute the defined business capabilities and function? Which instruments or tools are in use or should be introduced to capture the specified data or information? • Which information, knowledge facts, & knowledge patterns should be generated from data captured on scenarios that constitute the defined business capabilities and functions?
2.1.3.	• For each business capability or function, which applications will: o Support: (a) transaction processing, trends analysis, correlations or inferences that are relevant for evidence based planning and decision making within the enterprise; (b) quick dissemination of enterprise information or service requisition tools or reports to external associates (i.e. regulators, business partners, suppliers, and customers); (c) elicitation of information from external associates; (d) online and real-time information exchange or interaction with external associates; (e) vertical integration of processes to allow completion of transactions that cut across different hierarchical levels within a business function; (f) horizontal integration of processes across business functions to allow information sharing across functions and eliminate tendencies of capturing the same information by different functions so as to achieve integrated and proactive planning of service delivery initiatives? o Require input data/information from existing or planned applications of external associates in order to address an inter-organizational need or goal? o Will need or must send/dispatch their output to existing or planned applications of external associates in order to address an inter-organizational need? • For all the enterprise business capabilities or functions, which applications will require similar development technologies and can therefore be clustered, and what is the appropriate priority of each cluster in terms of development and implementation?
2.1.4	Which hardware, software, and security capabilities are required to support the specified application requirements, data requirements, and business requirements?
2.2.1	Which constraints do the enterprise principles, policies, standards, and values impose on the defined business, data, application, and technology capabilities?
2.2.2	Which constraints do regulations and policies from enterprise partnerships impose on the defined business, data, application, and technology capabilities?
2.3.1	• Which scenarios need to be integrated, need to be linked? • What are the inputs and outputs of scenarios that constitute capabilities?
2.3.2	What are the key performance indicators or quality criteria for scenarios, capabilities, and core functions that constitute them?
2.4.	• Which scenarios or capabilities are duplicated, misrepresented, incomplete, misplaced, not appropriately specified? • Which constraints are misinterpreted or not captured, and which additional constraints are necessary?
3.1.	• For each capability or unit, which views are misrepresented or incomplete with missing constraints? • Given the pros and cons of specific alternative unit-level views, which is the most appropriate unit or capability view at unit and at enterprise levels? Which trade-offs have been done at unit or capability level to accommodate enterprise level constraints?
3.2.	• Which components in the enterprise level views are misrepresented or incomplete with missing constraints? • Given the pros and cons of specific alternative enterprise-level views, which is the most

	appropriate unit or capability view at unit and at enterprise levels? Which trade-offs have been done at enterprise level to accommodate major constraints of units or capabilities?

Selected method chunks (in column 2 of table 7) and derived templates (in column 3 of table 7) for supporting the collaborative design and choice sessions are discussed below. A justification for each selected method chunk is also provided.

Template for Requirements and Scenarios Elicitation (Fig. 13): TOGAF ADM specifies key inputs for formulating requirements of an enterprise architecture and provides the business scenarios technique as a guideline for developing business requirements (see section 4.1 – step 2).[58] Although the business scenarios technique highlights mechanisms that can be used to elicit requirements, it hardly provides details on how to use the mechanisms to elicit and synchronize heterogeneous views on requirements from stakeholders. To address this gap, there was need to adopt SSM's CATWOE analysis and purposeful activity models. CATWOE analysis of a desired enterprise context or course of action involves determining its: corresponding Customers or clients, Actors to actualize it, Transformation process(es) that constitute it, World views that justify its significance, Owner(s) or sponsor(s) that govern its actualization, and Environmental or external factors that are likely to affect its realization.[12] To adopt this technique, five CATWOE analysis parameters (i.e. Customers, Actors, World views, Owners, Environmental factors) are represented using visualizations on the fives sides of the pentagon in Fig. 13, while the sixth parameter is represented in the top most inner corner of the pentagon (as the main desired change or Transformation process).

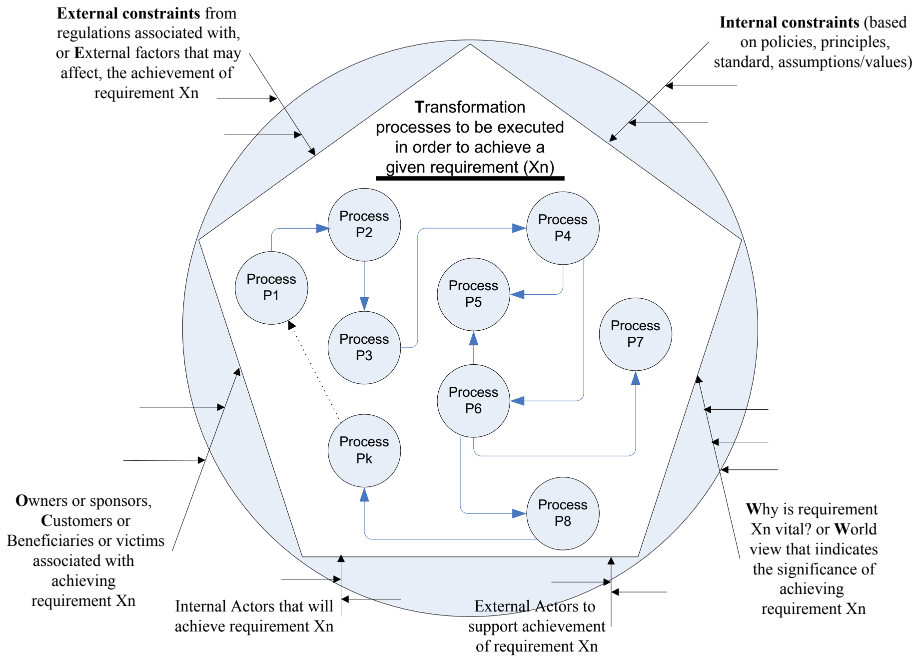


Fig. 13. Template for Requirements and Scenarios Elicitation (Nakakawa et al.[37])

In addition, a purposeful activity model (a technique of SSM) is obtained through assembling transformation processes for a desired enterprise context or course of action.[12] Thus, this technique was also adopted to derive the inner contents of Fig. 13 by providing a visual mechanism for specifying existing or desired scenarios that constitute the enterprise baseline or target contexts, towards achieving a given requirement or goal. Fig. 13 further shows how CATWOE analysis parameters can be used to detail and assess inputs for formulating requirements for an enterprise architecture. For example, the COE (Customers, Owner(s), Environmental factors) parameters help to derive architecture constraints from policies and standards; the Actors parameter helps to detail the domain-specific principles and specifications; the Transformation processes parameter helps to detail organization operational guidelines, and the World views parameter details the underlying assumptions. The template in Fig. 13 is used to represent stakeholder responses to the questions on scenarios that constitute business capabilities of an enterprise as key aspects for details requirements (see usage log in table 8, row 2.1.1).

Template for Specifying and Analyzing Requirements and Scenarios (table 9): Business requirements inform the design of a business architecture, which underpins other enterprise architecture domains by specifying the core attributes of an enterprise business environment such as product and service strategies, governance mechanisms, and business processes.[58] This demands that business requirements are comprehensive so as to yield a complete and unambiguous business process model. TOGAF's Business Scenarios technique highlights the need to specify constituents in the business and technical environment, process descriptions, actors and their roles, interrelationships, information flows, principles, and constraints of an enterprise.[58] However, details of how to document these aspects are implicit. The template in Fig. 13 attempted to provide an elicitation mechanism. However, there is need to further provide a mechanism of detailing aspects captured from stakeholders using the template in Fig. 13.

To provide guidance on how details of all key dimensions of business requirements can be structured or documented, there is need to understand how features of a business process model inform the formulation of business requirements. A business process model delineates specific ways of handling an occurrence or event by describing and assembling activities and sub activities that are triggered by an occurrence; and specifying fundamental dependencies, temporal attributes, data to be used, data to be created, constraints, and required resources.[60] Business requirements should, therefore, articulate the processes, sub processes, actors, tools/instruments, inputs, outputs, resources, interdependencies, and quality control criteria or indicators. Accordingly, the business process model concept in van der Aalst[60] was adapted to derive the layout of the template in table 9.

In table 9, content under columns 4 and 5 can be specified using names of tools used for data capture or collection and reporting such as forms, inventories, and reports; and/or other resources. In addition, prevailing trends demand that business process management systems use dashboards to reveal information on key performance indicators of organization initiatives with respect to predefined quality parameters.[60] Thus, business requirements need to also articulate quality indicators for specific processes, inputs, or outputs; and key performance indicators that are relevant for decision making in a particular business processes. This motivated the inclusion of the last column in table 9.

Table 9. Template for Specifying and Analyzing Requirements and Scenarios

Code & name of processes within a capability	Codes & names of sub processes within specific processes that constitute a capability & underlying constraints	Actors & Partners	Required input	Expected Output	Quality Indicator
[P1. Process name]	*[P1.1. Sub process name]* • *P.1.1.1. Underlying constraint*				
	[P1.2. Sub process name]				
[P2. Process name]	*[P2.1. Sub process name]*				
	[P2.2. Sub process name] • *P.2.2.1. Underlying constraint* • *P.2.2.2. Other underlying constraint*				

Although in Nakakawa et al.[37] an Ishikawa-based template is used for elaboration of business processes, using that template in a large enterprise leads to the challenge of text congestion and limited space for unpacking details associated with business requirements. The template in table 9 is used to represent detailed stakeholder responses to the questions on inputs, outputs, and performance indicators of business capabilities of an enterprise because these constitute detailed business requirements (see usage log in table 8, row 2.1.1).

Template for Elicitation of Enterprise Data and Information Requirements (table 10). Developing data architecture involves identifying existing and desired data elements or building blocks that are relevant in formulating the required logical and physical data properties and data management resources for supporting the target enterprise context.[58] Apart from formal data models, there is hardly a mechanism for supporting structuring of aspects that constitute data requirements in a way that is understandable to stakeholders who may be uncomfortable with the syntax and semantics of formal data models. Therefore, in order to enable comprehensive identification of the existing and desired data elements or building blocks in a traceable and understandable way, the Data–Information–Knowledge–Wisdom (DIKW) hierarchy was adopted to form the template in table 10.

According to Ackoff[1] and Rowley,[54] DIKW hierarchy comprises four aspects, where: (a) *Data* refers to relevant unprocessed and detailed representations of properties of events that occur in a given environment; (b) *Information* refers to treated data that is organized in a way that yields meaningful responses to contextual questions (such as who, what, where, how many, and when or how often) in a given environment; (c) *Knowledge* is a capability that synthesizes contextual information to yield a level of understanding that translates into directives that steer change in a given environment; (d) *Wisdom* is a capability that evaluates and uses interpretations and understanding obtained from knowledge of occurrences to increase efficiency and effectiveness in a given environment.

The adoption of DIKW hierarchy during the formulation of data requirements for an enterprise architecture, helps to create a holistic reasoning framework and pattern for guiding the identification and synchronization of baseline and target data and information needs for specific capabilities in an enterprise and for the enterprise as a whole. From table 10, column 1 represents business processes at all enterprise levels. Column 2 prompts stakeholders to articulate: (a) specific data required to execute each sub business process at operational level (row 5); (b) specific information that is required to guide operational managers to execute sub processes involved in their supervisory roles (row 4); (c) specific knowledge or synthesized information to allow tactical managers to effectively execute sub processes involved in their roles (row 3); and (d) specific knowledge patterns that executive or strategic managers need in order to steer enterprise growth (row 2). To properly structure content in column 2 of table 10, data capture tools that stakeholders specify in table 9 (under columns 4 and 5) can be referenced when listing data needs and information needs in rows 4 and 5 of table 10. Also, data reporting tools that stakeholders specify in table 9 can be referenced when listing information, knowledge, and wisdom related aspects in rows 2, 3, and 4 of table 10.

Thus, the template in table 10 helps organizational stakeholders to specify preliminary details that are crucial in generating or eliciting of data elements and corresponding data management resources for the baseline and target enterprise operations. Additional questions that probe discussions towards mutual responses for column 2 in table 10 are provided in the usage or questioning log in table 8 (see questions under task 2.1.2).

Table 10. Template for Elicitation of Enterprise Data and Information Requirements

Name of sub processes within a business function	Specifying Data, Information, Knowledge and Wisdom or applied knowledge (DIKW) needs for proper functioning of the capability/function	Application building blocks required to realize the specified DIKW needs
[Strategic management level sub processes]	***[Wisdom (knowledge patterns)*** *– Nth level of summaries and correlations required for planning e.g. key performance indicators of the function & possible interventions required]*	
[Tactical management level sub processes]	***[Knowledge facts (or proved information patterns)*** *– third and Nth level of summaries and correlations required for planning e.g. key performance indicators of units that constitute a function]*	
[Operational management level sub processes]	***[Information needs*** *– first and second level types of summaries required from the collected data about events in a function]*	
[Operational level sub processes]	***[Data needs*** *– type of data to gather about events in a function]*	

Template for Synchronizing Application Requirements (table 11): TOGAF[58] provides templates for specifying application and technology capabilities with respect to business capabilities. However, details are hardly provided on how to systematically and comprehensively derive and orchestrate basic elements or building blocks that constitute application capabilities, such that prescribed application requirements are traceable and understandable. Without such details, the elicitation and orchestration of application requirements and building blocks for application architecture only depends on insights from experienced enterprise architects. Thus, an *e-government maturity or growth model* is also adopted herein as an attempt to provide a systematic and traceable mechanism that can enrich the process of eliciting, analyzing, and synchronizing of elements or building blocks that constitute the application requirements.

Layne and Lee[32] indicate that e-government maturity or development entails four stages, i.e.: (a) catalogue – enabling information dissemination to clients; (b) transaction – supporting online processing of clients' requisitions; (c) vertical integration – enabling process integration across levels within particular functional systems; (d) horizontal integration – enabling process integration across diverse functional systems. This e-government maturity model can be envisioned as a reasoning framework for identifying the range of electronic services that can be offered within an enterprise and its environment (i.e. regulation authorities that oversee the enterprise, its business partners, its suppliers, and its customers). For these reasons, the four-stage e-government development model is adapted to inform the design of the matrix or template for synchronizing application requirements (as shown in table 11). Questions that probe stakeholder deliberations in order to yield mutual responses for populating cells in table 11 are provided in the usage or questioning log in table 8 (see questions under task 2.1.3).

Table 11. Template for Synchronizing Application Requirements

Enterprise business capability or function	Business process at respective management levels	Names of application capabilities required to yield DIKW needs & support (electronic) service delivery at each management level within each enterprise capability			
		Stage 1: Electronic Catalogue (information dissemination to key stakeholders)	Stage 2: Transaction (support for information elicitation & online transaction processing)	Stage 4: Vertical Integration (integration of transactional processes across different hierarchical levels within a business function or capability)	Stage 5: Horizontal Integration (integration of transactional processes across business functions for information sharing & elimination of redundancy towards proactive operations)
Capability 1	Strategic management level processes		X	T4	W1
	Tactical management level processes			Z1 T3	
	Operational management level processes			T2	
	Operational level processes		Y1	T1	
Capability 2	Strategic management level processes		K		W2
	Tactical management level processes			Z2	
	Operational management level processes				
	Operational level processes		Y2		

Each cell in table 11 (from columns 3 to 6) represents the name of the application element or capability required to support processes specified in cells of column 2. For example, cells marked Y1&Y2 in column 3 indicate the need to cluster the specific application elements (Y1 & Y2) if they require the same development technologies, so as to form a macro application capability Y_K that supports operational processes in business capabilities 1 and 2. The same semantics apply to cells marked W1-W2 or T1-T4 or Z1-Z2. However, cells marked X and K imply that required applications have completely different functionalities and therefore are to support different business capabilities, and the decision of whether to cluster them in a macro application capability depends on the nature of technologies and data required for their realization. In addition, the application names or capabilities plotted in columns 3 to 6 of table 11 may be derived from the application capabilities specified in the last column of table 10.

Template for Specifying Interoperability of Application Capabilities (table 12): Application capabilities that stakeholders specify in tables 10 and 11 need to be contextualized with respect to existing and/or already planned applications within the enterprise or at external partner agencies (i.e. regulatory authorities, donor or business partners, suppliers, or corporate customers). Thus, in addition to the template in table 11, a template is provided in table 12 to allow contextualization of application capabilities, so as to determine the expected information exchanges between applications and prevent duplications in functionalities and data redundancy. For example, names of (already existing and planned) external and internal applications are entered in column 2 of table 12, data exchanges are entered in columns 3 and 4. Columns 5 to 7 have been amended to capture details or specifics of the existing or already planned applications so as to inform the constraints and prioritization of application capabilities that are identified and specified during enterprise architecture creation. The template in table 12 helps to document additional stakeholder responses to cohesion-related questions in the usage log in table 8 (under tasks 2.1.2 and 2.1.3).

Table 12. Template for Specifying Interoperability of Application Capabilities

Name of macro application capability	Name of **existing or planned** application within the enterprise or its environment	Required **input from** already existing applications within the enterprise or existing & planned applications in the enterprise environment	Expected **output to** already existing applications within the enterprise or existing & planned applications in the enterprise environment	Source of application in column 2 – *internal or external?*	Status of application in column 2– *existing or planned?*	Type of Ownership *(i.e. enterprise itself, regulatory authority, business partners, supplier, customer, other)*
Application capability 1	*Internal app 1*			*internal*	*existing*	*enterprise itself*
	External app 2			*external*	*planned*	*business partner*
Application capability 2	*Internal app 3*			*internal*	*planned*	*regulatory agency*
	External app 4			*external*	*existing*	*supplier agency*

Table 13. Template for Elicitation of Technology Requirements

Enterprise function or business capability	Major business processes at respective management levels	Required Application capabilities	Required components of technology infrastructure
Business Capability 1	*Strategic management level processes*		
	Tactical management level processes		
	Operational management level processes		
	Operational level processes		
Business Capability 2	*Strategic management level processes*		
	Tactical management level processes		
	Operational management level processes		
	Operational level processes		

Template for Elicitation of Technology Requirements (table 13): To establish planned enterprise business and information systems services, there is need to specify the required software and hardware capabilities[58] that will yield quality information for evidence based planning and decision making. Thus, capabilities specified in preceding templates inform the elicitation and organization of existing and desired technology elements or components or building blocks, that are relevant for supporting the target enterprise context. To achieve this in a traceable way, a template has been derived as shown in table 13.

The template in table 13 extends the one provided by TOGAF[58] by amending columns 1 and 2 that prompt stakeholders to specify the business architecture building blocks that motivate the application capabilities in column 3 (derived from applications in table 10 and table 11) and technology capabilities in column 4. Thus, the specification of technology capabilities not only depends on application capabilities, but also business and data capabilities among others. The template in table 13 helps to document stakeholder responses to questions in the usage log in table 8 (under tasks 2.1.4).

Validation guide for unit-specific and enterprise-wide architecture views. To guide the execution of tasks 3.1 and 3.2 in table 7 and 8, table 14 presents a template for a validation brief and guide that architects can populate so as to guide stakeholder deliberations on critical architecture building blocks and design decisions; and their implications.

Table 14. Validation Brief and Guide for Enterprise Architecture Views

Name of business function or capability	Specific problem faced in baseline context or envisioned problem in target context	Specific building blocks or design decisions made in the architecture to address the existing and/or envisioned challenges	Implications of (not) adopting specific critical design decisions

Templates discussed in this section can be perceived as adapted method chunks for constituting SUREAC. Details of SUREAC assembling are provided in section 4.3.

4.3. *Assemble Method Chunks for SUREAC*

Situational methods need to be constituted in a modular pattern such that they are viewed as a collection of *interrelated autonomous components*.[50] To achieve this, at least two method chunks can be assembled using either *association strategy* – if the chunks achieve different intensions and the result of the first chunk is the source product of the second chunk, or *integration strategy* – if the chunks have similar goals but provide different ways to satisfy it.[51] Herein, association strategy was used to assemble SUREAC. This is because all method chunks adopted and the derived templates and patterns of working serve different purposes but their products are supplementary as discussed in section 4.2. This section presents a coherent view of SUREAC (in table 15 and Fig. 14).

Table 15. Assembling the Situational Method for SUREAC

Collaboration activity from Fig. 7	Name of template derived from adopting method chunks	Cross cutting patterns of working
1.1	• Set of symbols for guiding the design of an enterprise rich picture (Fig. 8) • Template for stakeholder categorization (Fig. 9) • Template for specifying stakeholder roles and constraints (table 4) • Combo template for contextualizing enterprise situation (Fig. 10)	• Governance based, specialization -based, interest-task

	• Template for problem analysis (Fig. 11)	based division of stakeholders • Mutual scroll for traceability analysis and negotiation *(i.e. logs in table 3 and table 8 and all templates in column 2 of this table – Fig. 8 to Fig. 13 and table 4 to table 14)*
1.2	• Template for elicitation of solution alternatives and implementation options (Fig. 12) • Template for specifying purpose and scope of enterprise architecture (table 5)	
1.3, 2.4, 3.1, 3.2	• Template for enterprise architecture communication plan (table 6) • Validation guide for enterprise architecture views (table 14)	
2.1, 2.2, 2.3	• Template for requirements and scenarios elicitation (Fig. 13) • Template for specifying and analyzing requirements and scenarios (table 9) • Template for elicitation of enterprise data and information needs or requirements (table 10) • Template for synchronizing application requirements (table 11) • Template for specifying interoperability of application capabilities (table 12) • Template for elicitation of technology requirements (table 13)	

Table 15 lists SUREAC templates and working patterns derived from adapted method chunks (as specified in tables 2, 3, 7, and 8 in section 4.2). Fig. 14 uses association strategy to show how the templates and working patterns are assembled for use during the three sessions that facilitate collaborative decision making during architecture creation, i.e.: collaborative intelligence – determining problems and appropriate solutions; collaborative design – developing detailed plans for realizing selected solutions; collaborative choice – choosing appropriate components of the plans for achieving selected solution. Nakakawa et al.[37, 38] discuss the detailed procedure of executing activities in these three sessions.

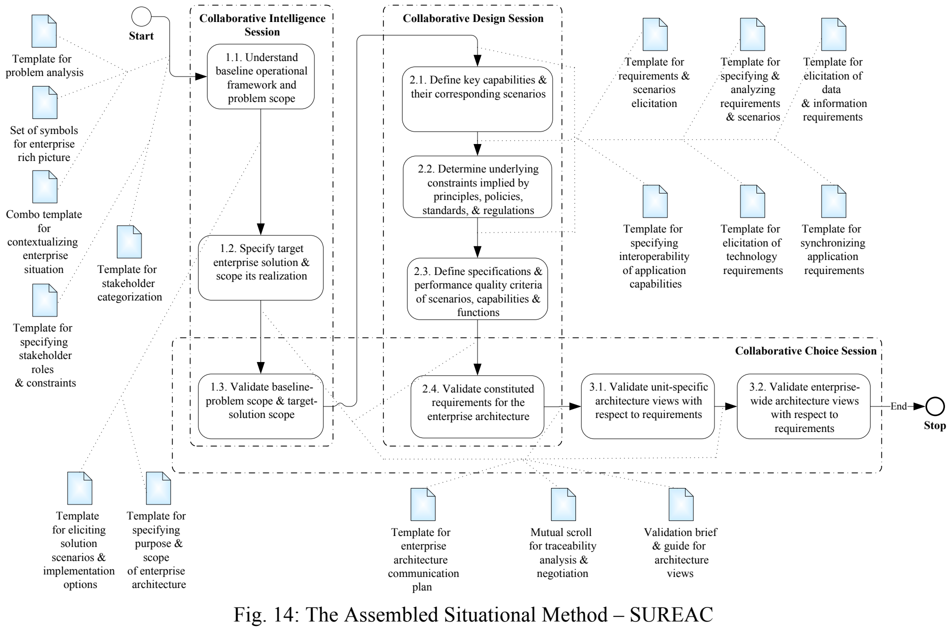


Fig. 14: The Assembled Situational Method – SUREAC

SUREAC provides templates that enable mutual comprehension of requirements by contextualizing the formulation of requirements with respect to their source, specifications, and implications on the architecture. Thus, effective use of SUREAC requires the use of the set of eight thinkLets or collaboration support modules that constitute CEADA – which is a broader method developed in our earlier work[37, 38] (see Fig. 2 in section 1, Fig. 3 in section 2, and Fig. 5 in section 3.1). SUREAC templates and working or reasoning patterns supplement the earlier efforts by providing a mechanism of addressing the feedback and recommendations presented in appendix B. Section 5 discusses the evaluation of SUREAC.

5. Evaluation of SUREAC

Design Science research demands systematic use of appropriate evaluation methods in order to reveal the quality attributes of the artifact and clearly demonstrate the research contribution to the knowledgebase of design foundations and methodologies.[26] Thus, as justified in section 2, Action Research was used herein to inform the systematic evaluation of SUREAC. With Action Research, utility of an artifact is examined and feedback on its performance is obtained through exposing it to real life societal or enterprise contexts.[26, 27, 68, 61] Accordingly, section 5.1 discusses how Action Research was conducted, while section 5.2 discusses major limitations/reflections.

5.1. *Set Up of Action Research to Evaluate SUREAC*

As indicated in section 1 (Fig. 2) and in the last paragraph of section 4, SUREAC is a component of CEADA and therefore can not be evaluated in isolation. Although CEADA was evaluated using several iterations,[70] it exhibited satisfactory performance in only six (6) enterprises or iterations. However, this paper presents only two enterprises/iterations in which CEADA manifested highest performance scores due to the additional plug-in of SUREAC. Thus, this section explains how Action Research was used to evaluate SUREAC in two enterprises, and thereafter provides a cross-sectional view of evaluation results before and after extending CEADA[37, 38] with the SUREAC method.

Susman and Evered[67] and Baskerville[4] indicate that Action Research involves five stages, i.e.: *Problem Diagnosis* – determining the need for transformation in a given societal or enterprise context; *Action Planning* – determining possible and appropriate action to address the need; *Action Taking* – deeply engaging subjects to implement required action; *Evaluate* – assessing effects of action taken; and *Specify Learning* – using knowledge obtained to improve context or theory. The instantiation of these stages to guide SUREAC evaluation in two enterprises based in Uganda is discussed in sections 5.1.1 to 5.1.4, and table 16 shows key highlights of this instantiation (see columns 1 to 3).

5.1.1. *Diagnosis Stage and Action Planning Stage*

Diagnosis stage involved two major tasks:

- Selecting enterprises to participate in the SUREAC evaluation effort. This involved sending a formal "request for participation" (describing what SUREAC evaluation would entail) to public entities that had no enterprise architecture programme; and selecting those that positively responded,[70] two of which are presented herein as indicated in table 16. Enterprise 1 is a national reference health laboratory that offers specialized laboratory testing services in Uganda, while enterprise 2 is a unit under Uganda's Ministry of Health that is responsible for regulating and supervising health laboratory services in Uganda.

- Determining a specific enterprise problem that could be considered in the SUREAC evaluation effort. Row 2 of table 16 highlights the problem that was considered in each enterprise.

Action Planning Stage involved the following three tasks:

- Determining the desired intervention/state to address the enterprise problem. Row 3 of table 16 highlights the desired state that was specified in each enterprise.
- Specifying the role of SUREAC in efforts towards actualizing the desired state. The shaded cell in row 3 of table 16 highlights this.
- Specifying the scope of the SUREAC evaluation effort in the enterprise based on the problem and desired state. In enterprise 1, eleven units that directly support the delivery of laboratory testing services were considered. In enterprise 2, eleven business capabilities that are responsible for directly supervising health laboratories in Uganda were considered.

Table 16: Instantiating Action Research to Evaluate SUREAC in Two Enterprises

Action Research Stages[4, 67]	**Enterprise 1 – Joint Clinical Research Center (JCRC)**	**Enterprise 2 – Central Public Health Laboratories (CPHL)**
1. Problem Diagnosis – *why is there need for change in the enterprise?*	A need to address information management challenges faced by all lab units and associated units in the enterprise, by implementing a Lab Information Management System (LIMS) that support specific needs of each unit	A need to achieve coordinated service delivery, effective data management, and effective communication with all health laboratories in Uganda by aligning business processes with IT
2. Action Planning – *which is the appropriate course of action to effect the desired change?*	Develop business and IT requirements for each unit to be supported by the desired LIMS, and use the developed requirements to customize an open-source LIMS for the enterprise	Develop business and IT requirements for addressing information management challenges in coordinating health laboratories in Uganda
	Specific purpose of SUREAC in the research effort at each enterprise: Use SUREAC to build shared understanding among stakeholders during the formulation of requirements that the enterprise architecture vision must accommodate	
3. Action Taking – *researchers closely work with subjects to implement action using the artifact*	SUREAC templates and reasoning patterns were used to elicit and document mutual views from stakeholders on information that was required to define requirements that had to be accommodated when designing the architecture vision of the two enterprises	
4. Evaluate – *actors judge the effect of action taken while using the artifact*	Checklist or questionnaire was given to participants. It contained parameters or evaluation goals or criteria (associated with shared understanding, communication, and negotiation) with response options derived from a five-point Likert scale where: (1) – strongly disagree, (3) – neutral, & (5) – strongly agree	

5. Specify learning – *use knowledge from stages 1 to 4 to improve artifact*	Lessons from evaluating SUREAC in the two enterprises informed its refinement: • Need to amend symbol set of rich picture (Fig. 9) & to avoid congestion in Fig. 11 • Need to find ways of coding input and output columns of template in table 9 • Need to continuously assess levels of shared understanding after each session

5.1.2. *Action Taking Stage*

This was set up and conducted as follows:[38, 70]

- *Aim of Intervention and Participants:* Aim is highlighted in row 4 of table 16. SUREAC evaluation effort involved 21 stakeholders in enterprise 1 and 15 stakeholders in enterprise 2.
- *Research Activities and Procedure of Stakeholder Engagement:* Stakeholders were engaged in turns using four major ways: (a) exploratory interview sessions and focus group sessions with heads of units or purposively selected members of units to populate diagram templates; (b) validation interview sessions and focus group sessions to validate (partially) populated diagram templates from the exploratory sessions; (c) workshop of key stakeholders to validate fully populated diagram templates from the focus group validation sessions; and (d) final validation workshop with selected key decision makers to consider the final populated and refined diagram templates that represent requirements for an enterprise architecture. The detailed step by step procedure of engaging stakeholders in each of these 4 sessions/workshops is provided by the CEADA process and script, that indicates *when* and *how* to invoke specific SUREAC templates. Since CEADA is comprehensively documented in earlier work,[38, 37, 70] the focus here is to demonstrate the improved performance of CEADA due to amendment of SUREAC templates.
- *Inputs and Outputs:* Diagram templates presented herein were the inputs to the action taking stage. The output was filled templates that articulate views on the requirements for an enterprise architecture of a given enterprise, and stakeholders' evaluation of the effort in a given enterprise.

5.1.3. *Evaluation Stage*

As indicated in table 16, involved providing stakeholders in the two enterprises with an evaluation checklist that was designed using evaluation goals that are associated with challenges of involving stakeholders in enterprise architecture creation. Challenges of interest are those plotted at the top of Fig. 1 in section 1 and in table 1. Thus, appendix C maps evaluation goals with specific challenges extracted from Fig. 1, in order to enable assessment of the extent to which specific challenges are accommodated in the design of SUREAC. Since CEADA and SUREAC are used jointly in an enterprise, evaluation goals listed below investigate support for all CEADA dimensions as indicated in Fig. 2, i.e.: collaboration (goals coded A1 to A3), communication (goal coded B), shared understanding (goals coded C1 to C4), and negotiation (goal coded D). However, the italicized goals are those that directly assess/indicate the relevance of SUREAC.

- A1. Participants' satisfaction with activities executed and how they were executed.
- A2. Support for constructive critical assessment of ideas generated by participants.
- A3. Participants' ability to understand objectives of the research engagement effort
- *B. Support for participants to freely express their views.*
- *C1. Support for increased participants' understanding of concerns and requirements that the architecture must address.*
- *C2. Participants' ability to understand results of the engagement effort.*
- *C3. Participants' ability to understand concerns/requirements of other participants.*
- *C4. Participants' satisfaction with results from the engagement effort.*
- *D. Participants' ability to understand why some of their views were not adopted*.

The type of the evaluation checklist that was used is specified in stage 4 in table 16. The statistic for performance measure was the *mean* of scores given by individual participants (against each evaluation goal). The performance indicator was the level of consensus (implied by the standard deviation of scores) among participants on the performance of the artifact with respect to each evaluation goal. Fig. 15 shows a line graph of performance values (on Y-axis) with respect to the evaluation goals/criteria (on X-axis). Since SUREAC is a component or module of CEADA, the line graph in Fig. 15 shows how evaluation scores of the latter increased after it was supplemented by the former.

The top part of Fig. 15 shows average performance scores before using SUREAC (see thick grey line) and after using SUREAC as an additional component to CEADA (see thick black line). The thick black line at the top of Fig. 15 shows improved overall performance. Also, the bottom of Fig. 15 shows the average standard deviation of scores assigned by participants before SUREAC use (see dashed grey line) and after SUREAC use (see dashed black line). The relatively low and stable trend of the dashed black line at the bottom of Fig. 15 indicates consistency in scores assigned by participants and therefore consensus among stakeholders on the performance of the artifact.

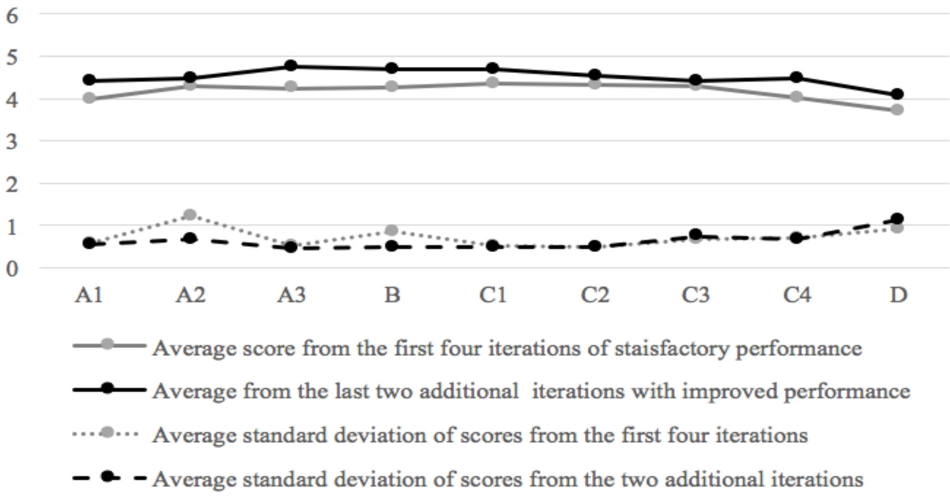


Fig. 15: Performance Scores from a Cross-Sectional Evaluation of SUREAC

5.1.4. *Specify Learning Stage of Action Research in SUREAC Evaluation*

As indicated in table 16, specify learning stage involved processing qualitative feedback from participants about the use of SUREAC templates; and reflecting over implications of feedback on artifact refinement. Qualitative feedback that informs artifact refinement from the two evaluation iterations indicated three major issues:

a) The need to amend the set of symbols for formulating an enterprise rich picture with symbols that probe one to indicate business functions, regulatory authorities, partners, programmes or projects, and existing information systems. If there are no symbols probing for these, then they are likely to be missing in the enterprise rich picture. This has been addressed (see Fig. 8 in section 3.2).
b) The template for problem analysis (in Fig. 11) becomes meshed with several arrows when a combination of some problem factors depicts a reinforcing or recursive pattern. This has not been addressed in the revised template because such systemic-like problems are better understood and represented using system dynamics causal loop diagrams.
c) There is need to devise means of coding input and output of processes in the template for specifying requirements and scenarios (table 9). This is because when table 9 is populated with actual data, it becomes too large because of the repetition cause by the fact that inputs of some processes are outputs of other processes; and some processes have similar inputs like other processes. Thus, the repetition of text in the input and output columns calls for systematic coding of contents therein so as to make the table less bulky and readable. Mechanisms of addressing this are still under pilot and will appear in future work.

Other qualitative comments include: the various number of templates that have to be filled in order to fully define requirements for the architecture; limited time allocated to populating the templates with data and validating resultant content; and lack of a mechanism of populating the templates using an automated tool.

5.2. *Reflections on the evaluation process of SUREAC*

Although SUREAC is a component of CEADA, it can not be evaluated in isolation. This is because: (a) the use of SUREAC templates is guided by the CEADA procedure; and (b) aspects of shared understanding can not be handled and evaluated separately from aspects of collaboration, communication, and negotiation. This is because these aspects are intertwined.[38] Thus, the relevance and value addition of SUREAC is reflected through the improved scores of evaluation criteria/goals that are associated with shared understanding, communication, and negotiation (i.e. B, C1 to C4, and D in figure 15).

Scores of these evaluation goals across the six enterprises reveal that a complete or full level of shared understanding is not attainable due to personality-related issues, but an acceptable range or level of shared understanding on architecture requirements can be reached. This has been demonstrated by findings from the evaluation of CEADA before and after SUREAC amendment. However, more research is needed to specify the various levels of group understanding, and indicators which can be relied on to show that an acceptable level of shared understanding of architecture requirements has been attained. Unfortunately, the evaluation of SUREAC was not done in a continuous approach to enable continuous assessment of the level of shared understanding among participants. Thus, there is need to design an evaluation checklist that can be used at the end of each CEADA-SUREAC session such that the level of shared understanding can be assessed at various stages from the start to end of a given engagement.

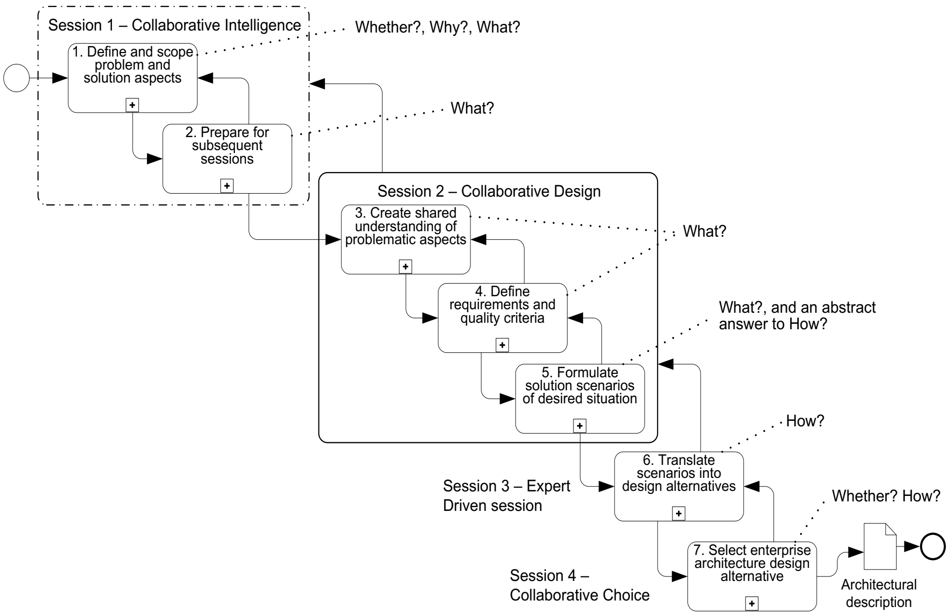
6. Conclusion and Future Work

This paper discusses how situational method engineering guided the joint use of components or chunks of approaches from fields such as enterprise architecture, collaborative decision making, and e-service delivery to address the gap of supporting the creation of shared understanding of requirements for enterprise architecture. Consequently, SUREAC has been devised to serve two purposes, i.e.: (a) support the elicitation and documentation of stakeholder views on key aspects that inform the formulation of requirements for an enterprise architecture; (b) guide group deliberations and reasoning on requirements for an enterprise architecture during its creation. Since evaluation findings indicate improved performance of the broader method that SUREAC supports, it can be concluded that SUREAC has to some extent achieved these two purposes. Thus, SUREAC can also be perceived as a method that answers the "how" details that arise when executing the "what-to-do" aspects in developing architecture requirements, which are articulated by architecture frameworks such as TOGAF.

Future Work. Further improvement of SUREAC points to four major aspects. First, there is need to further improve SUREAC by devising ways of adopting system dynamics causal loop diagrams to enrich understanding that arises from the template for problem analysis, as this would help to comprehensively assess dimensions of challenges in the baseline situation and those envisioned in the target situation. Second, there is need to develop a very detailed evaluation instrument for *continuously* assessing the levels of shared understanding as stakeholders progress in interacting with the various templates, as this would help to delve into more aspects that affect shared understanding as a key pillar for successful collaboration among enterprise architects and key stakeholders during architecture creation. Third, there is also need to devise an automated mechanism or interactive platform of supporting the population/filling and editing of the templates and increasing traceability of content across all templates, as this would supposedly help to combine some templates and improve their usability. Fourth, there is need to extend SUREAC with a method chunk that specifically supports an architect to assess and document the readiness of stakeholders to undergo transformation, by implementing the requirements of the target architecture that are documented using SUREAC templates.

Acknowledgement. Authors appreciate reviewers of the earlier versions of this paper for their insightful comments, and participants in enterprises used as case studies.

Appendix A: Model of activities for collaborative decision making in architecture creation (i.e. collaboration dependent tasks).[37, 38] The model below is an adaptation of the decision making model by Simon[66] (as discussed in Nakakawa et al.[37, 38]).



Appendix B: Specific changes made to the model above (in appendix A), so as to yield the SUREAC and improve granularity and specifics of tasks for defining requirements for an enterprise architecture.

#	**Feedback from earlier evaluation iterations** (from Nakakawa et al.[37])	**Changes made to accommodate concerns associated with formulating requirements for an enterprise architecture**
1	The model is at a very high level of granularity, with tables comprising several activities that are not evenly decomposed & batched (e.g. tasks listed in step 2 are few yet step 1 has 1.1 to 1.9)	The model's level of granularity and naming for each step has been refined, where each step contains not more than four (4) tasks. • Some aspects that were earlier presented as tasks have been accommodated by amending the templates with more fields to probe for (mutual) stakeholder responses.
2	In step 1 problem aspects are combined with solution aspects yet shared understanding must be sought on problem aspects for better output on solution aspects.	Step 1 has been split into steps 1.1 and 1.2 to separate problem aspects from solution aspects, and then rephrased to specify implied aspects.
3	It is not justifiable to separate steps 2 and 3, besides creating shared understanding should be a cross cutting activity throughout all steps	• Steps 2 and 3 have been merged into 1.3 • High level solution specifications are possible implementation strategies/options of a given business solution alternative, and this has been rephrased to reflect this meaning.
4	In the sub tasks of step 1, the distinction between business solution alternatives and	

	high level solution specifications is not clear	
5	Actual elements required in steps 4 and 5 are not clear in the model and in its activity tables, besides the template provided for requirements seems to only accommodate formulation of only business requirements and not other types of requirements (i.e. data, application, technology requirements)	Steps 4 and 5 have been decomposed into steps 2.1 to 2.4 so as to specify inputs needed for requirements formulation and to provide clear templates for defining the various types of requirements for an enterprise architecture. • DIKW hierarchy has been adapted to guide reasoning when formulating data and information needs; and e-services maturity model has been adapted to support reasoning when formulating application requirements.
6	Definition of quality criteria was not clear in step 4	Quality criteria now indicated as performance indicators or quality criteria in the refined tasks
7	There is no evaluation checklist provided to achieve step 7 effectively	The business, data, applications, and technology requirements serve as the detailed evaluation checklist for unit-specific architecture views and enterprise wide architecture views. • Consequently, evaluation checklist for requirements (step 2.4) and for problem-solution context (step 1.3) has been devised
8	Stakeholders consensus is needed on the customized architecture creation roadmap and execution plan for stakeholder engagements	The revised activity model includes task 1.3.3 to cater for building consensus on the architecture creation roadmap and communication plan
9	The diagram template for scenarios formulation & requirements elaboration does not specify institutional information assets that are key inputs or for formulating architecture requirements (as specified by architecture frameworks e.g. TOGAF)	This has been rectified in the revised template.

Appendix C: Mapping challenges in Fig. 1 with Evaluation Goals of SUREAC.

Shared understanding issues shown at the top of Fig. 1 and in table 1 (column 3)	**Codes of evaluation goals from section 5.1**
a). Difficult to bridge gap between abstract long term consequences & concrete examples for stakeholders to understand the impact of specific courses of action in an enterprise	C1, C2, C3
b). It is difficult for some stakeholders to imagine the new or future/desired/target situation of an enterprise	B, C1, C2, C3
c). Lack of documentation of existing practices and knowledge in an enterprise	C1, C4
d). Lack of common understanding of how business processes are to be aligned with ICT	C1, C2
e). Stakeholders communicate in abstract terms and they lack a common vocabulary for clearly expressing relevant details in the baseline & target contexts of an enterprise	C2, B
f). Lack of consensus on architecture results and how specific stakeholder concerns have been addressed	C4, C3, D
Although Fig.1 in section 1 shows several challenges, the focus of this paper is to assess ways of addressing the six challenges (a) to (f) that are plotted on the dotted lines at the top of Fig. 1. Thus, the mapping of evaluation goals in this table is only done for challenges and evaluation goals that are directly associated with shared understanding. Links between other challenges and the evaluation goals are traceable in earlier efforts.[37,70]	

References

1. R. L. Ackoff, From data to wisdom, *Journal of Applied Systems Analysis.* **16** (1989) 3–9.
2. D. F. Andersen, J. A. M. Vennix, G. P. Richardson and E. A. J. A. Rouwette, Group Model Building – Problem Structuring, Policy Simulation and Decision Support, *The Journal of the Operational Research Society*. **58**(5) (2007) 691–694.
3. G. Bafoutsou and G. Mentzas, Review and Functional Classification of Collaborative Systems, *International Journal of Information Management*. **22**(4) (2002) 281–305.
4. R. Baskerville, Investigating Information Systems with Action Research, *Communications of the Association for Information Systems*. **2**(3) (1999) 1–32.
5. R. O. Briggs, G.J. de Vreede and Jr. F. Nunamaker, Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems, *Journal of Management Information Systems*. **19**(4) (2003) 31–64.
6. R. O. Briggs, G. L. Kolfschoten, G. J. de Vreede and D. L. Dean, Defining key concepts for collaboration engineering, 12th Americas Conf. Information Systems, pp. 1–10, Acapulco, Mexico, Technologica de Monterrey, (2006).
7. S. Brinkkemper, Method Engineering – Engineering of Information Systems Development Methods and Tools, *Information and Software Technology.* **38** (1996) 275-280
8. S. Buckl and C. M. Schweda, On the State-of-the-Art in Enterprise Architecture Management Literature, Technical Report (Technische Universität München, Munich, 2011).
9. S. Buckl, F. Matthes and C. M. Schweda, A Method Base for Enterprise Architecture Management, In: Ralyté J., Mirbel I., Deneckère R. (eds), IFIP Advances in Information and Communication Technology, Vol. 351. Engineering Methods in the Service-Oriented Context. (Springer, Berlin, 2011).
10. S. Buckl, T. Dierl, F. Matthes and C. M. Schweda, Building Blocks for Enterprise Architecture Management Solutions. In F. Harmsen et al. (Eds.), LNBIP, Vol. 69, *PRET*, (Springer, Berlin, 2010), pp. 17-46.
11. Holland ASE Team, The Power of Group Dialog: Accelerated Solutions Environment (Capgemini, The Netherlands, 2005).
12. P. Checkland, *Systems Thinking, Systems Practice* (John Wiley and Sons, Chichester, 1999).
13. J. L. Dietz, *Enterprise Ontology* (Springer, Heidelberg, German, 2006)
14. A. Dietzsch, Adapting the UML to Business Process Modelling's Needs-Experiences in Situational Method Engineering, In J.-M. Jezequel, H. Hussmann, and S. Cook (eds.), Lecture Notes in Computer Science, Vol. 2460, *UML* (Springer, Berlin, 2002), pp. 73 – 83.
15. G. DeSanctis and R. B. Gallupe, A Foundation for the Study of Group Decision Support Systems, *Management Science.* **33**(5) (1987) 589–609.
16. A. J. DuBrin, *Essentials of Management*, 6th edn. (Peterborough, Ontario, Thomson South-Western, 2003).
17. C. Eden, On Evaluating the Performance of Wide-band GDSSs, *European Journal Operations Research* **81** (1995) 302-311.
18. C. Eden and F. Ackermann, Where Next for Problem Structuring Methods, *Journal of the Operational Research Society.* **57**(7) (2006) 766–768.
19. C. A. Ellis, S. J. Gibbs and G. L. Rein, Groupware – Some Issues and Experiences, *Communications of the ACM*, **34**(1) (1991) 38–58.
20. W. Engelsman and R. Wieringa, Goal-Oriented Requirements Engineering and Enterprise Architecture: Two Case Studies and Some Lessons Learned, In B. Regnell and D. Damian

(eds.), Lecture Notes in Computer Science, Vol. 7195, *REFSQ* (Springer-Verlag, Berlin Heidelberg, 2012), pp. 306–320.
21. J. A. Frechtling, *Logic modeling methods in program evaluation* (San Francisco, Jossey-Bass, 2007).
22. Gartner Inc., Gartner Identifies Ten Enterprise Architecture Pitfalls, Gartner Enterprise Architecture Summit 2009, http://www.gartner.com/it/page.jsp?id=1159617, (accessed on December 3rd 2016)
23. D. Greefhorst and H. A. Proper, Architecture Principles – The Cornerstones of Enterprise Architecture. (Springer, 2011).
24. J. Grudin, Computer Supported Cooperative Work–History and Focus, *IEEE Computer*. **27**(5), (1994) 19–26
25. A. F. Harmsen, *Situational Method Engineering* (Moret Ernst and Young Management Consultants, Utrecht, 1997).
26. A. R. Hevner, S. T. March, J. Park and S. Ram, Design Science in Information Systems Research, *MIS Quarterly*. **28**(1) (2004) 75–105.
27. A. R. Hevner, A Three Cycle View of Design Science Research, *Scandinavian Journal of Information Systems*. **19**(2) (2007) 87–92.
28. H. M. Ter Hofstede and T. F Verhoef, On the Feasibility of Situational Method Engineering, *Information Systems*. **22**(6) (1997) 401–422.
29. J. Iivari, A Paradigmatic Analysis of IS as a Design Science, *Scandinavian Journal of Information Systems*. **19**(2) (2007) 39–64.
30. K. Ishikawa, Guide to Quality Control, 2nd edn. (Asia Productivity Organization, Tokyo, 1986).
31. M. Lankhorst et al., Enterprise Architecture at Work: Modeling, Communication, and Analysis (Springer, Germany, 2005).
32. K. Layne and J. Lee, Developing Fully Functional e-government: A Four Stage Model, *Government Information Quarterly*. **18**(2) (2001) 122-136.
33. S. T. March and G. Smith, Design and Natural Science Research on Information Technology, *Decision Support Systems*. **15**(4) (1995) 251–266.
34. J. Mingers and J. Rosenhead, Problem Structuring Methods in Action, *European Journal of Operational Research*. 152, 530–554 (2004)
35. I. Mirbel and J. Ralyte, Situational Method Engineering: Combining Assembly- Based and Roadmap-Driven Approaches, *Requirements Engineering*. **11**(1) (2006) 58-78
36. D. D. Mittleman, R. O. Briggs, J. Murphy and A. Davis, Toward a Taxonomy of Groupware Technologies, In: Briggs, R.O., Antunes, P., Vreede, G.J. de, Read, A.S. (eds.), Lecture Notes in Computer Science, Vol. 5411, *CRIWG* (Springer, Heidelberg, Germany, 2008), pp. 305-317.
37. A. Nakakawa, P. van Bommel and H. A. Proper, Executing Collaboration Dependent Tasks in Enterprise Architecture Approaches – A Case of TOGAF, *Int. J. Coop. Inf. Syst. (IJCIS)* **22**(2) (2013) 1350007-1 to 1350007-79.
38. A. Nakakawa, P. van Bommel and H. A. Proper, Definition and validation of requirements for collaborative decision making in enterprise architecture creation, *Int. J. Coop. Inf. Syst. (IJCIS)* **20**(1) (2011) 83–136. doi:10.1142/S021884301100216X.
39. J. F. Nunamaker Jr., R. O. Briggs, D. D. Mittleman, D. R. Vogel and P. A. Balthazard, Lessons from a dozen years of group support systems research: A discussion of lab and field findings, *MIS* **13**(3) (1996) 163–207.
40. B. Nuseibeh and S. Easterbrook, Requirements Engineering: a roadmap, Conference on The Future of Software Engineering, pp. 35-46, Limerick, Ireland, (2000).
41. M. Op ‘t Land, H. A. Proper, M. Waage, J. Cloo and C. Steghuis, Enterprise Architecture – Creating Value by Informed Governance (Springer, Germany, 2008).

42. M. Op ‘t Land and H. A. Proper, Impact of Principles on Enterprise Engineering. In: Sterle, H., Schelp, J., Winter, R. (eds.), European Conference on Information Systems, pp. 1965–1976, St. Gallen, Switzerland, (2007)
43. G. Pervan, L. F. Lewis and D. S. Bajwa, Adoption and Use of Electronic Meeting Systems in Large Australian and New Zealand Organizations, *Group Decision and Negotiation*. **13** (2004) 403–414.
44. M. Pidd M, Tools for Thinking – Modelling in Management Science. (John Wiley and Sons, Chichester, 2009).
45. H. A. Proper, S.J.B.A. Hoppenbrouwers and G. E. Veldhuijzen van Zanten, Communication of Enterprise Architectures, In: Lankhorst, M. (ed.), Enterprise Architecture at Work: Modeling, Communication and Analysis. pp. 67–82 (Springer, 2005)
46. D. Quartel, W. Engelsman, H. Jonkers, M. van Sinderen, A goal-oriented requirements modelling language for enterprise architecture, IEEE International Enterprise Distributed Object Computing Conference, pp. 3-13 (IEEE Computer Society Washington, DC, USA, 2009)
47. B. van der, Raadt, S. Schouten and H. van Vliet, Stakeholder perception of enterprise architecture, Lecture Notes in Computer Science, Vol. 5292, European Conf. Software Architecture (Springer, Heidelberg, Germany, 2008), pp. 19–34.
48. J. Ralyte, Requirements Definition for the Situational Method Engineering, In C. Rolland, S. Brikkemper, and M. Saeki, editors, IFIP WG8.1 Working Conference on EISIC02, pp. 127–152. (Kluwer Academic Publishers, 2002).
49. J. Ralyté, Situational Method Engineering in Practice: A Case Study in a Small Enterprise, In: CAISE Forum, pp. 17-24. (2013)
50. J. Ralyte, Fundamentals and Challenges of Situational Method Engineering, 16th IEEE Conference on Business Informatics (2014).
51. J. Ralyte, R. Deneckere and C. Rolland, Towards a Generic Model for Situational Method Engineering, In: Eder, J., Missikoff, M. (eds.), Lecture Notes of Computer Science, Vol. 2681, *CAISE* (Springer, 2003), pp. 95-110.
52. E. A. J. A. Rouwette, J. A. M. Vennix and A. J. A. Felling, On Evaluating the Performance of Problem Structuring Methods: An Attempt at Formulating a Conceptual Model, *Group Decision and Negotiation.* **18**(6) (2007) 567–587
53. E. A. J. A. Rouwette and J. A. M. Vennix, System Dynamics and Organizational Interventions, *Systems Research and Behavioral Science*. **23**, (2006) 451–466
54. J. Rowley, The wisdom hierarchy: representations of the DIKW hierarchy, *Journal of Information Science,* **33** (2) (2007) 163–180.
55. J. Schekkerman, How to Survive in the Jungle of Enterprise Architecture Frameworks – Creating or Choosing an Enterprise Architecture Framework (Trafford Publishing, Canada, 2004)
56. I. Sommerville, *Software Engineering*, 7th edn. (Pearson Addison Wesley, 2004). ISBN:0321210263
57. Standish Group, *CHAOS Manifesto 2014 – Values versus Success and the Orthogonals* (The (Standish Group International Inc., 2014).
58. The Open Group Architecture Forum, *TOGAF Version 9* (Van Haren Publishing, The Netherlands, 2009).
59. S. H. Spewak, Enterprise Architecture Planning: Developing a Blue Print for Data, Applications, and Technology (John Wiley and Sons Inc, New York, 1992).
60. W. M. P. van der Aalst, Business Process Management: A Comprehensive Survey – Review Article, *ISRN Software Engineering.* (2013). http://dx.doi.org/10.1155/2013/507984
61. R. Wieringa, Design Science Research Methodology: Principles and Practice. Tutorial/Master Class on Design Science methodology, (School for Information and Knowledge Systems, Netherlands, 2010).

62. P. R. Smart, Understanding and Shared Understanding in Military Coalitions, University of Southampton (2011).
63. E. A. C. Bittner and J. M. Leimeister, Why Shared Understanding Matters - Engineering a Collaboration Process for Shared Understanding to Improve Collaboration Effectiveness in Heterogeneous Teams, in *HICSS* (IEEE Press, 2013).
64. P. Checkland, Soft Systems Methodology: A Thirty Year Retrospective, *Systems Research and Behavioral Science Syst. Res.* **17** (2000) S11–S58.
65. H. Raiffa, J. Richardson and D. Metcalfe, Negotiation Analysis – The Science and Art of Collaborative Decision Making, (Harvard University Press, Cambridge, 2002).
66. H. A. Simon, The New Science of Management Decision, (Harper and Row, New York, 1960).
67. G. Susman and R. Evered, An Assessment of the Scientific Merits of Action Research, *Administrative Science Quarterly.* **23**(4) (1978) 582–603.
68. J. Iivari, J. and J. Venable, Action research and design science research - Seemingly similar but decisively dissimilar, European Conference in Information Systems, pp. 1642-1653 (2009)
69. J. Venable, A Framework for Design Science Research Activities, Proceedings of the Information Resource Management Association Conference, USA, 21-24 May (2006), Idea Group Publishing, Hershey, Pennsylvania, USA.
70. A. Nakakawa, A Collaboration Process for Enterprise Architecture Creation, PhD Thesis, (Radboud University Nijmegen, 2012).
71. B. Henderson-Sellers and J. Ralyté, Situational Method Engineering: State-of-the-Art Review, *Journal of Universal Computer Science.* **16** (3) (2010) 424-478.
72. J. Ralyté, Requirements Definition for the Situational Method Engineering, Working Conference on Engineering Information Systems in the Internet Context, In C. Rolland, S. Brinkkemper, M. Saeki (Eds.), pp.127-152. (Kluwer Academic Publishers, 2002).
73. P. S. Seligmann, G. M. Wijers, H. G. Sol, Analyzing the structure of IS methodologies, Proceedings of the 1st Dutch Conference on Information Systems, Amersfoort, the Netherlands, 1989.
74. The Open Group (2017). ArchiMate 3.0.1 Specification, http://pubs.opengroup.org/architecture/archimate3-doc.