On a Cohesive Set of Requirements Engineering Terms

Michael J. Ryan, UNSW Canberra, m.ryan@adfa.edu.au
Louis S. Wheatcraft, Requirement Experts, louw@reqexperts.com

Abstract. The purpose of a requirement expression is to transform the needs of various entities into a formal language such that the intent is clearly understood by all involved. Although many sources provide definitions of the terms associated with a requirement expression, few contain guidance as to the process of developing a requirement, there are only occasional agreements on common definitions, and the defined terms are too narrowly focused to be useful across the full requirements engineering domain. This paper develops a cohesive set of definitions of the terms associated with a requirement expression. First, a framework for the transformation of needs into requirements is discussed and existing definitions are presented from the major relevant sources. These definitions are then analyzed and an integrated set of definitions is developed for the key terms associated with a requirements expression: entity; need; requirement expression; requirement statement; characteristics of a well-formed requirement statement and a well-formed set of requirements; and requirement attributes that, along with the requirement statement itself, comprise a well-formed requirement expression.

1. INTRODUCTION

The purpose of a requirement expression is to transform the needs of various entities into a formal language such that the intent is understood by all involved: those who are to implement the requirement, those responsible for proving the completed system meets the requirement, and those responsible for proving the resulting system meets the needs of the relevant entity. Writing a requirement statement is not simply an exercise in grammar; rather it is fundamentally an exercise in engineering. Requirements therefore do not just appear—a framework is essential for the formal transformation of needs into requirements and for the development of well-formed requirement expressions and well-formed sets of requirements. Unfortunately, however, there is no such agreed-to framework, nor is there consensus on the definitions or usage of the terms associated with the expression of a requirement.

Definitions of a requirement are contained in many sources, such as technical and process standards and books on systems engineering or requirements engineering. Most of the definitions, however, describe ‘what’ a requirement statement is, without including in the definition any action words that give the reader an insight into the thought processes that are required for the development of a well-formed requirement expression. Further, the definitions are provided by considerably different individuals or groups of individuals from varying domains, and with varying levels of experience. As a result, for those that do not quote a standard, the definitions tend to be unique to the writers and it is rare that any two agree. Additionally, it is rare that any two organizations have common definitions—while most would agree in a general sense; the specific terms are not common, nor commonly interpreted. Those definitions that are provided in standards and major sources are most commonly presented as assertions, without any accompanying intellectual underpinning, often at a middle-to-low level of the system development hierarchy. The result is therefore a disparate and inconsistent range of definitions that is not as useful as it needs to be.

Through a formal top-down process, this paper develops a cohesive set of definitions of the terms associated with a requirement expression. First, a framework is presented for the transformation of needs into requirements and existing definitions are presented from the major relevant sources. These definitions are then analyzed and an integrated set of definitions is developed for the key terms associated with a requirements expression: entity, need, requirement expression, requirement statement, characteristics of a well-formed requirement statement, requirement attributes, set of requirements, and characteristics of a well-formed set of requirements.

2. EXISTING DEFINITIONS OF NEEDS AND REQUIREMENTS

2.1 Needs and Requirements Views

When describing system development, some form of distinction is commonly made between ‘needs’ and ‘requirements’. Needs are typically considered to be expectations stated in the language of those at the business management level or of stakeholders at the business operations level. Requirements are considered to be formal statements that are structured and can be verified and validated. Requirements are generated from needs through

a process of requirements analysis (which is also called business analysis or mission analysis at the higher levels)—there may be more than one requirement defined for any need.

As illustrated in Figure 1 [1], needs and requirements exist at a number of levels—see also the "systems engineering sandwich" [2,3]. There is an enterprise view in which enterprise leadership sets the enterprise strategies in the form of a Concept of Operations (ConOps) or Strategic Business Plan (SBP); a business management view in which business management derives business needs and constraints as well as formalizes their requirements; a business operations view in which stakeholders define their needs and requirements; and a systems view in which the system to be developed is defined in logical and physical views. Subsequently, there are views at the lower-level of the subsystem and other system elements—note that a system may comprise a number of elements including products, people, and processes.

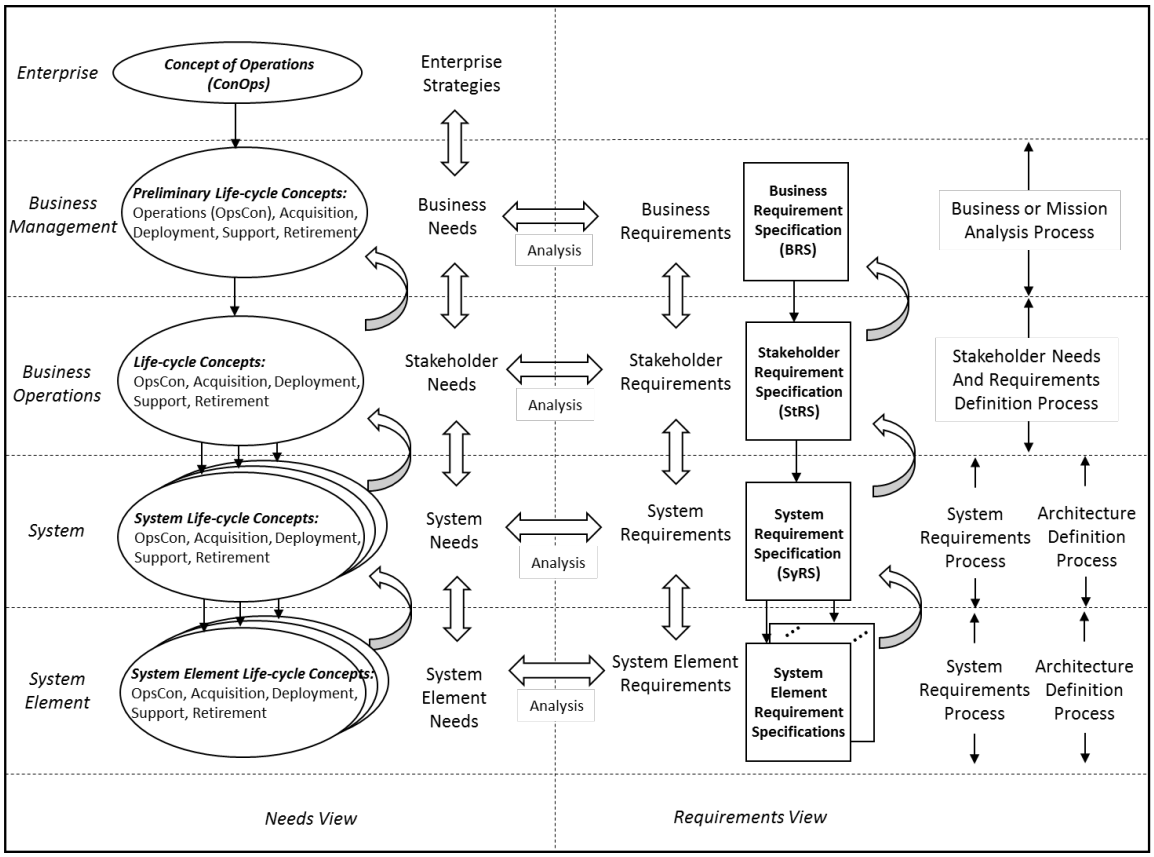

Figure 1. Transformation of needs into requirements (from [1]).

At the highest level, the enterprise has a number of strategies that will guide its future. A system has its genesis in the enterprise ConOps which communicates the leadership's intentions with regard to the operation of the organization—in terms of existing systems and systems to be developed. Some organizations may develop an SBP instead of a ConOps. At this level the ConOps, or SBP, defines the enterprise in terms of 'brand' and establishes a mission statement and corresponding goals and objectives, which clearly state the reason for the enterprise and its strategy for moving forward.

The Business or Mission Analysis Process begins with business management using the guidance in the enterprise ConOps or SBP to define business needs, largely in the form of a set of life-cycle concepts, which capture the business management's concepts for acquisition, development, marketing, operations, deployment, support, and retirement. Chief among these concepts is the business management Operational Concept (OpsCon). These concepts are used to define specific business needs for that level, which are then elaborated and formalized into business requirements, which are documented in the Business Requirements Specification (BRS) or Business Requirement Document (BRD). The process by which business needs are transformed into business requirements is called mission analysis or business analysis.

Once business management are satisfied that their needs and requirements are reasonably complete at the necessary level of abstraction, they are passed on to the business operations level. Here, the Stakeholder Needs and Requirements (SNR) Definition Process starts with the business management ConOps and the concepts contained in the life-cycle concepts as guidance. The requirements engineer (RE) or business analyst (BA) leads stakeholders from the business operations level through a structured process to elicit stakeholder needs and develop life-cycle concepts, which are documented in the form of a business operations OpsCon or similar document (see Figure 1). Structured processes used by the RE or BA to elicit specific stakeholder needs include user stories, use cases, scenarios, system concepts or operations concepts.

Stakeholder needs are then transformed into a formal set of stakeholder requirements, which are documented in the Stakeholder Requirement Specification (StRS) or Stakeholder Requirement Document (StRD). That transformation is guided by a formal process of requirements analysis. This requirements analysis can involve the use of functional flow diagrams, timeline analysis, N2 Diagrams, design reference missions, modelling and simulations, movies, pictures, states and modes analysis, fault tree analysis, failure modes and effects analysis, and trace studies.

At the system level, the RE or BA transforms the stakeholder needs and life-cycle concepts documented in the business operations OpsCon or similar document in to a system OpsCon or similar document. While the focus of the business operations OpsCon is from the stakeholders' viewpoint, the system OpsCon is from the system viewpoint. At this level, the System Requirements Definition Process, the system needs in the system OpsCon and requirements in the StRS are then transformed by the RE or BA into System Requirements for the system to be developed, which are documented in the System Requirement Specification (SyRS) or System Requirement Document (SyRD). As in the preceding processes, the RE or BA transforms needs into requirements using the same requirements analysis methods described above. At each level, the resulting requirements are documented, agreed-to, baselined and placed under configuration management. Note that some organizations may prepare individual life-cycle concepts for each of a number of systems that are to be developed to meet the business needs.

A set of requirements that has been documented, agreed-to, and baselined at one level will flow down to the next level as shown in Figure 1. At each level, the requirements are a result of the transformation process of the needs at that level as well a result of the elaboration (decomposition and/or derivation) of the requirements from the previous level. As such, a number of SyRS or SyRD requirements may be either decomposed from (that is, made explicit by the requirements of) or derived from (that is, implied by the requirements of) the system OpsCon and StRS or StRD. Consequently, a number of systems, each described by the system OpsCon and SyRS, may be defined to meet the capability required by the StRS. The same is true at the subsystem level, where a number of the subsystem requirements may be either decomposed or derived from the subsystem OpsCon and SyRS or SyRD. In all cases, for each level shown in Figure 1, the set of requirements can be traced back to the needs from which they were transformed as well as the requirements at the previous level from which they were either decomposed or derived.

From the preceding discussion, successful completion of the activities involved in transformation, analysis, decomposition, and derivation require a clear understanding of what is meant by a 'need' and a 'requirement' as well a number of associated terms. Unfortunately, these terms are not used with any great precision in the literature—even dictionary definitions blur the distinction by providing at least two definitions of a requirement as either a mandatory obligation or as something that is needed or desired. A useful set of definitions must not only state what each term is, but also provide guidance on how it should be created. Further, a useful taxonomy would comprise definitions that can be applied in a similar way to each entity at each level of the needs and requirements views illustrated in Figure 1.

2.2 Existing Definitions of a Need

Although most standards and texts assume the systems engineering processes begin with statements of a customer's "needs" or something similar, there are very few formal definitions of the term. For example, the premier requirements-engineering standard, ISO/IEC 29148 [4], does not define a need but simply states that "Defining requirements begin with stakeholder intentions (referred to as needs, goals, or objectives), that evolve into more formal statements before arriving as valid stakeholder requirements." ISO/IEC 15288 [5] doesn't make matters any clearer when it refers generically to the "needs, wants, desires, expectations and perceived constraints" of stakeholders. The Project Mangement Institute (PMI) does not define a need but define needs

assessment as “The domain of requirement management concerned with understanding business goals and objectives, issues, and opportunities, and recommending proposals to address them.” [57] Other standards assume the existence of, but do not define, a need or needs.

Similarly, many authors do not explicitly define needs but most refer to the generic issue of understanding the business owner’s problem (that is, what they ‘need’ in the general sense of the word) and then begin the requirements engineering process with ‘requirements’ that have been gathered (elicited) from users or stakeholders—see, for example, Robertson and Robertson [6] for a comprehensive description of a typical requirements-engineering process. In that sense, most authors treat needs and requirements as being synonymous—for example, Gottesdiener [7] states that “… the term requirements is broadly used to mean the needs of or conditions to be satisfied on behalf … users and suppliers”; Davis [8] describes a requirement as a “user need or a necessary feature, function, or attribute of a system”; and Ferdinandi [9] states that a requirement is a need to be satisfied by the product or service.

Other authors imply some progression from needs to requirements—for example, Young [10] suggests that customer needs and expectations are the source from which business and user requirements are derived. Larson and Larson [11] propose that the elaboration of requirements is assisted by moving from business requirements to technical requirements, from high-level requirements to detailed requirements.

Many authors describe needs using terms such as ‘goals’, ‘vision’, and ‘objectives’ (see, for example, Alexander and Beus-Dukic [12]; Laplante [13]; and Adzic [14]). In particular, Pohl [15] provides a detailed examination of goals and proposes a formal method for defining and documenting goals as a precursor to the elicitation of requirements.

Several authors do, however, define a need (sometimes called a goal) in some detail:

- “… a reflection of the business, personal, or operational problem (or opportunity) that must be addressed in order to justify consideration, purchase, or use of a new system” [16].
- “the lack of something desirable” and a need is always of a user—therefore, there are only ever user needs [17].
- “A goal is an intention with regard to the objectives, properties, or use of the system.” [15].
- “… an informal expression of something that has to be provided, ensured or avoided by a system or the development project of this system, from the viewpoint of one or several stakeholders.” [51] “Each need, if followed up, will lead to the development of a goal hierarchy.” [51]

2.3 Existing Definitions of a Requirement

As discussed in the previous section, after a general recognition of the “needs”, most sources propose that the starting point for system development is the elicitation and generation of requirements. For example, PMI [18] states “The project’s success is directly influenced by active stakeholder involvement in the discovery and decomposition of needs into requirements …” and “Requirements include the quantified and documented needs and expectations of the sponsor, customer, and other stakeholders.” Although all sources would tend to agree on the basic concept of a requirement and all acknowledge the importance of a proper set of requirements to successful development of a system, each of the major standards has a unique definition of a requirement statement:

- “Characteristics that identify the accomplishment levels needed to achieve specific objectives for a given set of conditions.” [19]
- “A statement that identifies a product or process, operational, logical, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (by consumers or internal quality assurance guidelines).” [20]
- “Something that governs what, how well, and under what conditions a product will achieve a given purpose.” [21]
- “A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.’ [22,23,57]
- “A statement which translates or expresses a need and its associated constraints and conditions.” [4], which goes on to say: “A well-formed requirement is a statement that: 1) can be verified, 2) has to be met or possessed by a system to solve a stakeholder problem or to achieve a stakeholder objective, 3)

defines the performance of the system when used by a specific stakeholder or the corresponding capability of the system, but not a capability of the user, operator, or other stakeholder [4]
- "A condition or capability that is required to be present in a product, service, or result to satisfy a contract or other formally imposed specification." [18]
- "The agreed upon need, desire, want, capability, capacity, or demand for personnel, equipment, facilities, or other resources or services by specified quantities for specific periods of time or at a specified time expressed as a "shall" statement." [24]

Most major sources (not identified explicitly here) quote one of the above standards when defining a requirement, but some (listed below) offer their own definition (or some modified standard definition) of a requirement:

- "A statement identifying a capability, physical characteristic, or quality factor that bounds a product or process need for which a solution will be pursued." [54]
- " … mandates that something must be accomplished, transformed, produced, or provided …" [52].
- "… capability needed by the user to solve a problem to achieve an objective" [25].
- "… capability that must be possessed by a system or system component to satisfy a contract, standard, specification, or otherwise formally imposed document" [25].
- "A user need or a necessary feature, function, or attribute of a system that can be sensed from a position external to the system." [8]
- "User needs plus set of all legal behaviors plus actual product's behavior" [8].
- "… how the system should behave, or a system property or attribute … [or] a constraint on the development process" [26].
- "A statement of a system service or constraint." [27]
- "A function or capability that a system must have to provide needed business support for the users of the proposed system." [28].
- "... the broad, top-level statements that describe the purposes and functioning of a system and its major components in support of client or enterprise needs and objectives. Requirements relate to the effective needs that have been identified in a needs assessment." [29]
- "… the effects that the client wishes to be brought about in the problem domain" [30].
- "… a need to be satisfied by the product or service" [9].
- "… the needs of or conditions to be satisfied on behalf of users and suppliers" [7].
- "A requirement is a necessary attribute in a system, a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have some value and utility to a customer or user." [10]
- "A requirement is a statement of a customer need—a statement that identifies a condition, capability, characteristic, or quality factor of a system that is necessary for the system to have value and utility to a user." [31]
- "A statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand- alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability". [32]
- "… a statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (by consumers or internal quality assurance guidelines)." [3]
- "… something the product must do to support its owner's business, or a quality it must have to make it acceptable and attractive to the owner" [6].
- "… a detailed expression of specific aspects of a less detailed stakeholder need, via the elaborated root goals of the latter." [51]

Although there is not a common definition, it is obvious from the above list that there are a number of common elements—that is, requirements are product/system characteristics, conditions and constraints that are unambiguous, testable and measurable. There are, however, unique elements: for example, only IEEE 610.12 [22], which is quoted by the Business Analysis Book of Knowledge (BABOK) Guide [23] and PMI [18], makes reference to the notion of a contract, only ISO/IEC 29148 [4] refers to a translation of a need into a requirement, and only NASA's NPR 7123.1B [24] uses the term "agreement", use of "shall" and "sets" of requirements. It should be noted that almost all of the above definitions are focused at the system/product level and do not

acknowledge the existence of other entities that may have requirements—such as enterprise, business management, business operations, or system elements.

3. EXISTING DEFINITIONS OF ASSOCIATED TERMS

In order to support the system development effort and particularly the requirements engineering effort, there are a number of other terms associated with needs and requirements. In particular, it is common to refer to types (also called categories) of requirements, characteristics of well-formed requirements and of sets of requirements, and attributes of requirements. This section describes how these associated terms are defined and used in major sources.

3.1 Existing Definitions of Types of Requirements

Types (often referred to as categories) of requirements do not directly affect the definition of a requirement but the concept can be useful in the transformation process, can be used to organize requirements, and can be used as an indication of completeness of a requirement set. From a completeness perspective, when transforming needs into requirements, the categories of requirement can be used to focus attention on various perspectives of the system to make sure all areas have been addressed resulting in a complete set of requirements. Each organization should define, document, and agree on how types or categories will be used to classify and organize their requirements.

The following summarizes the lists of types of requirements where they are provided by major sources:

- purposeful, functional, non-functional (security, modifiability, testability, reliability, portability and others) [33].
- system, user interface, database, communications, security [26]
- functionality, external interfaces, performance, design constraints [34].
- functional requirements, non-functional requirements, and design constraints [16].
- functional and non-functional (which includes qualities and constraints) [27].
- business, quality, performance, interface, functional (or behavioral or operational), non-behavioral (designability, portability, reliability, efficiency, human engineering, testability, understandability, and modifiability) [35].
- functional and non-functional (including constraints) [36].
- functional and non-functional (performance, design constraints, commercial constraints, preferences) [30].
- business, user, product, environmental, unknowable, high-level (or system-level), non-functional (system properties and -ilities), derived, design constraints, performance, interface, subsystem, component, verified, validated, and qualification (recognizing that some should not be used because they tend to create confusion and misunderstanding) [10]
- functional, non-functional, goal level, domain level, product level, design level, primary, derived, business, product, role-based (customer, user, IT, system, security) [37].
- functional, performance, and constraint [20].
- functional, performance, quality (resource saving, workload capacity, resource), resource, and design constraint [38].
- business, functional, performance, product, engineering, design, interface [31].
- functional; supplemental (non-functional), including look and feel, usability, operational, maintainability and portability, security, and cultural and political ; and constraints (technical and business) [39].
- functional, non-functional, domain [13].
- business, technical, high-level, functional, non-functional (performance, quality, availability, reliability, retention/purge, security, confidentiality and privacy, regulatory / legal / compliance, physical, safety, operational, scalability, data integrity, business continuity, usability) [11].
- functional, non-functional (which includes quality, compliance, architectural, development) [40].
- functional requirements, quality requirements, and constraints [15].
- functional, performance, interface, design, process, non-functional (quality and human-factors) [4].

- functional (an action), non-functional (properties or qualities: look and feel, usability and humanity, performance, operational, maintainability and support, security, cultural and political, legal), constraints [6].
- business requirements (business and project objectives, business rules, guiding principles of the organisation); stakeholder requirements; solution requirements (functional, non-functional, technology and standards compliance, support and training, quality, reporting); project requirements; transition requirements; assumptions, dependencies, and constraints [18].
- functional, performance, operational, -ilities (quality including safety and security), physical characteristics, and design and construction standards. [41].
- functional and non-functional (which are constraints) [50,51].
- functional/performance, design, environmental, suitability [55].
- functional (behaviors) and non-functional (properties including interface, environment and quality attributes) [57]

Once a certain set of types or categories has been defined, they can be included in the set of attributes that make up the requirement expression, which is how they are treated here. Consequently, we do not consider types of requirement any further in this paper, but include type as one of the attributes of a requirement which are discussed in more detail in the following sections.

3.2 Existing Definitions of Characteristics of a Requirement and of a Set of Requirements

It is common to refer to the characteristics of individual requirements that can be used as a measure of a well-formed ('good') requirement. Again, each source provides a slightly different list of such characteristics:

- type (primary, derived), application (product parameter, program parameter), compliance level (mandatory, guidance, information), priority [52].
- necessary, verifiable, attainable, clear [53].
- necessary, concise, implementation free, attainable (achievable or feasible), complete, consistent, unambiguous, standard constructs, verifiable [54]
- internally consistent, non-ambiguous, externally consistent, minimal, complete, non-redundant [42].
- understandable, non-redundant, complete, unambiguous, consistent, organized, conform to standards, traceable [27].
- the acronym SMART is commonly used: specific, measurable, attainable/achievable, reliable/realistic, and testable/traceable/time-constrained [11].
- feasible, valid (correct), unambiguous, verifiable, modifiable, consistent, complete, traceable [43].
- clear, concise, complete, confirmed, consistent, testable, traceable [11].
- necessary, verifiable, attainable, unambiguous, complete, consistent, traceable, allocated, concise, implementation free, standard constructs, unique identifier [35].
- complete, correct, feasible, necessary, prioritized, unambiguous, and verifiable [44].
- clarity, deliverable, testable, itemized, traceable, ambiguous, consistent, identifies assumptions, stated in the positive, identifies objectives, identifies major functions, identifies business rules, alternative consequences defined, magic words, complete [45].
- necessary, feasible, correct, concise, unambiguous, complete, consistent, verifiable, traceable, allocated, nonredundant, written using a standard construct, assigned a unique identifier, devoid of escape clauses [10,31].
- necessary, implementation independent, unambiguous, conforming, complete, singular, achievable, verifiable [32].
- necessary, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, verifiable [4].
- necessary, implementation independent, unambiguous, conforming, complete, singular, feasible, verifiable, correct [46].
- complete, relevant, testable, coherent, traceable, and several others (unspecified) [6].
- "Acceptable form for a requirement statement is individually clear, correct, feasible to obtain, unambiguous in meaning, and can be validated at the level of the system structure at which stated." ([24]
- necessary, attainable, clear and unambiguous, verifiable, not premature design, complete [51].

In addition to the characteristics of well-formed individual requirements, it is common to list the characteristics of a well-formed set of requirements. The following summarizes the descriptions of the characteristics of sets of requirements from major sources:

- complete, consistent, feasible, testable (although they were called 'verification and validation criteria' by Boehm, rather than characteristics of a set of requirements) [47].
- communicable, true, complete, feasible, verifiable, maintainable [17].
- complete and consistent plus supporting characteristics of: project unique ID, level, category, compliance level, allocated to, parent UID, source, specification number, rationale, verification method, verification documents, change notices, risk, TPM parameter, current status, standards [54]
- complete, consistent, modifiable, and traceable [44].
- feasible, valid (correct), unambiguous, verifiable, modifiable, consistent, complete, traceable, concise [43].
- complete, consistent, affordable, bounded [4].
- complete, consistent, feasible, bounded [46].
- "In pairs of requirement statements or as a set, collectively, they are not redundant, are adequately related with respect to terms used, and are not in conflict with one another." [24]
- complete, consistent, non-redundant, structured, validated, approved [51]
- accuracy, class, complexity, conciseness, conformance, correctness, criticality, level, priority, risk, unambiguousness, verifiability [55].

In some sources the descriptions refer to both the characteristics of individual requirements as well as the set of requirements (without making a distinction between the two):

- correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable [34].
- understandable, non-redundant, complete, unambiguous, consistent, organized, conform to standards, traceable [27].
- correct, unambiguous, complete, consistent, ranked for importance and stability, verifiable, modifiable, traceable, and understandable [16].
- allocatable, attainable, complete, concise, consistent, correct, design-independent, feasible, measurable and testable, modifiable, necessary, organized, prioritized, traceable, unambiguous, understandable [39].
- affordability, boundedness, completeness, consistency, orthogonality [55].

Although these characteristics are intuitively appropriate, they are not presented with any intellectual underpinning—they are, despite their intuitive appeal, assertions. In the following sections, we provide a logical development of the characteristics of both a well-formed requirement expression and a requirement set.

3.3 Existing Definitions of Attributes of a Requirement

ISO/IEC 29148 [4] defines an attribute of a requirement as an "inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means" and goes on to say "To support requirements analysis, well-formed requirements should have descriptive attributes defined to help in understanding and managing the requirements. The attribute information should be associated with the requirements in the selected requirements repository." Larson and Larson [11] add that "Attributes provide information about the requirement, such as the source, the importance of the requirement, and other facts. Attributes aid in the ongoing management of the requirements throughout business analysis." Young adds that "An attribute is a characteristic of a requirement that is useful in sorting, classifying, and managing requirements." [10]. Larson and Larson [11] note that the benefit of defining these attributes is to provide a means to validate the requirements as well as to provide metrics that will help to show the status of requirement development activities.

The following summarizes descriptions of the attributes of requirements from the literature:

- identifier, date entered, date changed, sources, rationale, status, dependents, is-dependent on, model links, comments [27].
- (of features) status, priority/benefit, effort, risk, stability, target release, assigned to, reason [16].

- source of requirement, acceptance status, urgency, priority, verification method, constraints, questions associated with the requirement [36].
- origin or source, creation date, author, person responsible for implementation, status, rationale, priority, owner, product release number the requirement is allocated to, architecture elements the requirement is allocated to, verification method, stability, and version number [44].
- unique ID, source, owner, rationale, priority, status, cost, difficulty, stability, assigned to, location, author, revision, date, reason, traced from, traced to, root tag, history, verification, validation, release, module (plus others might exist automatically: name, created by, created on, created through, last modified by, last modified on, absolute number, link mapping, object heading, object text, object short text, description) [10].
- unique identifier, acceptance criteria, author, complexity, ownership, performance, stability, urgency, business value, status, type, priority, and source [39].
- requirement quality attributes (rationale, verification method, traceability, and allocation) as well as requirement management attributes (owner, author, risk, priority, requirement validation status, and change status [11].
- identification (identifier, name); context relationships (source, context facet, reason, responsible person, using stakeholders); documentation aspects (documentation formats, documentation rules, specification formats, specification rules, validation status of the documentation, validation status of the specification); content aspects (requirement type, short description, additional information, cross references, status of the content, validation status of the content); negotiation aspects (negotiation status, validation status of the achieved agreement, identified conflicts, decisions); and validation aspects (compliance with entry criteria, validation techniques, current validation step, overall validation status [15].
- identification, stakeholder priority, dependency, risk, source, rationale, difficulty, type (such as functional, performance, interface, design constraints, process requirements, non-functional (quality and human factors)).” [4]
- identification (identifier, name), intrinsic characteristics (basic type, quality factor sub-type, product/process type, quantitative/qualitative type, life-cycle phase); priority and importance (priority, importance); source and ownership (derivation type, source, owner, approval authority); context (requirements set/document, subject, scope); verification and validation (V&V method, V&V stage, V&V status, satisfaction argument, validation argument); process support (agreement status, qualification status, satisfaction status, review status); elaboration (rationale, comments, questions, responses); miscellaneous (maturity (stability), risk level, estimated cost, actual cost, product release) [3].
- identifier, name, description, version, author, source, stability, criticality, priority [48].
- requirement number, requirement type, event/business use case/product use case, description, rationale, originator, fit criterion, customer satisfaction/dissatisfaction, priority, conflicts, supporting materials, history [6].
- unique ID, created by, object type, requirement statement, rationale, owner, stakeholder, requirement type, scenario, use case, function, fit criteria, version, priority, additional information, assumption, source, requirement status, means of requirement verification, requirements approval status, means of design verification, design verification status, means of product verification, product verification status, allocation [51].

As can be seen from the above list, there are a very large number of possible attributes that can be appended to a requirement statement. Again, while each of the attributes is intuitively appealing, there is no logic underpinning their selection and no structure provided for their consideration and use.

4. SHORTFALLS IN EXISTING DEFINITIONS

As is evident in the preceding section, there are a number of shortfalls in the existing definitions of terms associated with a requirement expression:

- There are a large number of definitions that provide a variety of base definitions, as well as significant differences in use of terms such as “characteristic” and “attribute” as they apply to requirement statements.

- Almost universally, major sources assume that there is one level of need (called variously business needs, goals, and objectives) and there are only two levels of requirements: user/stakeholder requirements and system/product requirements with the more modern standards adding a third business level. For example, Wieringa [17] states that there are only ever user needs and Dorfman and Thayer [25] define a requirement as a capability required by a user. As illustrated in Figure 1, the terms must be able to be applied to a much wider context— as shown in Figure 1, needs and requirements exist from the business management level to the system element level.
- Similarly, major sources assume that requirements are of the single system. Even if that were the only level of interest, the focus is too narrow because a single StRS may well define a capability required by the business that is to be implemented through the delivery of a number of systems (each of which is described in its own SyRS) as well as processes and people.
- The entity that is the subject of the requirement statement can vary further. Requirements can refer to either the system under development or the organization responsible for the development—the requirements on the organization belong in a project plan or statement of work; the system requirements belong in a system requirements document or specification.
- The various terms are used reasonably loosely or at least very narrowly. A much more precise ontology is needed.

These issues are addressed by the definitions derived in the following section.

5. PROPOSED DEFINITIONS

5.1 Proposed Definition of an Entity

We noted in the introduction that we seek definitions that apply to the development of needs and requirements for all entities at all levels of needs and requirements in the hierarchy of Figure 1. Terms such as system and subsystem (system element) are level-specific, so an improved set of definitions must start with the definition of a term that can apply at any level and to any single thing at that level, whether an enterprise, business unit, system, or system element (which could be a product, process, human or organisation). So we define an *entity* as:

> An *entity* is a single thing to which a need or requirement refers: an enterprise, business unit, project, system, or system element (which could be a product, process, human, or organisation).

The entity can be both the subject of any need statement or requirement statement at the level of that entity as well as the supplier/acquirer responsible for realizing the need or implementing the requirement. That is, in a business requirement ("The business shall …"), the business is not only the subject of the requirement but is also the party responsible for the implementation.

5.2 Proposed Definition of a Need

At any level, an entity is associated with a number of concepts that are the result of its environment, its intended use, or the expectation(s) of the other entities at the higher, lower, or peer levels. Entities therefore have needs, which we can define as:

> A *need* is the result of a formal transformation of one or more concepts into an agreed-to expectation for an entity to perform some function or possess some quality (within specified constraints).

5.3 Proposed Definition of a Requirement Statement

As illustrated in Figure 1, the two primary actions associated with the development of requirement statements are the 'transformation' process that turns 'needs' into 'requirements' (the left to right flow from needs to requirements) as well as the 'agreement' process between levels of requirements (the vertical flow of needs and requirements from one level down to another as well as the traces of requirements at one level to the needs at that level and to the requirements at the previous level). Figure 1 illustrates there is a formal transformation of one or more needs into a structured statement that, for an entity, describes a function or task to be performed; the extent or how well, and under what conditions a function or task is to be performed or the entity is required to operate; and the constraints that limit design options or the properties of the entity. Further, a requirement

represents an agreement to which the entity will be held. An appropriate definition of a requirement statement is therefore:

> A *requirement statement* is the result of a formal transformation of one or more needs into an agreed-to obligation for an entity to perform some function or possess some quality (within specified constraints).

A requirement statement is a single succinct sentence written in a standard format—a well-formed requirement statement has a number of characteristics (discussed in more detail in the following sections) and conforms to a number of rules for writing well-formed requirements (see the INCOSE Guide for Writing Requirements [46]).

5.4 Proposed Definition of a Requirement Expression

A requirement is more than just a requirement statement. The full expression of a requirement includes associated attributes that aid in the management of the requirement (the attributes may not be visible with the requirement statement but may be appended to it in the requirement document or in an accompanying or hosting database). So, we can define a requirement expression as:

> A *requirement expression* is comprised of *a requirement statement* and a set of associated *attributes*.

5.5 Derivation of the Characteristics of a Well-formed Requirement Statement

If a requirement statement results from a formal transformation of one or more needs into an agreed-to obligation for an entity responsible for implementing the requirements, those two key elements can be elaborated further in order to derive specific characteristics of a well-formed requirement that meets those conditions:

- Formal Transformation. For each need the RE or BA determines what the entity to be developed has to do in order for the need to be realized. Reference to a formal transformation makes it clear that a requirement is the result of formal engineering analysis, which is referred to as either mission analysis, or requirements analysis. Given that the requirement is a result of a formal transformation, the following characteristics of a well-formed requirement can be derived:
  - *Necessary*. The formal transformation must result in a requirement that is necessary in order to meet one or more needs. Once each requirement is proven to be necessary, the set of requirements must represent a sufficient solution to the corresponding set of needs.
  - *Singular*. The formal transformation is a one-to-one or a many-to-one transformation, so each requirement statement must represent a single thought. Since a single thought must be expressed in a single sentence, so too must a requirement statement,
  - *Conforming*. For the transformation to be formal, the resultant sentence structure must be formal. A requirement must conform to a standard (internal to the organization or external) that defines the formal sentence structure and rules governing the formulation of the requirement statement. For example, all requirements may be required to have the formal sentence structure: The <subject clause> shall, when <condition clause>, <action verb clause> <object clause> <optional qualifying clause>.' [46]
  - *Appropriate*. The requirement must be sufficiently detailed for the level at which it is stated, and not be any more detailed or specific than is necessary for the level at which it is stated. In particular, the subject of the requirement must be appropriate at that level. Consequently, unless there is a good reason, a requirement should refer to the entity at the level of the requirement (not higher or lower). For example, a requirement statement at the system level would begin: "The {*system*} shall …" and a requirement at the system element level would begin: "The {*system element*} shall …". At levels above the system level, the entity can be the organization responsible for developing the system; in which case, the requirement would begin: "The {organization} shall ……"
  - *Correct*. The transformation must be formal and able to be verified and validated. The requirement must communicate correctly that which is necessary for the need to be met and must be able to be shown that achievement of the requirement (as it is written) will result in meeting the need(s) from which it was transformed.

- Agreed-to Obligation. This aspect of the definition makes it clear that, before a requirement is valid, both the customer and provider must agree with the requirement statement. Many people may want to levy requirements on an entity, but until that requirement is formally agreed-to and is part of a contract, it is not a valid requirement for the entity. If the requirement is to be a part of a fair agreement to meet an obligation, the following characteristics of a requirement can be derived:
  - *Unambiguous*. An agreement is difficult to enact unless both parties are clear on the exact obligation. Consequently, a requirement statement must lend itself to a single interpretation of intent.
  - *Complete*. An agreement is not useful unless the obligation is complete and does not need further explanation. Each requirement statement must therefore be complete. For example, if a requirement implies an interaction between the system under development and another system, there is an interface. The requirement is not complete unless the requirement references where the interaction is defined. "The {system} shall {interact} with {the other system} as defined in {document where the interaction is defined}. Without this information the requirement is not complete and thus not verifiable.
  - *Feasible*. There is little point in agreeing to an obligation for a requirement that is not feasible. Each requirement must therefore be feasible within the appropriate constraints including cost, schedule, technology, legal and ethical.
  - *Verifiable*. Unless a requirement is in some way verifiable or testable, there is no way to tell if it has been satisfied. Consequently, the requirement must be verifiable in order to know (prove) that the obligation has been met. In this context, "verifiable" refers to the ability of an organization to prove that the design and built system meets the requirement.

Based on the above analysis, the characteristics of a well-formed requirement are therefore derived to be: necessary, singular, conforming, appropriate, correct, unambiguous, complete, feasible, and verifiable.

5.6 Proposed Definitions of Attributes of a Requirement

Earlier, a requirement expression was defined as comprising a requirement statement with a set of associated attributes. We propose the following definition for a requirement attribute (which conforms to existing usage):

> An *attribute* is additional information included with a requirement statement, which is used to aid in the management of that requirement.

There is not sufficient space to discuss fully here a taxonomy for attributes (of which there may be a large number), nor for how such attributes could be defined and used. (For a more detailed discussion on attributes see the INCOSE Guide for Writing Requirements [46]) Here, however, based on earlier discussion, we note that attributes can be organized within four broad categories [49]:

- *Attributes to help define the requirement and its intent.* Examples include: rationale, system of interest primary verification method, system of interest verification approach, parent requirement, source, condition of use, and states and modes.
- *Attributes associated with the system of interest verification.* Examples include: system of interest verification level, system of interest verification phase, system of interest verification results, and system of interest verification status.
- *Attributes to help maintain the requirements.* Examples include: unique identifier, unique name, originator/author, date requirement entered, owner, stakeholders, change board, change status, version number, approval date, date of last change, stability, responsible person, requirements verification status, requirement validation status, status (of requirement), status (of implementation), trace to interface definition, trace to peer requirements, priority, criticality, risk, key driving requirement (KDR), additional comments, type/category (including functional/performance, interactions with external systems - input, output, external interfaces, environmental, facility, ergonomic, compatibility with existing systems, logistics, users, training, installation, transportation, storage; the -ilities (quality), reliability, availability, maintainability, accessibility, safety, security, transportability, quality provisions, growth capacity; physical characteristics; standards and regulations—policy and regulatory, constraint, business rule, business requirement)
- *Attributes to show applicability and allow reuse.* Examples include: applicability, region, country; state/province, application, market segment, business unit, business line.

5.7 Proposed Definitions for a Set of Requirements

Although each individual requirement expression is important, it is ultimately the set of requirements that will describe the complete entity in question and will be agreed-to as a contractual obligation between the acquirer and supplier—that description is contained in a requirement specification or requirement document, which can be defined as:

> A *set of requirements* is a structured set of agreed-to requirement expressions documented in an Entity (Enterprise/Business Unit/System/System Element/Process) Requirements Specification (Document).

If a requirement statement results from a formal transformation of one or more needs into an agreed-to obligation for an entity responsible for implementing the requirements, a set of requirements results from the formal transformation of the set of needs for the entity to be developed that represents an agreed-to obligation for the entity responsible for implementing the requirements. Again, the key elements of that definition are: a "formal transformation" and an "agreed-to obligation". Each of those two elements can therefore be elaborated further to develop specific characteristics of a well-formed set of requirements:

- Formal Transformation. Given the set of requirements is the result of a formal transformation, the following characteristics of the requirement set can be derived:
  - *Consistent*. For the transformation to be formal, individual requirements must not conflict with other requirements in the set.
  - *Complete*. If the formal transformation from one or more needs into a requirement results in individual requirements that are necessary, the set of requirements must be a sufficient solution to the set of needs—that is, all requirements have been included to meet the needs, and that all irrelevant requirements have been excluded.

- Agreed-to Obligation. If the set of requirements is to be the basis of a fair agreement to meet an obligation, the following characteristics of the set can be derived:
  - *Comprehensible*. An agreement is difficult to enact unless both parties are clear on the exact obligation—the set of requirements must explain itself so that the reader can 'visualize' the entity it describes and its relation to the higher-level entity of which it is a part.
  - *Feasible*. Just as there is little point in agreeing to an obligation for an individual requirement that is not feasible, the set of requirements must be achievable within the appropriate project constraints including cost and schedule, as well as other constraints including legal and ethical. The feasibility of the set cannot be determined by simply determining the feasibility of each of the requirements, since it is possible for a collection of feasible individual requirements to be infeasible as a set.
  - *Able to be validated*. The transformation must be formal and able to be validated, not just for individual requirements, but also for the set. It must be able to be shown that achievement of the set of requirements will result in meeting the set of needs from which it was transformed.

From the above analysis, the characteristics of a well-formed set of requirements can therefore be derived to be: consistent, complete, comprehensible, feasible, and able to be validated.

5.8 Summary

Figure 2 summarizes the relationships among the requirements engineering terms developed in this work.

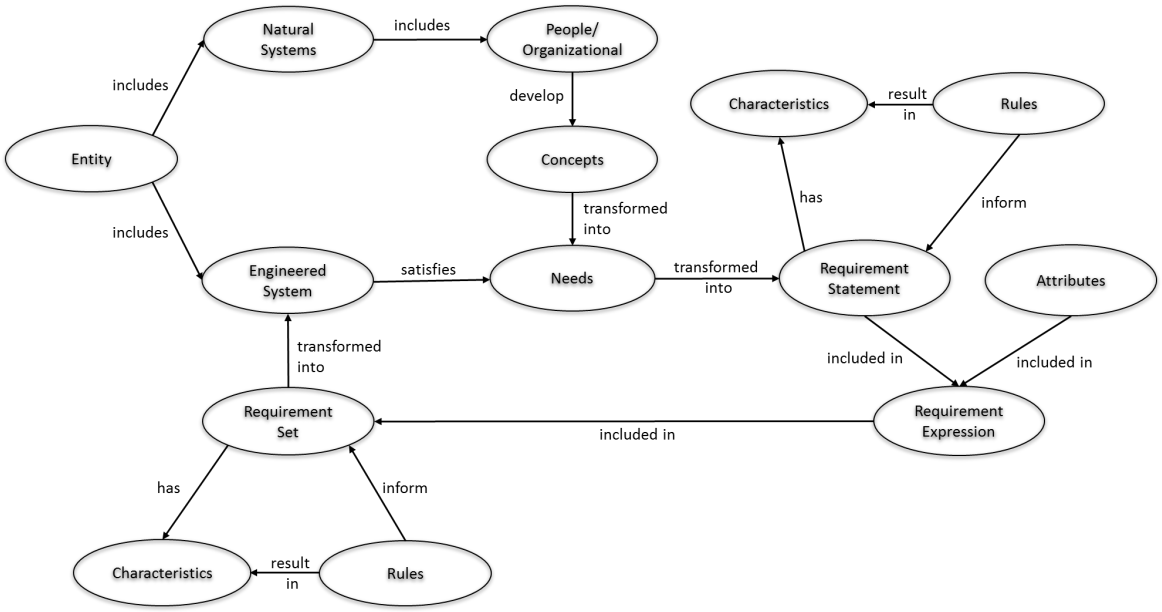

Figure 2. Entity-relationship diagram for requirements engineering terms.

6. CONCLUSIONS

Writing requirements is not simply an exercise in grammar; rather it is fundamentally an exercise in engineering. The purpose of a requirement expression is to communicate clearly the needs of various entities into a formal language such that the intent is clearly understood by those whose job it is to implement the requirement, those responsible for proving the built system meets the requirement, and those responsible for proving the resulting system meets the needs of the relevant entity.

A framework for the formal transformation of needs into requirements is essential but is not provided by the existing definitions presented in the major relevant standards for the terms entity, need, requirement expression, requirement statement, characteristics of a well-formed requirement statement and set of statements, and requirement attributes. Those definitions that are provided in standards and major sources are most commonly presented as assertions, with little accompanying intellectual underpinning, often at a middle-to-low level in the hierarchy of terms. The result is therefore a disparate and inconsistent body of knowledge that is not as useful as it needs to be.

This paper has demonstrated that a more rigorous approach to the definition of terms results in an integrated taxonomy that provides a much better basis for the important activity of requirements engineering. Specifically, the definition of a requirement statement includes action words, ‘transformation’ and ‘agreement’, that indicate the processes that result in a well-formulated requirement statement, requirement expression, and requirement set.

9. REFERENCES

[1] Ryan MJ (2013) An improved taxonomy for major needs and requirements artefacts. INCOSE International Symposium IS2013.

[2] Dick J, Chard J (2004) The systems engineering sandwich: combining requirements, models and design. INCOSE International Symposium IS2004.

[3] Hull E, Jackson K, Dick J (2011), Requirements engineering. Springer, London.

[4] ISO/IEC 29148 (2011) ISO/IEC 29148 FDIS Systems and software engineering—life cycle processes—requirements engineering.

[5] ISO/IEC 15288 (2015) ISO/IEC 15288:2015 Systems engineering—system life cycle processes.

[6] Robertson S, Robertson J (2013) Mastering the requirements process. Addison-Wesley, Harlow England.

[7] Gottesdiener E (2002) Requirements by collaboration: workshops for defining needs. Addison Wesley, Reading, MA.

[8] Davis AM (1993) Software requirements: objects functions and states. Prentice Hall International, Engelwood Cliffs NJ.

[9] Ferdinandi PL (2001) A requirements pattern: succeeding in the internet economy. Addison-Wesley Boston MA.

[10] Young R (2004) The requirements engineering handbook. Artech House, Norwood MA.

[11] Larson E, Larson R (2009) Practitioner's guide to requirements management part 1: requirements planning. Watermark Learning, Minneapolis MN.

[12] Alexander IF, Beus-Dukic L (2007) Discovering requirements: how to specify products and services. John Wiley & Sons, Chichester England.

[13] Laplante PA (2009) Requirements engineering for software and systems. CRC Press, Boca Raton FL.

[14] Adzic G (2011) Specification by example. Manning Publications Co, Shelter Island NY.

[15] Pohl K (2010) Requirements engineering: fundamentals principles and techniques. Springer-Verlag, Heidelberg.

[16] Leffingwell D, Widrig D (2000) Managing software requirements. a unified approach. Addison-Wesley, Reading MA.

[17] Wieringa RJ (1996) Requirements engineering. John Wiley & Sons, Chichester.

[18] Project Management Institute (PMI) (2013) A guide to the project management body of knowledge (PMBOK® Guide). PMI, Pennsylvania PA.

[19] MIL-STD-499B (1994) MIL-STD-499B Military Standard—Systems engineering—draft. United States of America Department of Defense, Washington DC.

[20] IEEE Computer Society (2005) IEEE-STD-1220-2005 IEEE standard for application and management of the systems engineering process. IEEE Computer Society, New York.

[21] ANSI/EIA-632 (1999) ANSI/EIA-632-1998 EIA Standard—processes for engineering a system. Electronic Industries Association, Arlington VA.

[22] IEEE Computer Society (1990) IEEE-STD-61012-1990 IEEE Standard Glossary of software engineering terminology. IEEE Computer Society, New York.

[23] International Institute for Business Analysis (IIBA) (2009) A guide to the business analysis body of knowledge® (BABOK® Guide) Version 2. IIBA.

[24] NASA (2013) NASA NPR 71231B NASA Systems engineering processes and requirements. National Aeronautics and Space Administration (NASA).

[25] Dorfman M, Thayer RH (1990) Standards guidelines and examples of system and software requirements engineering. IEEE Computer Society Press

[26] Sommerville I, Sawyer P (1997) Requirements engineering. John Wiley & Sons, Chichester England.

[27] Kotonya G, Sommerville I (2000) Requirements engineering: processes and techniques. John Wiley & Sons, Chichester England.

[28] Wiley W (2000) Essential system requirements: a practical guide to event-driven methods. Addison-Wesley, Reading MA.

[29] Sage AP, Armstrong JE (2000) Introduction to systems engineering. John Wiley and Sons, New York.

[30] Bray IK (2002) An introduction to requirements engineering. Addison Wesley, Boston MA.

[31] Young R (2006) Project requirements: a guide to best practices. Management Concepts, Vienna VA.

[32] Walden DD, Roedler GJ, Forsberg KJ, Hamelin RD, and Shorthill TM (eds) (2015) Systems engineering handbook: a guide for system life cycle processes and activities. International Council on Systems Engineering, San Diego CA.

[33] Andriole S (1996) Managing systems requirements: methods tools and cases. McGraw-Hill, New York.

[34] IEEE-SA Standards Board (1998) IEEE-STD 830-1998 IEEE Recommended practice for software requirements specifications. IEEE-SA Standards Board, New York.

[35] Young R (2001) Effective requirements practices. Addison-Wesley, Boston MA.

[36] Alexander IF, Stevens R (2002) Writing better requirements. Addison Wesley, London.

[37] Aurum A, Wohlin C (eds) (2005) Engineering and managing software requirements. Springer.

[38] Gilb T (2005) Competitive engineering: a handbook for systems engineering requirements engineering and software engineering using planguage. Butterworth-Heinemann.

[39] Hass KB, Wessels DJ, Brennan K (2008) Getting it right: business requirement analysis tools and techniques. Management Concepts, Vienna VA.

[40] van Lamsweerde A (2009) Requirements engineering: from system goals to UML models to software specifications. John Wiley & Sons, Chichester England.

[41] Hooks IF, Farry KF (2001) Customer-centered products – creating successful products through smart requirements management. AMACOM.

[42] Loucopoulos P, Karakostas V (1995) System requirements engineering. McGraw-Hill, New York.

[43] Berenbach B, Paulish DJ, Kazmeier J, Rudorfer (2009) A software and systems requirements engineering in practice. McGraw-Hill, New York.

[44] Wiegers KE (2003) Software requirements. Microsoft Press, Redmond WA.

[45] Goldsmith RF (2004) Discovering real business requirements for software project success. Artech House, Norwood MA.

[46] INCOSE (2012) INCOSE-TP-2010-006-01 Guide for writing requirements.

[47] Boehm BW (1984) Verifying and validating software requirements and design specifications. IEEE Softw 1: 75–88

[48] Pohl K, Rupp C (2011) Requirements engineering fundamentals. Rocky Nook, Santa Barbara CA.

[49] Wheatcraft LS, Ryan MJ, Dick J (2014) On the use of attributes to manage requirements. Volume 19, Issue 5, September 2016, pp. 448–458.

[50] Sommerville I (2016) Software engineering. Pearson, Boston MA.

[51] Kossmann M (2013) Requirements management: How to ensure you achieve what you need from your projects. Gower, Farnham UK.

[52] Harwell R, Aslaksen E, Hooks I., Mengot R, Ptack K (1993) What is a requirement? INCOSE International Symposium. Crystal City VA.

[53] Hooks I (1994) Writing Good requirements. INCOSE International Symposium. San Jose CA.

[54] Kar P and Bailey M (1996) Characteristics of Good Requirements. INCOSE International Symposium. Boston MA.

[55] Carson R, Aslaksen E, Gonzales R (2004) Requirements completeness. INCOSE International Symposium. Toulouse France.

[56] Carson R (2015) Implementing Structured requirements to improve requirements quality. INCOSE International Symposium. Seattle WA.

[57] Project Management Institute (PMI) (2016) Requirements management: a practice guide. PMI, Pennsylvania PA.

ACKNOWLEDGEMENT

The authors are co-chairs of the Requirements Working Group (RWG) of the International Council on Systems Engineering (INCOSE) and wish to acknowledge the review of the various RWG members who attended the working group sessions at the INCOSE International Workshop in January 2014.

BIOGRAPHIES

Dr Mike Ryan is the Director of the Capability Systems Centre, University of New South Wales, Canberra, at the Australian Defence Force Academy. He holds Bachelor, Masters, and Doctor of Philosophy degrees in electrical engineering, as well as a Graduate Diploma in Management Studies. He lectures and regularly consults in a range of subjects including communications and information systems, systems engineering, requirements engineering, and project management. He is the conference chair of two annual international conferences, he is the editor-in-chief of the Journal of Battlefield Technology, and is chair of the Requirements Working Group in the International Council on Systems Engineering (INCOSE). He is the author or co-author of eleven books, three book chapters, and over 170 technical papers and reports.

Lou Wheatcraft is a senior instructor/consultant for Requirements Experts (RE) who educates organizations on the importance of writing good requirements and helps them implement Requirement Development and Management (RD&M) processes based on industry best practices. Lou has taught over 180 requirement seminars over the last 16 years. Lou works with both government and industry clients to tailor training for their organizations and provides just in time team training for specific projects. Lou has spoken at Project Management Institute (PMI) Chapter meetings, International Council of System Engineering (INCOSE) conferences and NASA's PM Challenge and delivered tutorials to PMI and INCOSE chapters at multiple locations. Lou has had published and presented a multitude of papers on requirement RD&M topics for NASA's PM Challenge, INCOSE, INCOSE INSIGHT Magazine, and Crosstalk Magazine. Lou is a member of INCOSE, co-chair of the INCOSE Requirements Working Group, a member of PMI, the Software Engineering Institute (SEI), the World Futures Society, and the National Honor Society of Pi Alpha Alpha. Lou has a BS degree in Electrical Engineering from Oklahoma State University, an MA degree in Computer Information Systems from the University of Houston – Clear Lake, an MS degree in Environmental Management from the University of Houston – Clear Lake, and has completed the course work for an MS degree in Studies of the Future from the University of Houston – Clear Lake. Lou is the primary contributor to RE's blog on requirements best practices. The blog can be assessed at: http://www.reqexperts.com/blog .