On the Need for a System Need Statement

Michael Ryan

THE UNIVERSITY OF
NEW SOUTH WALES

The System Need Statement

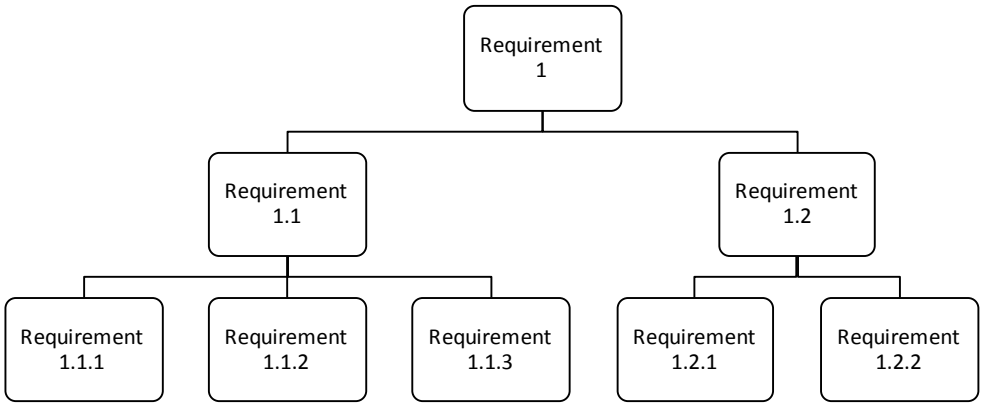
- Despite universal agreement that a system need statement (or problem statement) is an essential start point for system development, there strangely little guidance on an appropriate form, nor on any methods for development.
- Based on the common English definition of a statement as an account of facts, it would seem that any form of statement (from a sentence to several pages) is acceptable and that its contents are self-evident in that no particular process is required to guide its creation.
- The aim of this paper is to propose suitable attributes and an appropriate form for the system need statement, and a process for its development.

The Need for Good Problem Definition

- The need for good problem definition is not a new notion:
  - '*A problem is half-solved if properly stated*.'(John Dewey);
  - '*Every problem has in it the seeds of its own solution.*' (Norman Vincent Peale).
  - or, as put in the negative by Robert Mager, '*…if you're not sure where you're going, you're liable to end up someplace else*'.
- Poor definition of the requirements for a system will invariably result in the development of a poor system.
- Requirements definition is therefore critical.

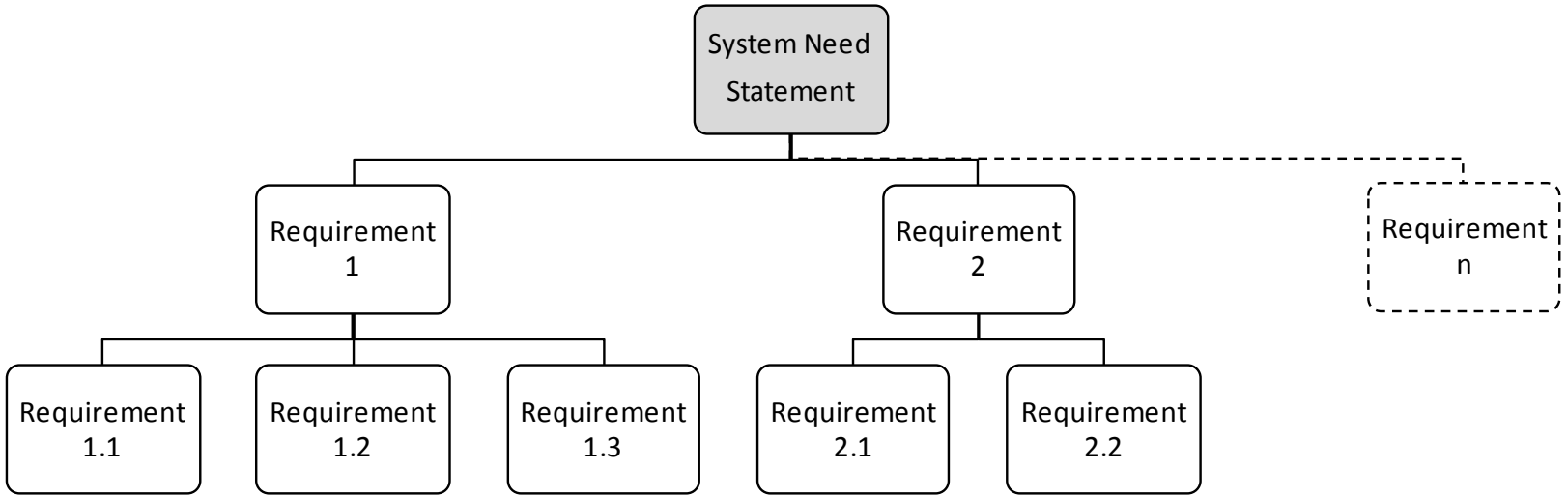
Hierarchical System Representation

- For well-behaved systems, requirements in System and Subsystem Specifications are normally grouped in a hierarchical fashion:
  - Through *forward traceability*, design decisions can be traced from any given system-level requirement (a parent requirement) down to a detailed design decision (a child requirement).
  - Similarly, through *backward traceability*, any individual design decision (any child) must be able to be justified by being associated with at least one higher-level requirement (a parent).



Ultimate System Requirement

- Backwards traceability from children to parent requirements leads to a useful observation that there is a single parent statement which serves as the highest-level requirement statement back to which all subsequent requirement statements should be able to be traced.



Desirable Attributes of a System Need Statement

- To extend the hereditary relationship, it follows that the parent need statement should have the same desirable attributes as those requirement statements that are its children.
- At the system level, requirements should be *necessary*, *unique*, *singular*, *complete*, *correct*, *unambiguous*, *feasible*, *independent of the method of implementation*, *justifiable*, and *verifiable* (IEEE-STD-1233).
- The ultimate antecedent statement—the system need statement—should therefore have the same attributes.

Desirable Attributes of a System Need Statement

- The attributes of unique, necessary, complete, correct, feasible, and verifiable are a related to the content of the system need statement and must be considered carefully during its drafting.
- The attributes of singular, unambiguous, independent of implementation, and justifiable are similarly a matter of the content, but are also assisted by a focus on an appropriate format for the system need statement.
  - A single sentence.
  - No conjunctions.
  - Contain no more than 5−7 concepts. .
  - Include an ‘in order to’ clause.
  - Avoid physical terms.

A Single Sentence

- To ensure that the system need statement is singular, it must be able to be expressed in a single sentence.
- In good written English, a sentence may be defined as encapsulating a single thought—if we want to express two thoughts, a second sentence is normally required.
- The discipline of writing a single sentence to describe the system need therefore keeps us at the level of abstraction that defines the system as a ‘single thought’, which is the level of abstraction we are seeking at this stage in the project in order to define a unique endeavour.

No Conjunctions

- The system need statement should not contain any conjunctions if it is to be singular.
- The statement cannot state, for example, that the purpose of the system is to ‘do this… and … do that…’.
- The presence of the ‘and’ in the statement implies that there are two purposes, and therefore two systems.
- Having said that, we can of course have conjunctions in phrases and adjectival clauses.
- As a general rule, however, we should not have any conjunctions because, if elements of the same type are joined by ‘and’, there is a group noun or adjective that can replace them.

No more than 5-7 Concepts

- The system need statement cannot contain any more than five-to-seven key elements if it is to remain within the intuition of the reader (Miller's Rule) and to remain unambiguous.

Include an ‘in order to’ Clause

- The system justification, or rationale, can be made explicit by the inclusion of an “… in order to…” clause at the end of the system need statement.
- This clause then ties the system need statement back to the business case for the system.

Avoid Physical terms

- To ensure that the subsequent system need statement is independent of the physical implementation means, the elements of the need should always be couched in functional, not physical, terms.
- The system need statement must not imply any particular physical solution.

A Process to Guide Statement Development

- The development of the system need statement has four main activities :
  - Identify candidate elements of the system need statement by simply listing them.
  - Iterate the set of system need statement elements by a process of review and test (determine whether every element should be included, and what other elements might be included).
  - Form the selected elements into a single concise system need statement.
  - Confirm the system need statement with stakeholders.

Identify Candidate Elements

- Crafting a complete balanced need statement in one (top-down) pass is very difficult for most stakeholders, particularly if the system is large, complicated, has a number of interfaces, and/or is a new system (contains elements with which they have no direct experience).
- It is therefore often productive to ask stakeholders to simply list (in a bottom-up manner) the key words or phrases that are candidates to be considered for inclusion in the need statement.
- Once each of these candidate need elements has been assessed for suitability, these are grouped and re-grouped into five-to-seven elements at a consistent level of abstraction.

Identify Candidate Elements

- The identification of candidate need elements can be obtained through a workshop with key stakeholders, through individual interviews with those same parties, or through any other appropriate requirements engineering technique.
- Care must be taken to ensure that each of the elements of the system need statement has the desirable attributes of being unique, necessary, complete, correct, feasible, and verifiable.

Identify Candidate Elements

- Note that when designers and stakeholders are considering these high-level elements, they are in fact articulating the major requirements of the system.
- The inclusion or exclusion of an element will significantly change the design, but that effect can be considered at a useful level of abstraction that is within the intuition of the stakeholders and their current level of understanding.

Identify Candidate Elements

- For example, when developing a need statement for a domestic burglar alarm, stakeholders might list such candidate elements as:
  - Alarm properties: flexible, reliable, sustainable, easy to use, and affordable.
  - Alarm functions: deterrence, detection, classification, and reporting of unauthorised entry.
  - Rationale: to alert the resident that the security of the residence has been compromised.

Iterate (Review and Test) Elements

- Having made a start, we iterate our understanding and converge to an agreed need through a process of review and test.
- Briefly, we have two main tests—whether every need element we have identified should be included; and whether other important elements have not been included.

"Would the resultant system be significantly different if this element of the need was included or omitted?"

Iterate (Review and Test) Elements

- In our domestic burglar alarm example there will be a need element that refers to ‘residents’.
- If the system is to be used by users of all ages, ethnic backgrounds, languages, abilities, and so on, then the need element ‘resident’ should not be defined any further so that lower-level designers infer from the need that all the system should be able to accommodate every nature of user.
- If however, support is only required for adult, English-speaking users, then the need element should be reworded to make the nature of resident explicit, because the resultant system will be identifiably different as a result.

Iterate (Review and Test) Elements

- Functional grouping should also be considered at this stage. Since the highest level of abstraction is sought in the need statement, stakeholders should identify when need elements can be combined.
- This aggregation will also assist in reducing the length of the sentence. The content of the functional groups should not be discarded, however, as those terms will no doubt be explicit in the subsequent decomposition.

Iterate (Review and Test) Elements

- For example, the list of burglar alarm properties (flexible, reliable, sustainable, easy to use and affordable) could be grouped into ‘market leading’, if that term was useful and meaningful in the system context.
- On the other hand, it may be considered that the grouping of ‘deterrence, detection, classification, and reporting of unauthorised entry’ into ‘preventing unauthorised entry’ is not appropriate, since ‘preventing’ does not capture the meaning contained in the individual elements of ‘deterrence, detection, classification, and reporting’.

Iterate (Review and Test) Elements

- Note that iteration is very productive. In a facilitated workshop, for example, participants could be asked to diverge and suggest candidate elements by listing them, and then be guided to convergence on an agreed set of elements through the positive and negative tests outlined above. Van Gundy (1992) suggests that this divergence/ convergence is part of a creative way to address problems in a structured manner.
- Continual iteration is required because stakeholders should not at all be confident that they have captured every element of the need since they cannot assume that the original statement of a problem is necessarily the best, or even the right one (Maier & Rechtin, 2000).

Form Need Statement

- Once the major elements of the need have been identified, stakeholders pause with that level of understanding and form the elements into a single concise sentence—the system need statement.
- With regard to purpose, the need statement should always include a short "… in order to …" clause at the end of the need statement describing why the system is to exist.
- This is an important step because this portion of the need statement is in fact a system-level rationale, which begins the good requirements-engineering practice of recording a rationale for each functional requirement.

Form Need Statement

- For example, a draft need statement for our domestic burglar alarm might be:

'To provide a market-leading domestic alarm system that can deter, detect, classify, and report unauthorised entry to a residence in order that the resident is aware of the residence's state of security'.

Form Need Statement

- There is often tension at this point between the need to be concise and the desire to include in the need statement every aspect of the design that is important (at the appropriate level of abstraction)—the statement should be as specific as possible so that the detail is not missed.
- One solution to this dichotomy is to provide for selected key words a footnote in which the detail is described.
- Alternatively, and probably most usefully, a project glossary can be developed, in which the term in question can be explicitly defined in the context of this particular project.

Form Need Statement

- For example, if there are to be a number of classes of user of the system, the need statement can simply refer to ‘users’ and the project glossary can be used to define the various classes of user in the required detail.
- In our burglar alarm example, there is great utility in defining the terms ‘resident’, ‘unauthorised entry’, and ‘residence’ in a glossary (as well as perhaps ‘deter’, ‘detect’, ‘classify’ and ‘report’).

‘To provide a market-leading domestic alarm system that can deter, detect, classify, and report unauthorised entry to a residence in order that the resident is aware of the residence’s state of security’.

Confirm Need Statement

- The need can be confirmed within the group of stakeholders that developed it or, more usefully, it can be communicated to a wider group and used to elicit comment that can inform subsequent revision.
- Because the need is effectively a single-sentence description of project scope, early communication facilitates a shared understanding of the system among all stakeholders.

Conclusion

- The system need statement is an essential start point for system development and should therefore be developed with more than a passing interest.
- To be useful as the antecedent for all subsequent system requirements, the system need statement should share the attributes of system requirements.These attributes are a related to the content of the statement and must be considered carefully during its drafting.
- The attainment of these attributes is greatly assisted if the system need statement is confined to a single sentence containing five-to-seven concepts (couched in functional terms with no conjunctions) and justified by an ‘in order to’ clause.