3A REQUIREMENTS AND SPECIFICATIONS

Abstract

Defining requirements to establish specifications is the first step in the development of an embedded system. However, in many situations, not enough care is taken in establishing correct requirements upfront. This causes problems when ambiguities in requirements surface later in the life cycle, and more time and money is spent on fixing these ambiguities. Therefore, requirements must be established in a systematic way to ensure their accuracy and completeness, but this is not always an easy task. This difficulty in establishing good requirements often makes it more of an art than a science. The difficulty arises from the fact that establishing requirements is a tough abstraction problem and often the implementation gets mixed with the requirements. In addition, it requires people with both communication and technical skills. As requirements are often weak about what a system should not do, this poses potential problems in the development of dependable systems, where these requirements are necessary to ensure that the system does not enter an undefined state. The development of dependable embedded systems requires even more complicated requirements as the embedded system not only interacts with the software but also with the outside world. Therefore, the importance of establishing good requirements is even greater in embedded systems design.

Introduction

Requirements and specifications are very important components in the development of any embedded system. Requirements analysis is the first step in the system design process, where a user's requirements should be clarified and documented to generate the corresponding specifications. While it is a common tendency for designers to be anxious about starting the design and implementation, discussing requirements with the customer is vital in the construction of safety-critical systems. Activities in this first stage have a significant impact on the downstream results in the system life cycle. For example, errors developed during the requirements and specifications stage may lead to errors in the design stage. When this error is discovered, the engineers must revisit the requirements and specifications to fix the problem. This leads not only to more time wasted but also the possibility of other requirements and specifications errors. Many accidents are traced to requirements flaws, incomplete implementation of specifications, or wrong assumptions about the requirements. While these problems may be acceptable in non-safety-critical systems, safety-critical systems cannot tolerate errors due to requirements and specifications. Therefore, the requirements must be specified correctly to generate clear and accurate specifications.

There is a distinct difference between requirements and specifications. A requirement is a condition needed by a user to solve a problem or achieve an objective. A specification is a document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system, and often, the procedures for determining whether these provisions have been satisfied. For example, a requirement for a car could be that the maximum speed is at least 120mph. The specification for this requirement would include technical information about specific design aspects. Another term that is commonly seen in books and papers is requirements specification which is a document that specifies the requirements for a system or component. It includes functional requirements, performance requirements, interface requirements, design requirements, and development standards. So the requirements specification is simply the requirements written down on paper.

Establishing Correct Requirements

The first step toward developing accurate and complete specifications is to establish correct requirements. As easy as this sounds, establishing correct requirements is extremely difficult and is more of an art than a science. There are different steps one can take toward establishing correct requirements. Although some of the suggestions sound fairly obvious, actually putting them into practice may not be as easy as it sounds. The first step is to negotiate a common understanding. There is a quote by John von Neumann that states "There's no sense being exact about something if you don't even know what you're talking about." [Gause89] Communication between the designer and customer is vital. There is no point in trying to establish exact specifications if the designers and customers cannot even agree on what the requirements are.

The problem stems from ambiguities in state requirements. For example, say the requirement states that we want to create a means that would transport a group of people from Boston to Washington D.C. Possible interpretations of this requirement include building a bus, train, or airplane, among other possibilities. Although each of these transportation devices satisfies the requirement, they are certainly very different. Ambiguous requirements can be caused by missing requirements, ambiguous words, or introduced elements. The above requirement does not state how fast the people should be transported from Boston to Washington D.C. Taking an airplane would certainly be faster than riding a bus or train. These are also missing requirements. "a group of people" in the above requirement is an example of ambiguous words. What exactly does "group" imply? A group can consist of 5 people, 100 people, 1000 people, etc. The requirement states to "create a means" and not "design a transportation device". This is an example of introduced elements where an incorrect meaning slipped into the discussion. It is important to eliminate or at least reduce ambiguities as early as possible because the cost of them increases as we progress in the development life cycle.

Often the problem one has in establishing correct requirements is how to get started. One of the most important things in getting started is to ask questions. Context-free questions are high-level questions that are posed early in a project to obtain information about the global properties of the design problem and potential solutions. Examples of context-free questions include who is the client? what is the reason for solving this problem? what environment is this product likely to encounter? and what is the trade-off between time and value?. These questions force both sides, designer, and customer, to look at the higher issues. Also, since these questions are appropriate for any project, they can be prepared in advance. Another important point is to get the right people involved. There is no point in discussing requirements if the appropriate people are not involved in the discussion. Related to getting the right people involved is making meetings work. Having effective meetings is not as easy as it sounds. However, since they play a central role in establishing requirements it is essential to know how to make meetings work. There are important points to keep in mind when creating effective meetings, which include creating a culture of safety for all participants, keeping the meeting to an appropriate size, and other points. [Gause89]

Exploring the possibilities is another important step toward generating correct requirements. Ideas are essential in establishing correct requirements, so it is important that people can get together and generate ideas. Every project will also encounter conflicts. Conflicts can occur from personality clashes, people that cannot get along, intergroup prejudice such as those between technical people and marketing people, and level differences. A facilitator must be present to help resolve conflicts.

In establishing requirements, it is important to specifically establish the functions, attributes, constraints, preferences, and expectations of the product. Usually, in the process of gaining information, functions are the first ones to be defined. Functions describe what the product is going to accomplish. It is also important to determine the attributes of a product. Attributes are characteristics desired by the client, and while 2 products can have similar functions, they can have completely different attributes. After all the attributes have been clarified and attached to functions, we must determine the constraints on each of the attributes. Preferences, which is desirable but optional condition placed on an attribute, can also be defined in addition to its constraints. Finally, we must determine what the client's expectations are. This will largely determine the success of the product.

Testing is the final step on the road to establishing correct requirements. There are several testing methods used, as listed below. [Gause89]

- Ambiguity poll - Used to estimate the ambiguity in a requirement. This involves asking questions such as how fast? how big? how expensive? and then determining if there is ambiguity between the high and low values.
- Technical review - A testing tool for indicating the progress of the requirements work. It can be formal or informal and generally only deals with technical issues. Technical reviews are necessary because it is not possible to produce error-free requirements and usually it is difficult for the producers to see their own mistakes.
- User satisfaction test - A test used regularly to determine if a customer will be satisfied with a product.
- Black box test cases - Constructed primarily to test the completeness, accuracy, clarity, and conciseness of the requirements.
- Existing products - Useful in determining the desirable and undesirable characteristics of a new product.

At some point it is necessary to end the requirements process as the fear of ending can lead to an endless cycle. This does not mean that it is impossible to revisit the requirements at a later point in the development life cycle if necessary. However, it is important to end the process when all the requirements have been determined, otherwise, you will never proceed to the design cycle.

Establishing good requirements requires people with both technical and communication skills. Technical skills are required as the embedded system will be highly complex and may require knowledge from different engineering disciplines such as electrical engineering and mechanical engineering. Communication skills are necessary as there is a lot of exchange of information between the customer and the designer. Without either of these two skills, the requirements will be unclear or inaccurate.

Requirements in safety critical embedded systems must be clear, accurate, and complete. The problem with requirements is that they are often weak about what a system should not do. In a dependable system, it is just as important to specify what a system is not supposed to do as to specify what a system is supposed to do. These systems have an even greater urgency that the requirements are complete because they will only be dependable if we know exactly what a system will do in a certain state and the actions that it should not perform. Requirements with no ambiguities will also make the system more dependable. Extra requirements will usually be required in developing a dependable embedded system. For example, in developing a dependable system for non-computer-literate people, extra requirements should be specified to make the system safe even in exceptional or abusive situations.

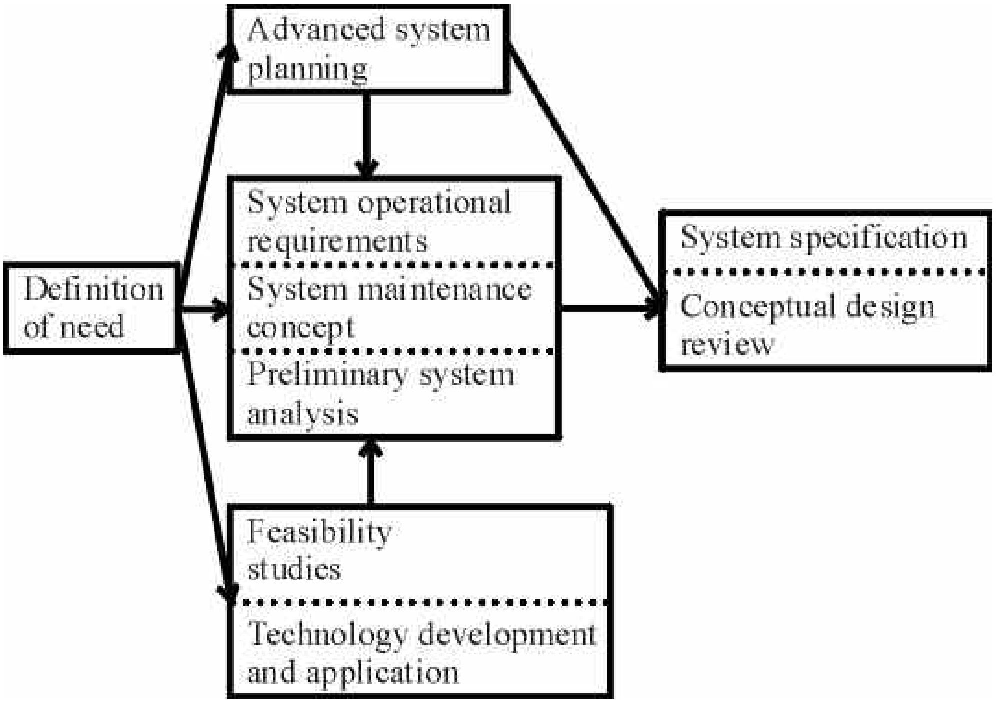
Requirements and Specification's Role in System Design

Systems exist everywhere in the universe we live in. The universe can be considered a system, and so can an atom. A system is very loosely defined and can be considered as any of the following definitions. [Blanchard90]

- A combination of elements forming a complex or unitary whole (i.e. river system or transportation system)
- A set of correlated members (i.e. system of currency)
- An ordered and comprehensive assemblage of facts, principles, or doctrines in a particular field of knowledge (i.e. system of philosophy)
- A coordinated body of methods, a complex scheme, or a plan of procedure (i.e. system of organization and management)
- Any regular or special method or plan of procedure (i.e. a system of marking)

The important characteristic of a system is that there is unity, functional relationship, and useful purpose. Systems engineering is not a technical specialty but is a process used in the evolution of systems from the point when a need is identified through production and construction to the deployment of the system for consumer use. [Blanchard90] Systems engineering requires knowledge from different engineering disciplines such as aeronautical engineering, civil engineering, and electrical engineering. The development of embedded systems also requires the knowledge of different engineering disciplines and can follow the techniques used for systems engineering. Therefore, it is appropriate that the steps used in establishing system requirements also apply to requirements for embedded systems.

The conceptual system design is the first stage in the systems design life cycle and an example of the systems definition requirements process is shown in Figure 1. Each box will be explained below.


Advanced system planning
Definition of need
System operational requirements
System maintenance concept
Preliminary system analysis
System specification
Conceptual design review
Feasibility studies
Technology development and application

In establishing system requirements, the first step is to define a need. This need is based on a want or desire. Usually, an individual or organization identifies a need for an item or function, and then a new or modified system is developed to fulfill the requirement. After a need is defined, feasibility studies should be conducted to evaluate various technical approaches that can be taken. The system operational requirements should also be defined. This includes the definition of system operating characteristics, maintenance support concept for the system, and identification of specific design criteria. In particular, the system operational requirements should include the following elements. [Blanchard90]

- Mission definition - Identification of the primary operating mission of the system in addition to alternative and secondary missions.
- Performance and physical parameters - Definition of the operating characteristics or functions of the system.
- Use requirements - Anticipation of the use of the system.
- Operational deployment or distribution - Identification of transportation and mobility requirements. Includes a quantity of equipment, personnel, etc., and geographical location.
- Operational life cycle - Anticipation of the time that the system will be in operational use.
- Effectiveness factors - Numbers specified for system requirements. Includes cost-system effectiveness, mean time between maintenance(MTBM), failure rate, maintenance downtime, etc.
- Environment - Definition of the environment in which the system is expected to operate.

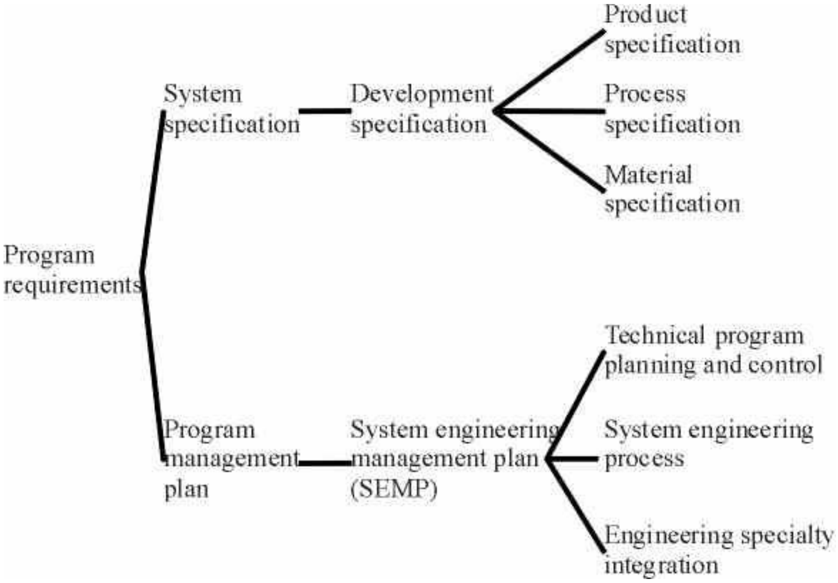
The system operational requirements define how the system will be used in the field by the customer.

Usually, in defining system requirements, the tendency is to cover areas that are related to performance as opposed to areas that are related to support. However, this means that emphasis is only placed on part of the system and not the whole system. It is essential to take into consideration the entire system when defining system requirements. The system maintenance concept describes the overall support environment that the product is supposed to exist in.

After the system operational requirements and system maintenance concept are defined, the preliminary system analysis is performed to determine which approach for system development should be adopted. The following process is usually applied. [Blanchard90]

1. Define the problem - The first step always begins with clarifying the objectives, defining the concerned issues, and limiting the problem so that it can be effectively studied.
2. Identify feasible alternatives - All the alternatives should be considered to make sure that the best approach is chosen.
3. Select the evaluation criteria - The criteria for the evaluation process can vary considerably, so the appropriate ones must be chosen.
4. Applying modeling techniques - A model or series of models should be used.
5. Generate input data - The requirements for appropriate input data should be specified.
6. Manipulate the model - After data is collected and inputted, the model may be used. Analysis after using the model will lead to a recommendation for some kind of action.

After the preliminary system analysis, advanced system planning will be done. Early system planning takes place from the identification of a need through the conceptual design phase. The results from these plans will be defined as either technical requirements included in the specifications or management requirements included in a program management plan. The documents associated with these requirements are shown in Figure 2. The system specification includes information from the operational requirements, maintenance concept, and feasibility analysis. The System Engineering Management Plan(SEMP) contains three sections. The technical program planning and control part describes the program tasks that have to be planned and developed to meet system engineering objectives such as work breakdown structure, organization, risk management, etc. The system engineering process part describes how the system engineering process applies to program requirements. Finally, the engineering specialty integration part describes the major system-level requirements in the engineering specialty areas such as reliability, maintainability, quality assurance, etc.


Program requirements
System specification
Development specification
Product specification
Process specification
Material specification
Program management plan
System engineering management plan (SEMP)
Technical program planning and control
System engineering process
Engineering specialty integration

Finally, the conceptual design review is also performed during the conceptual design stage. It usually occurs early in the system engineering development life cycle after the operational requirements and the maintenance concept have been defined.

Requirements Traceability

It is very important to verify that the requirements are correctly implemented in the design. This is done with requirements traceability which is usually referred to as "the ability to follow the life of a requirement, in both forwards and backward direction (i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases.)" [Ramesh95] Requirements traceability captures the relationships between the requirements, specifications, and design. Standards for systems development such as the one from the U. S. Department of Defense (standard 2167A) require that requirements traceability be used. Although requirements traceability has been around for more than 2 decades, there has been no consensus as to what kind of information should be used as part of a traceability scheme. The problem is that the definition of traceability differs when taken from different points of view of the system. (i.e. the view of the system is different for customers, project managers, test engineers, etc.) Each organization has a different purpose and methodology for requirements tracing. While it is not the purpose of this paper to dwell on a long discussion about requirements traceability, a short example of the methodology used at one organization will be given. [Ramesh95]

The projects typically involved at Abbott Laboratories Diagnostics Division are real-time, embedded, in vitro diagnostic instruments approaching 200,000 lines of code. They have found that traceability aids project managers in verification, cost reduction, accountability, and change management. Traceability helps in verifying that software requirements are satisfied in the design process and that they are tested for during the verification process. Traceability allows the allocation of product requirements early in the development cycle thereby reducing costs of correcting defects (due to untraceable components) in the integration and system test phase. Providing quantitative traceability analyses also allows for accountability in making sure that project milestones are approved, deliverables are verified, and customers are satisfied. The documentation from traceability also keeps information organized during changes in staff or management.

A specific in vitro diagnostic instrument contained approximately 175,000 lines of source code and approximately 1,600 software requirements that needed to be traced. While the division also has an automated traceability system (ATS) that allowed them to automate many of the tasks, it was the process and not the tool that led to their success. The main purpose of the traceability program is to identify links and determine that the links are complete and accurate. The traceability analysis consists of 4 aspects: forward requirements analysis, reverse requirements trace, forward test analysis, and reverse test trace. These steps are used to trace each software requirement through its design elements and test traces. The ATS can be used to design documentation matrices and test matrices that are used to perform the different analyses required. The ATS is also able to give feedback about the design components that are not yet implemented during the life cycle. In the test phase, the ATS gives input to what requirements are covered by the test cases. [Watkins94]

Requirements Standards

There are many requirements and specification standards. They are mostly military standards as opposed to "commercial" standards. In addition, most of the standards are in the systems engineering area and particularly deal with the software aspects. A good reference to many of these standards is Standards, Guidelines, and Examples on System and Software Requirements Engineering from the IEEE Computer Society Press. [Dorfman90] This book is a compilation of international requirements standards and U. S. military standards. There is also a section on requirements analysis methodologies and examples. Listed below are several relevant standards, but the list is by no means exhaustive.

- IEEE Recommended Practice for Software Requirements Specifications (IEEE std 830-1998)

- British Standard Guide to Specifying User Requirements for a Computer-Based Standard (BS6719 - 1986)
- Canadian Standard, Basic Guidelines for the Structure of Documentation of System Design Information (Z242.15.4-1979)
- U. S. Military Standards
- System/Segment Specification (DI-CMAN-80008A, 2/29/1988)
- Software Requirements Specification (DI-MCCR-80025A, 2/29/1988)
- Interface Requirements Specification (DI-MCCR-80026A, 2/29/1998)