UDC 004.031.6

A. Parkhomenko, O. Gladkova
Zaporizhyzhya National Technical University,
Software Tools Department

COMPLEX REQUIREMENTS ANALYSIS FOR THE HIGH-LEVEL DESIGN OF EMBEDDED SYSTEMS

КОМПЛЕКСНИЙ АНАЛІЗ ВИМОГ ПРИ ВИСОКОРІВНЕВОМУ ПРОЕКТУВАННІ ВБУДОВАНИХ СИСТЕМ

© Parkhomenko A., Gladkova O., 2014

In the paper the requirements analysis that must be considered at designing embedded systems was performed. Describes peculiarities of the structuring requirements and developed a model of requirements to the created embedded system. Methodology of creating requirements, taking into account the processes of their definition and analysis was proposed. The results of application of the developed requirements model during project implementation the embedded system to control of mobile platform were presented.

Key words: embedded system, requirements classification, requirements analysis, requirements model, methodology of creating requirements, regulatory documentation, mobile platform.

У роботі виконано аналіз вимог, які необхідно враховувати під час проектування вбудованих систем. Описані особливості структурування вимог та розроблена модель вимог для вбудованих систем. Запропоновано методику створення вимог з урахуванням процесів їх визначення та аналізу. Наведено результати практичного застосування розробленої моделі вимог при реалізації проекту вбудованої системи управління рухомою платформою.

Ключові слова: вбудована система, класифікація вимог, аналіз вимог, модель вимог, методика створення вимог, нормативна документація, рухома платформа.

Introduction

Embedded Systems (ES) is one of the most complex design objects for computer technology developers [1]. Even a cursory analysis of typical requirements and constraints that must be considered when creating the ES, confirms this:

- minimal own power (possibly self-powered);
- minimal own size and weight;
- toughness and rigidity of the design;
- thermal control;
- radiative and electromagnetic resistance (possibly working capacity in vacuum);
- guaranteed time between failures;
- term availability solutions on the market, etc. [2]

Requirements definition, analyze and correct distribution between hardware and software components of the architecture at the stage of high-level design is a critical success factor in the implementation of the ES.

However, as experts note in the near future the existing significant gap between the requirements of the ES and the efficiency of the hardware and software design, the required amounts of verification and testing devices will only increase. Mainly, the problem lies in the insolvency of traditional approaches to the design of the ES amid of modern requirements for such systems. One of perspective directions in terms of the creation of modern design methodologies ES, is a complex accounting of the requirements specification and constraints within the overall project [1].

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Analysis of the requirements types for the ES

As described in [3], the system design includes the following technical processes work with requirements:

- stakeholder requirements definition process;
- requirements analysis process.

Stakeholder requirements definition process result is a list of all stakeholders and their system requirements, which are the basis for further analysis and identification of functional and non-functional requirements of the system.

Requirements analysis process result is a complete transformation holder (stakeholder) requirements in the technical vision of the developed system, i.e. a well-defined classification system requirements. Requirements analysis process subsequently affects the architectural design, as well as the means to implement it.

Today, problems work with the system requirements are discussed in [4,5,6]. Analysis of the standard [6] showed that the term "System Requirements" means the combinations of:

1) requirements for the system generally (business requirements, external interfaces);
2) requirements for the functions (tasks), which performs system (functional requirements, quality attributes);
3) requirements for the types of support (constraints).

The first type of requirements also includes description of the system structure, which includes a list of its components. For complicated projects including a plurality of components (subsystems), it is necessary to produce a placement of high-level requirements on these components [5].

The standard [7] gives a definition of an embedded system – a collection of hardware and software components designed to perform a specific function or set of functions. Since the software (SW) development is an essential part of designing the ES, necessary to adopt an integrated approach based on the processes of the system and software engineering, and to determine both the system (high-level requirements to the project) and the program requirements (i.e., requirements, distributed between software components of the project). The first step to the joint set of standards that describe systems and software life cycle processes is the international standard [4]. It establishes a strict connection between the system and applied it in software, interpreting software as part of the system, by allocating requirements to the software from the system requirements. The standard shows a difference between the analysis of system requirements and requirements analysis of the SW product. In the general case, the construction of the system architecture defines system requirements for the various components of the system. Analysis of the requirements to software predetermines the requirements for them, based on the system requirements, assigned to each program component.

Addition to the above processes work with the system requirements in the standard [3], also in the standard [4] has special processes of software intended to create a specific program element of the system, in particular the software requirements analyzing process. During this process, the requirements are determined by the software elements of the system and their interfaces, the impact of software requirements on the operating environment is installed, as well as compatibility and traceability between the software requirements and system requirements.

In [7] highlights the fact that the input to the process of determining the requirements for the software are the system requirements, the description of the hardware interface and system architecture (if they are not included in the system requirements). A goal of the design process software consists to develop software architecture and requirements of the lower level on the basis of top-level requirements for software. It follows that the definition of requirements for software development and the development of the architecture of the software only after the hardware description of the project and its architecture.

Leffingwell [8] discussed in detail the requirements management process and methods for developing software applications. Leffingwell requirements model consists of 3 clusters (levels) requirements:

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

- needs – users/stakeholders requirements;
- features – system provides services to meet the needs of stakeholders;
- software requirements – concretized software requirements.

Cluster "needs" relates to the field of the users problem (customers) understanding, for the further construction of the system that satisfies their needs. The following two clusters "function" and "software requirements" refer to solving the user problem.

As noted in [9], in a variety of software development methodologies, an approach based on the definition of requirements groups to the software are used. This approach typically comprises group (type, class) requirements, for example: system, program, functional, non-functional (e.g., attributes quality) etc.

A classic example of a high-level structuring requirement groups as requirements of the software product, is presented in the book K.E. Wiegers [5]. Requirements model proposed by Wiegers for designing software consists of three levels of requirements [5]:

- business – why should developing a system;
- user – describe the users of the system, as well as their work with it;
- functional – determine the functionality (behavior) of a software system.

Model envisages two types of requirements:

- functional – that the system must do;
- non-functional – what conditions must be met when working.

In this requirements model under the external interface understand "user interface", i.e. user interaction with the software. For the development of the ES in the standard [7] provides a more expanded interface definition – it is the interdependence between two or more objects that share and ensure information or exchange them. As such objects may be: software configuration element (SWCE) – is a set of software components. And hardware configuration element (HWCE) – is a set hardware. Accordingly, the interfaces may be the following variants: SWCE/SWCE, SWCE/HWCE, SWCE/user, HWCE/user or module HW/ module HW.

Thus, of the SW requirements model presented in [5,8] can be considered as a basis for developing a requirements model, taking into account the specific features and aspects of the design of ES. However, they must undergo extensive revision to take into account other types of requirements and work processes with the system requirements presented in the standards [4,6,7].

Purpose of this work is to develop complex model of the requirements for the ES, as well as methodology of creating requirements for efficiency high-level design.

Structure and features of requirements model for the ES

Based on the performed analysis, a requirements model for the design of ES was proposed. Schematic representation of which is shown in Fig. 1. Like the requirements model by Wiegers and by Leffingwell, has 3 levels, however, in contrast to the model Wiegers, does not use the concepts of requirements functional type and functional level. Since the ES represents set of hardware and software, the proposed scheme shows the distribution of levels in the model, on the requirements to hardware and software components of the system.

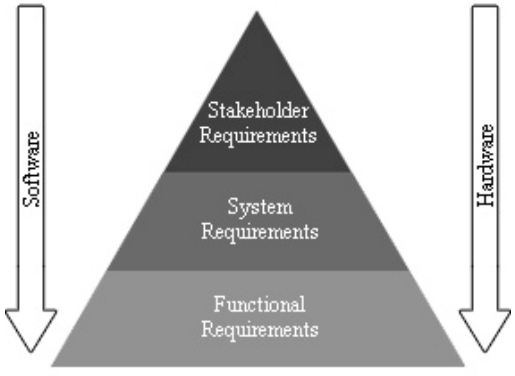

Fig. 1. Requirements model for the ES

In the proposed requirements model highlighted in the following levels:

- stakeholder requirements – include requirements all project stakeholders, such as customers and end users of the product. Can be represented by: business requirements, use cases, quality attributes;
- system – system requirements, comprising a combination of interconnected components (hardware, software, users, etc.). Can be represented by: system requirements (the whole system), constraints, interfaces, required operating conditions;

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

- functional – function (functional characteristics) distributed among software and hardware components of the system.

Table 1 shows the features of the proposed requirements model and the models Wiegers and Leffingwell. Can emphasize such distinctions proposed model from Wiegers:

1. On the upper level are combined business requirements and user requirements.
2. On the introduction of the second level are placed high-level requirements to developed embedded system and identifies the data requirements, interfaces, and system constraints.
3. On the lower level, considering features of the ES, decomposes the functional requirements on the software and hardware components of the system.

Table 1

Distinctive features of requirements models

Proposed model	By Wiegers	By Leffingwell
3 levels of requirements: – stakeholder requirements; – system requirements; – functional requirements.	3 levels of requirements: – business requirements; – user requirements; – functional.	3 levels of requirements: – stakeholder needs; – features; – software requirements
2 types of functional requirements: – hardware; – software.	2 types of requirements: – functional; – nonfunctional.	
Interface: user, SW/SW, SW/HW, SW/user, module HW/module HW	Interface: user, SW/SW, SW/user	Additional category: user interface

For requirements model which was developed provides a methodology of creation requirements for an embedded system (Fig. 2), which includes a sequence of steps: 1. Identify business requirements: identification of the product concept which we want to create. 2. Identification of the users, combining them into a groups and fixing their requirements. 3. Definition of use cases (scenarios) of the product by describing the user's workflow. 4. Develop use-case: fixation of use cases and a thorough study of the sequence of interaction between the system and external actors (example of an effective way of formulating and documenting requirements of this type is shown below). 5. Emphasizing quality attributes: identification of high-quality system performance characteristics of concrete actions. 6. Identify: system requirements, including the whole environment, constraints, requirements to the operating conditions, interfaces. 7. Check the requirement specifications that have been developed for errors, completeness, unambiguousness, consistent and others. Desirable that in checking specifications involving all stakeholders. After that follow the steps of determining the functional requirements for HW (8) and SW (9), which implies a more detailed identification and transformation of user requirements into a form useful for developers. Fixation of functional software requirements in the specification of software requirements, may be performed by presenting the expected behavior of the system in the form of an “event-reaction” (example shown below). This form is convenient for the further development of test cases. 10. Checking developed requirements specifications. 11. Development of test cases: description of the states and transitions between states, as well as the conditions under which transitions from state to state. 12. Checking developed specifications to system requirements, functional requirements and use cases for errors, completeness, inconsistency, etc.

Creating requirements is an iterative process. The first three stages are usually carried out once (but they need to be reviewed and revised), the remaining steps are repeated for each new version of the product.

One of the important stages controls of the created requirements is traceability (relationship). To provide traceability, each requirement must be unique and identified to be able to refer to it. Best to manage traceability, using requirements management tool. Can be identified such as requirements management tools: IBM Rational Requisite Pro, Borland CaliberRM, DOORS. For Rational Requisite Pro, you can point out a number of advantages:

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

- support Word to create project documentation;
- availability of templates for the preparation of project documentation;
- easy navigator between Microsoft Word and powerful infrastructure of the database;
- integration with the tools of object-oriented modeling;
- the possibility of tracing the relationship of different levels requirements by Traceability matrix.

Traceability matrix created in IBM Rational Requisite Pro allows the modification of some requirements easily determine which requirements affect this change and what should be checked [10]. Traceability created requirements in accordance with the scheme requirements model in Requisite Pro is as follows (Fig. 3).

On the upper level there are Stakeholder Requirements (ID of requirements this type – STRQ), a and the System Requirements (SYST), made of stakeholder needs.

Functional Requirements – it is functionality provided by the system, usually formulated by a business analyst, the appointment requirements – to meet the needs of the customer. Functional requirements are divided on the various subsystems of project on the basis the System Requirements: FEAT_HW – functional hardware requirements; FEAT_SW– functional software requirements.

Use Case (UC) – a description of the system behavior in terms of action sequences.

Supplementary Requirement (SUPL) – other requirements (usually non-functional), which can not be described Use Cases.

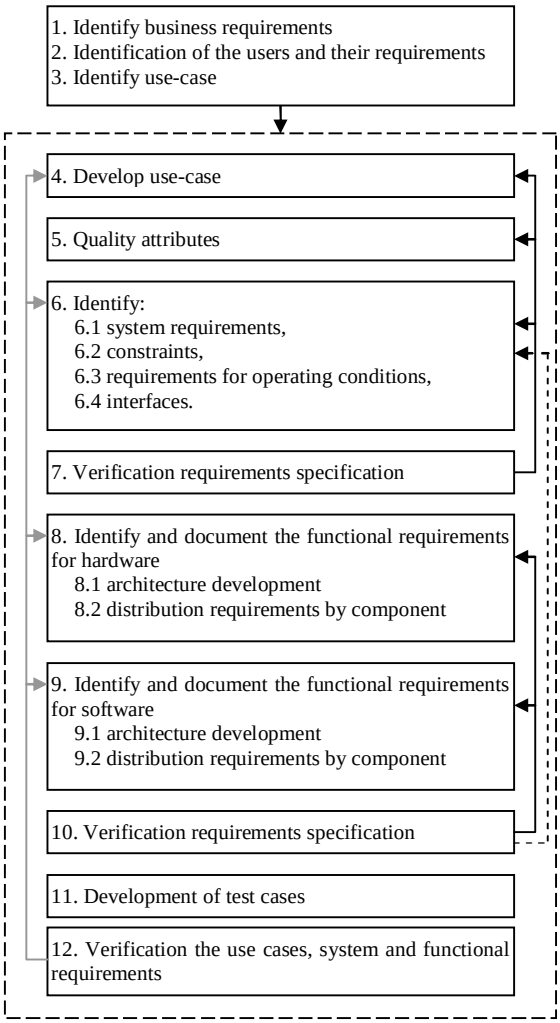

Fig. 2. Methodology of creating requirements

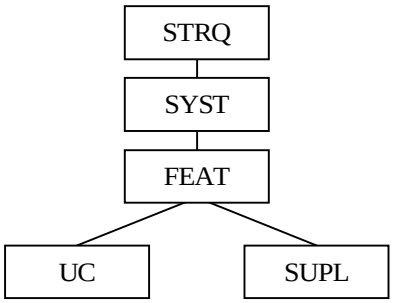

Fig. 3. Traceability requirements in Requisite Pro

The application of the model requirements

The proposed requirements model, as well as the process of creating requirements was practically applied in the implementation of the project to create the Embedded Control System of Mobile Platform. The following examples represent a traceability sequence of requirements relating to a particular part of the system.

Example 1: Stakeholder requirements.

STRQ7: Change the maximum permissible speed of the mobile platform, depending on the category of user (adults and children), as well as the level of management skill.

Example 2: Use-case.

UC3: “Switching speed of the mobile platform”.

Actors: Operator.

Output conditions: operator included main control unit (MCU) and made presetting race mobile platform.

Normal direction:

1. Operator changes the speed mode mobile platform.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

2. The system sends a command to the mobile platform of switching speed.

3. Movable platform receives the command and processes it.

4. Message is displayed on switching speed of the mobile platform in the selected mode.

An alternative direction: no

Exceptions: The command was not sent – system shows the error number on the display.

Example 3: System requirements

SYST4: The system should switch (increase or decrease) the maximum speed of radio-controlled mobile platform.

When switching speeds, the system must output messages on the display of the speed mode that has been selected.

Example 4: Constraints.

SUPL2: Having two modes of installation of radio-controlled mobile platform the maximum speed:

– Kids mode – 6 km/h.

– Normal mode – 20 km/h.

In this case, the requirement is partitioned into functional requirements for hardware and software.

Example 5: HW requirements.

FEAT_HW7: Engine mounted on a mobile platform should at least reach speeds of 20 km/h.

Functional requirement for the software is determined on the basis of the specified system requirements and hardware specifications. Since our project on the mobile platform was installed engine, maximum speed is 60 km/h, the functional requirements for the software will look like as it is written in the following example.

Example 6: SW requirements.

Event: The speed setting on the main control unit in Kids mode.

Response: The installation 10 % speed of maximum speed of the machine. On the display MCU is displayed inscription confirming that the rate of cars was changed “Speed Mode: Kids”.

Event: The speed setting on the central control unit in Normal mode.

Response: The installation 30 % speed of maximum speed of the machine. On the display MCU is displayed inscription confirming that the rate of cars was changed “Speed Mode: Normal”.

Also, examples of Stakeholder requirements for ES control of mobile platforms that is created are the following requirements:

- STRQ1: object (platform) to be remotely controlled;
- STRQ3: create the effect of the user presence in the mobile FPV_Platform;
- STRQ10: ensure the collection and storage of the system states for all time period.

Which were later formulated in the high-level system requirements:

- SYST3: transfer of data from the MCU to the object should be carried out using the radio channel at a distance of not more than 20 m;
- SYST4: transmit video via radio channel from a camera mounted on FPV_Platform for users video glasses;
- SYST12: organize the storage of data about the current state of development on the flash memory installed in the MCU (this requirement can be traced to the functional requirements for software: the program should save the received data to the MCU in txt format delimited).

For successful implementation of the project by the requirements management system IBM Rational RequisitePro was performed to develop the necessary project documentation:

- Glossary – all terms of the project;
- Vision – complete description of the system and system-level functions (system requirements);
- Use Case Specific – using scenarios and algorithms;
- Functional Specification – functional requirements;
- Supplementary Specification – functional requirements, not related to the using scenarios or nonfunctional (additional) requirements.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

According to the results the work with requirements using Requisite Pro, developed a set of documents including all requirements to ES of mobile objects which is developed. Once formulated customer requirements and system requirements, the system architecture has been successfully developed, analyzed of the requirements for HW and SW and SW architecture was developed.

Conclusions

Working with the requirements is very important when creating a project of embedded system and achieves the goals for its implementation. However, existing for today standards, as well as work in the creation and analysis of requirements describe only certain aspects of this problem and do not include all the features of embedded systems design. Therefore, in this paper, based on requirements models Wiegers and Leffingwell a requirements model for the ES has been developed. The model is realized 3 levels and 2 types of requirements for the system as a whole, and for its software and hardware components.

Proposed a methodology of creating requirements, allowing to organize the phased identification, analysis, documentation and verification requirements, taking into account the features of the ES.

The practical application of the proposed model and methodology has allowed to perform efficient development of the requirements and to successfully implement the project for embedded control system of mobile platform.

1. Платунов А. Е. Высокоуровневое проектирование встраиваемых систем. Часть 1 : учеб. пособие / А. Е Платунов, Н. П. Постников. – СПб. : НИУ ИТМО, 2011. – 121 с. 2. Встраиваемые системы [Электронный ресурс]/ Родник. – Режим доступа: www/ URL: http://www.rodnik.ru/ product/spa/embedded-solutions/ 3. ДСТУ ISO/IEC 14288:2004. Інформаційні технології. Процеси життєвого циклу системи (ISO/IEC 14288:2002, IDT). – Введ. 01.07.2007. – К. : Держстандарт України, 2004. – 48 с. 4. ГОСТ Р ИСО/МЭС 12207-2010 Information technology. System and software engineering. Software life cycle processes(ISO/IEC 12207:2008). – Введ. 01.03.2012. – Москва : Стандартинформ, 2011. – 104 с. 5. Wiegers K. E. Software Requirements. / Karl E. Wiegers. – 2nd Edition. – Microsoft Press, 2003. – 544 p. ISBN:978-0-7356-1879-4. 6. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – Введ. 01.01.1990. – М. : ИПК Издательство стандартов, 2004. – 11 с. 7. ГОСТ Р 41904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию. – Введ. 24.07.2002. – ИПК Издательство стандартов, 2002. – 62 с. 8. Leffingwell D. Managing Software Requirements: A Use Case Approach / Dean Leffingwell, Don Wirding – Addison-Wesley, 2003. – 402 page. – ISBN 0-321-12247-X. 9. Основы программной инженерии [Электронный ресурс]/ SWEBOK. – Режим доступа: www/ URL: http://swebok.sorlik.ru 10. Zielczynski P. Requirement Management Using IBM Rational RequisitePro / Peter Zielczynski. – IBM Press, 2007. – 360 pages. – ISBN 0-321-38300-1.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua