PRReSE – Process of Non-Functional Requirements Reuse for Embedded Systems Based on a NFR-Framework

Cristiano Marçal Toniolo
Faculty of Exact and Natural Sciences
Methodist University of Piracicaba (UNIMEP)
Piracicaba – São Paulo – Brazil
e-mail: cmtoniolo@gmail.com

Luiz Eduardo Galvão Martins
Institute of Science and Technology
Federal University of São Paulo (UNIFESP)
São José dos Campos – São Paulo – Brazil
e-mail: legmartins@unifesp.br

***Abstract* — The Embedded Systems are increasingly present in society's daily life. Their demand in several home appliances makes them more complex, bringing the necessity of a more careful requirements engineering than for traditional systems. The requirements reuse for embedded systems, especially those addressed to non-functional requirements, is still a challenge for industries that develop products based on Embedded Systems (ES). This paper presents a Process of Non-Functional Requirements Reuse for Embedded Systems – called PRReSE - using NFR-Framework as an approach to improve the concept, design and development of such systems. The process was instantiated in a case study to illustrate the reuse of non-functional requirements in a product family; the family chosen was for microwave oven.**

Keywords-Embedded Systems; Requirements Reuse; Non-Functional Requirements; NFR-Framework.

I. INTRODUCTION

Software Engineering and Requirements Engineering work together to find new ways to ensure the quality of software development. To achieve this, a step of great importance in the process is the requirements elicitation, which seeks the understanding of the user's needs. The elicitation process defines and documents the steps so that an organization can elicit, analyze, specify and verify the requirements [15].

The advancement of techniques and methodologies allows us thinking about a systematic requirements reuse throughout the project development. According to Sommerville and Sawyer, the requirements reuse saves time and efforts in their elicitation. About 80% of the requirements are reused when dealing with similar systems [10].

The requirements reuse is performed in several ways, e.g., software components and requirements to make the reuse process even more efficient and able to answer the market demands. However, the reuse performed in industries is somehow intuitive, since engineers and developers reuse methods and documents in new projects based on their own experience.

Non-functional requirement is a central concept in this work, which means a quality feature that can affect the entire system to be developed. This study presents a Process of Non-Functional Requirements reuse for Embedded Systems (PRReSE is a Portuguese acronym for *Processo de Reuso de Requisitos Não-Funcionais para Sistemas Embarcados*). The proposed process is based on a NFR-Framework [16], which is a method to assist engineers and software designers to produce software in a faster and with more quality way, in a high level of quality with the lowest possible cost. This is precisely the role of engineering, namely, look for best quality systems within a cost compatible with this quality [8].

In industry context, many projects are related to each other and their requirements can be stored and reused in new projects in the future. Such projects can be divided into innovation of previous projects or into product families.

The innovations are related to implementations of new features in products that do not have them yet. So, a new version can be available. Product Families are related to new versions of products from the same family, e.g., microwave ovens – or to the creation of new ones, though with features previously used.

Therefore, the motivations for this study is the fact that the requirements reuse is a widely used approach to management, web, financial and administration systems, but poorly used in embedded systems and even more if dealing with non-functional requirements.

This paper is organized as follows: Section 2 presents background and related works in requirements reuse. Section 3 shows PRReSE process. Section 4 presents the results and analysis of the performed case study. Section 5 concludes the paper and points out to future works.

II. REQUIREMENTS REUSE

Software Requirements have to be carefully elicited in order to not compromise the whole systems development. As discussed in Kotonya and Sommerville "requirements are defined at the first phases of the system development as being a specification to be implemented" [2].

Requirements describe the user's needs guiding developers how the system must behave, where it has to be applied and with some quality constraints. Several techniques have been used in order to solve the problems of reuse. One of them is the requirements reuse which, according to Sommerville and Sawyer, occurs "when developing requirements for a new system is necessary, wherever possible, reuse requirements from other systems which have been developed to the same area of application",

 ISBN: 978-1-61208-367-4

i.e., the same field of the system [10]. Reuse can reduce the cost, coding and testing of the project if it is systematically done; therefore, reducing the effort of new elicitation for several applications [2][3][9].

The advantages to adopt requirements reuse are the elicitation time saving, analysis and requirements validation, reduction of the risk of new elicitations that might hinder the requirements implementations, leading to a requirements reuse without alterations, or with minimal settings leaving the elicitation process only for new requirements of the system. This leads the system's development life cycle to start earlier.

The identification, capture and organization of a requirements process with the purpose of reuse in new systems can be considered a domain reuse approach. Kotonya and Sommerville show some situations where the reuse is possible [2]:

1. If the requirement shows information about the application domain: several requirements do not specify the system's functionality but presents its constraints or operation derived from the application domain.
2. If the requirement is consistent with the presentation of information style: if possible, common sense to organize, to have a consistent interface for all systems. It means that the user's errors are smaller when they change from one system to another.
3. When the requirements reflect the company's policies such as security and performance, they must be reflected in the system requirements. In this context, requirements are developed for the system and can be considered encapsulated requirements, which are common to a large number of different systems.

This way, for many systems, 50% of requirements are in such classes, which are a considerable scope for the cost reduction by requirements reuse [2]. Other reasons to perform the requirements reuse can be: requirements already analyzed tend to suffer few or no alterations; cost reduction of new requirements elicitation, which may lead to an incompatibility with other systems generating unexpected problems. The requirements reuse process has to be agreed for engineers and developers aiming to improve the system development cycle [13].

Some reuse processes can be cited as follows: analysis of domain, textual analysis, use cases patterns, scenarios, frameworks, direct and indirect reuse. The indirect reuse is specified as follows [10]:

1. Identify the requirements that are close or similar to the stakeholders requirements for the system being developed.
2. Show these requirements to the stakeholders and explain their meanings.
3. Ask where the requirements would be adequate or inadequate.
4. Rewrite the requirements according to suggestions and repeat the process until all the stakeholders agree with them.

The elicitation steps for direct reuse are as following:

1. Identify the common requirements between the existing system and the one to be developed.
2. Recognize the potentially reusable and relevant requirements in the existing system to identify the common features.
3. Evaluate the possible reusable requirements with the purpose of validate them with the stakeholders for the new system to be developed.
4. Check with users if the requirements meet their needs.

A product family approach is another usual way of requirements reuse. This process is based on two concepts: strong reuse and weak reuse [7]. In the strong reuse, the requirements must be synchronized with the associated products, and any alterations on them affect the entire product family.

In the weak reuse, the requirements are copied from the beginning of the project and they can evolve from other requirements.

Another way to identify product families, according to Lam, McDermind and Vickers [4], is that "requirements are sensitive to the context and are specified to a set of problems", then in product families it is possible:

- To identify commonalities between the system "father" and the system "son";
- To impose a common requirements engineering process or a pattern inside the organization;
- To antecipate some types of alterations and specializations;
- To recognize labor patterns to assist the planning of the project [4].

The reuse problems may go through some issues that make them a difficult task, such as: the engineering methods and techniques that are not specifically designed to reuse; the process which does not prioritize an integrated development and exchanging experience among the team members; the organization which works with individual projects without reuse planning; and the business that aims profit and works with financial return only.

Sommerville [11] lists a number of problems such as: maintenance cost, lack of tools and specific development, "non-invented-here" syndrome, creation of a library or repository to store the components. These factors impact the development costs. When it comes to reuse, it is possible to have models for requirements patterns, reuse of documents and artifacts, which makes this area very embracing and without a definite pattern to reuse system requirements along with the lack of specific tools but with several techniques addressed to each type of context and problem.

III. PRReSE: Process of Non-Functional Requirements Reuse for Embedded Systems Based on a NFR-Framework

Studies are performed in order to shorten the steps of the development process and consequently save time with development and cost reduction. Researches has shown that many efforts have been made to reuse software components and requirements, aiming to accelerate the development process in computer programming since the market competition is increasingly fierce. The paper focuses on a Process of Non-Functional Requirements Reuse for

 ISBN: 978-1-61208-367-4

Embedded Systems based on a NFR-Framework, which intends to become a guidance for professionals in appliance industry, assisting them when developing new systems reusing non-functional requirements for embedded system previously developed.

A. PRReSE: General Flow

PRReSE starts with the phase of separation of artifacts which can generate inputs for the requirements reuse activities [9][12]. These inputs correspond to previously analyzed data, which are considered as requirements feeding the requirements reuse process. According to [5][6][9], the following documents can be considered as being input artifacts: questionnaires, reference models, checklists, documents based on patterns, documented interviews, catalogues or technical descriptions of a system. Other artifacts can be found in *IEEE Std 830-1998* models and in *Volere Template* [1][14]. The general model for the requirements process described in this study is shown in Figure 1. The main activities of the proposed process are explained in the next sections.

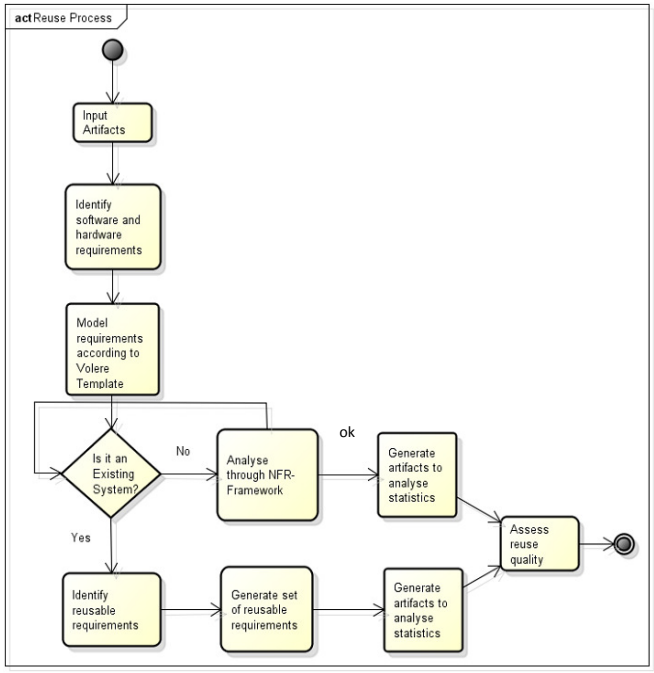

Figure 1. General Flow of Requirements Reuse Process for Embbeded Systems (PRReSE).

B. Identification of Software and Hardware Requirements

In this phase, the requirements engineer has to identify, select and prioritize requirements through input artifacts with the purpose of producing output artifacts to the next step, which can be divided into software and hardware non-functional requirements.

Input artifacts are the analyzed data in the documents collect from the stakeholders, whose specific goal is to create knowledge of the software context to be modeled by the requirements engineer, electronic engineers and technicians. Technical manuals and catalogues can be cited as examples of such documents. After separating these documents, requirements engineering process is in charge to identify the non-functional requirements for software and hardware. The next step is modeling according to Volere Template.

C. Modeling Non-Functional Requirements according to Volere Template

After the first phase of PRReSE, when the requirements document is generate, the next step is to create the non-functional requirements cards and store them to be used in NFR-Framework.

The non-functional requirements documents created in the previous step are the start up for the modeling process. Thus, the non-functional requirements can be catalogued using the *Volere Template* cards. The output artifact for this PRReSE step is the creation of all non-functional requirements cards to model according to NFR-Framework.

D. Checking the Existence of Legacy System

This step shows two conditions when modeling an embedded system: the existence of a legacy system or the lack of a legacy system.

E. Lack of a Legacy System

An analysis to verify the existence of a modeled and compatible system is performed after to model non-functional requirements according to *Volere Template* to apply PRReSE. Otherwise, the following steps are:

1. When the system under consideration has not been elicited or do not have any relationship with another system – *product families* – it is necessary to perform all the analysis for the artifacts described by the NFR-Framework. This means creating a catalogue of knowledge to form a basis to research future systems.
2. Develop SIG graphs to model the relationship among non-functional requirements and expose them graphically.
3. Generate the modeled graphs for the entire systems, as output artifacts.

F. Existence of Legacy System

The analysis to verify the existence of an already modeled system compatible to apply PRReSE is performed after the creation of the *Volere Template* cards. The steps are:

1. Identify the reusable requirements according to the following procedures:
   a. If there is any relation to some existing product or product families, search the catalogues already created, verify the presence of requirements that can be reused and analyze only the compatible requirements.
   b. This identification will be performed through the comparison of the requirements modeled in SIG graphs with the requirements collected in the input artifacts (*Volere* cards). Thus, when the requirements are in the upper levels of the graph,

 ISBN: 978-1-61208-367-4

the reuse will have little or no alterations at all. In the lower levels of the graph, the reuse will tend to be performed.

2. Generate a set of reusable requirements: it happens after the comparison with the legacy systems. Thus, the SIG graphs are created and the candidate requirements for reuse are identified.
   a. Identical requirements are reused without any alteration.
   b. Requirements with some similarities to a specified NFR must be reused, with the necessary alterations. Also, they must be identified in the SIG graphs with dashed circle.
   c. Requirements that are not catalogued must be entirely elicited and represented in the SIG graphs by a solid circle.
3. Create Artifacts for Statistical Analysis: after creating SIG graphs and identifying the requirements which were or were not entirely used, they are quantified and tagged as entirely used, partially used or not used. Then the steps bellow are followed:
   a. Recover all the stored SIG Graphs;
   b. Quantify the requirements entirely used;
   c. Quantify the requirements partially used;
   d. Quantify the requirements not used.

G. Analysis of Reuse Quality

This analysis defines the reuse viability and how the indicators quantify the reuse of non-functional requirements. From these results, the quantity is verified and a reuse pattern of quality will be established in percentage:

a. Reused without alterations;
b. Reused with alterations;
c. New requirements elicited with the stakeholders.

Such analysis must be performed based on the equation (1) and Table 1:

$$(1) \quad RP = (QRR / QR) * 100$$

where:
RP = Reuse Percentage
QRR = Quantity of reused requirements
QR = Total quantity of requirements

TABLE I. REUSE INDEX FOR NON-FUNCTIONAL REQUIREMENTS

Reuse Index	Reuse Percentage
High	RP ≥ 85%
Adequate	RP ≥ 60% e RP < 85%
Insufficient	RP ≥ 40% e RP < 60%
Inadequate	RP < 40%

Table 1 presents indicators which may assist software engineers to get an idea of how much will be necessary to elicit new software from existing requirements and the quantity of reuse it may be generated providing time and profit.

H. Analysis of the Obtained Results

This analysis defines which percentage of requirements reuse was obtained with PRReSE, and also if the work of preparing the software design will become viable, consuming less effort from those involved in the project. It will also allow the software and requirements engineers to get the parameters to develop new projects, such as time to analyze and elicit requirements, making this process to work as a knowledge and learning basis with experiments from previous projects.

IV. CASE STUDY: MICROWAVE OVEN

This case study performed the steps of the Reuse of Non-Functional Requirements Process for Embedded Systems showing that PRReSe becomes a feasible alternative for requirements reuse for embedded systems.

A. Requirements Identification

The requirements for the microwave oven family were extracted from the catalogue of products previously studied, because there is a lack of documentation related to the embedded systems used in microwave ovens. This way it was possible to elaborate the requirement cards according to the model suggested by *Volere Template*.

Panasonic's microwave oven manual of the flat and family models were used as input artifacts, and the following non-functional requirements were extracted: security, usability, customization, learning, accessibility and capacity.

B. Non-Functional Requirements Cards

The requirements cards based on *Volere Template* were fulfilled after the requirements identification. Such cards offer a pattern structure to describe the requirements, which turn easy the work of requirement engineers during the reuse process.

C. Non-Functional Requirements Catalogue

Non-Functional Requirements Catalogues are modeled for embedded systems (ES) after the requirements and the cards get ready. These catalogues have the purpose to show non-functional requirements in a hierarchical form with the generic NFRs displayed above the more specific ones, as showed in Figure 2.

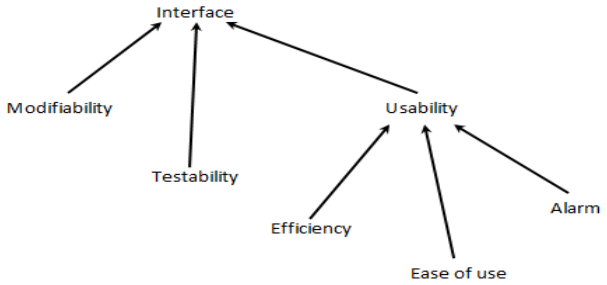

Figure 2. Interface Non-Functional Requirements for Microwave Oven.

The SIG graphs for such requirements were subsequently modeled, and the resulted specification model are presented in Figure 3. The reuse model illustrated in the figure shows three levels of requirements, which are: requirements without alterations - cloned objects, requirements with some type of alterations – derived objects, and new requirements specified for the project being developed – new objects. The

requirements for the Panasonic Flat-Style microwave NF-SF560WRU™ were initially modeled becoming the base for the requirements related to the product family, as shown in Figure 4. This model of oven has simple features and basic commands. Monzon [7] explains that to have the requirements reuse done there must be a set of common requirements, which will be reused in new projects. This set of requirements becomes the core for the reuse in embedded systems projects or a product family, which will be reused in each evolution or new version of a product.

Figure 3. Model of Non-Functional Requirements Reuse from a common requirements set [7].

D. Systems Evolution

The model NN-GF580MRU™ from the Panasonic Flat Family™ was used and the requirements were elicited and analyzed. SIG graphs were created to compare it with the previous model – according to Figures 5 and 6. The softgoals which needed to be elicited from the catalogue of product due to advanced features were added to the core requirements already presented in the previous model. An analysis of a model of a microwave oven, which is not part of the adopted product family was performed to show the requirements reuse process extent. The used model was NNST669WRU™ with innovative features. The SIG graph referring to the model above was created to become possible the comparison with the core requirements in the previous models, as illustrated in Figure 6.

E. Analysis of the Results

Analysis of the results was performed from the creation of both requirements cards and SIG graphs, according to what has been proposed in the non-functional requirements process. Table 2 summarizes the requirements from the reuse process, which are commented in the next sections.

TABLE II. RESUME OF NON-FUNCTIONAL REQUIREMENTS OF MICROWAVE OVEN

Model	(a)	(b)	(c)	(d)	(e)	Reuse
NN-SF560WRU	31	0	0	0	0	0%
NN-GF580MRU	33	31	31	0	2	94%
NN-ST669WRU	35	31	31	1	3	88%

(a) Non-Functional Requirements Total
(b) Reused Requirements Total
(c) Reused Requirements without alterations
(d) Reused Requirements with alteration
(e) New Requirements

F. Reused requirements without alterations

The model NN-SF560WRU was the first to be analyzed and 31 non-functional requirements were elicited. The second model, NN-GF580MRU had 33 non-functional requirements elicited and 31 of them were reused from the first model, which correspond to 94% of requirements reuse. The third model NN-ST669WRU had 35 non-functional requirements elicited. From this total, 31 requirements were reused, meaning 88% of reuse.

G. Reused requirements with alterations

The first two models of microwave oven used in this case study belong to the same product family and the third model belongs to a different product family. The non-functional requirements "stand by key" in the model NN-GF580MRU was the one reused with alterations. Here, the "stand by key" was matched to the "clock key" resulting in a "stand by/clock key". This model had 35 elicited requirements, which 2.8% correspond to requirements reused with alterations.

H. New Requirements

Two new requirements were created for the second model, which belongs to the same family of the first model. These two new requirements mean 6% of the elicited requirements. For the third model – which does not belong to the family of the first one – three new requirements were created, meaning 8.5% of the elicited requirements.

I. Reuse Analysis

The Requirements Reuse in the second model of microwave oven analyzed saved time in the analysis of requirements phase since the biggest efforts were performed in the base model. Thus, the requirements reuse in products of the same family leaded to a significant time saving.

The study case showed that with PRReSE adoption was possible to obtain 94% of requirements reuse in the second model in relation to the first one, which belong to the same family indicating a promising reuse process.

PRReSE allowed 88% of requirements reuse in the third model of microwave oven in relation to the first, even considering that the third model belongs to a different product family. This result also can be considered as a promising one. Since this model belongs to a different family, could be expected a lower percentage of reuse.

V. CONCLUSION AND FUTURE WORK

The requirements reuse in traditional systems – not embedded - inspired PRReSE creation in a yet not enough explored context, i.e., non-functional requirements for embedded systems. NFR-Framework was adopted because it is a specific methodology to NFRs, and also it is largely known in the Requirements Engineering community.

The productivity of embedded systems development can significantly increase using the requirements reuse techniques, especially because in embedded systems development is very common to use product families. The statement above can be seen in the microwave oven case study, which applied PRReSE to reuse the requirements in three models, being two from the same family and one from a different product family.

There was a large effort to elicit requirements at the first model, then using PRReSE process it became easier to reuse

 ISBN: 978-1-61208-367-4

the resulting requirements in the following models. According to PRReSE, the visualization of non-functional requirements through SIG graphs becomes the reuse identification easier by the requirements engineering.

According to the evidences observed in the case study, during the creation of the second model 94% of non-functional requirements were reused from the first model – both belonging to the same product family. PRReSE allowed the reuse of 88% of the non-functional requirements in the third model, considering that this last one did not belong to the same product family.

As future work, a software tool to support PRReSE will be developed to facilitate its use. Another case study performing all steps of PRReSE is being planned; such study will be in the area of medical devices.

REFERENCES

[1] IEEE, "IEEE Recommended Practice for Sotware Requirements Specifications". IEEE Std 830-1998.

[2] G. Kotonya and I. Sommerville, Requirements Engineering: process and techniques, Ed. Wiley, 2001.

[3] W. Lam, "A case-study of Requirements Reuse through product families". In: Annals of Software Enginering, vol. 5, 1998, pp. 253 – 277.

[4] W. Lam, J. McDermind, and A. Vickers, "Ten Steps Towards Systematic Requirements Reuse", Proceedings of the Third IEEE International Symposium on Requirements Engineering. Jan/1999, pp. 6-15.

[5] X. Liu, S. Lui, and X. Zheng, "Adapting the NFR Framework to aspectual use-case driven approach", In: 7th ACIS International Conference on Software Engineering Research, Management and Applications. Dec/2009, pp. 209-214.

[6] O. López, M. A. Laguna, and F. J. Garcia, "Metamodeling for Requirements Reuse", In: Proceedings of the 5th. International Workshop on Requirements Engineering (WER'02). Valencia, Espanha Nov/2002.

[7] A. Monzon, "A pratical approach to Requirements Reuse in product families of on-board systems", In: 16th IEEE International Conference on Requirements Engineering. Sep/2008, pp. 223-228.

[8] A. R. C. Rocha, Qualidade de Software: teoria e prática. São Paulo : Prentice Hall, 2001.

[9] J. Rumbaugh, Object-Oriented Modeling and Design. International Publlisher: Prentice-Hall. International Pub., 1990.

[10] I. Sommerville and P. Sawyer, Requirements Engineering, John Wiley, 2000.

[11] I. Sommerville, Software Engineering, 9th Edition, Addison-Wesley, 2009.

[12] S. Supakkul, "Capturing, organizing, and reusing knowledge of NFRs: an NFR pattern approach", In: 2nd International Workshop on Managing Requirements Knowledge (MARK). Sep/2009, pp. 75-84.

[13] O. Villegas and M. A. Laguna, "Requirements Reuse for software development", In: 5th IEEE International Symposium on Requirements Engineering. Toronto, Canada, Aug/2001, pp. 27-31.

[14] J. Robertson and S. Robertson, Volere Requirements Specification Template. Atlantic Systems Guild London, 2011.

[15] K. L. Wiegers, "Requirements specification template". Microsoft Press, 1999.

[16] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, "Non-Functional Requirements in Software Engineering" In: The Kluwer International Series in Software Engineering, Vol. 5, 1999.

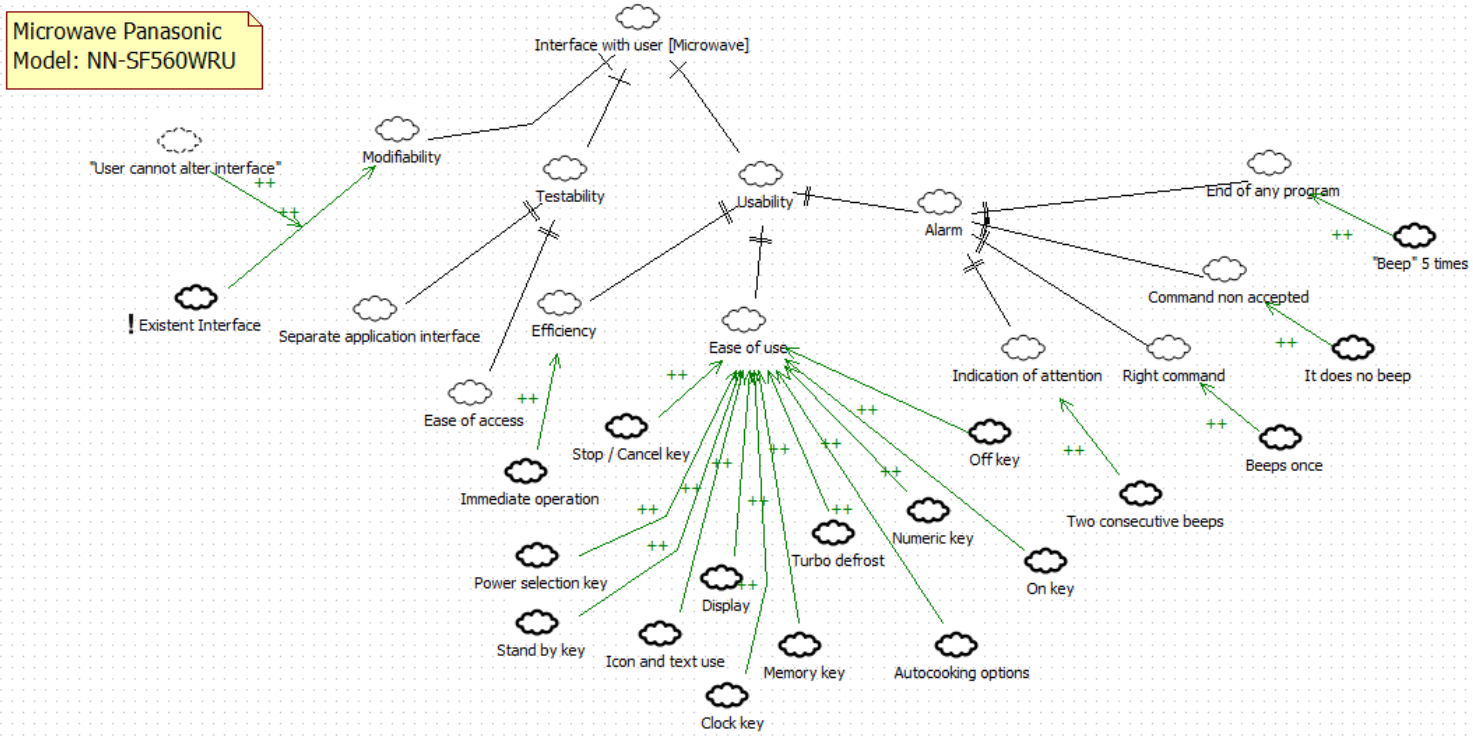

Figure 4. SIG Graph for "User Interface" requirement for microwave oven model NN-SF560WRU.

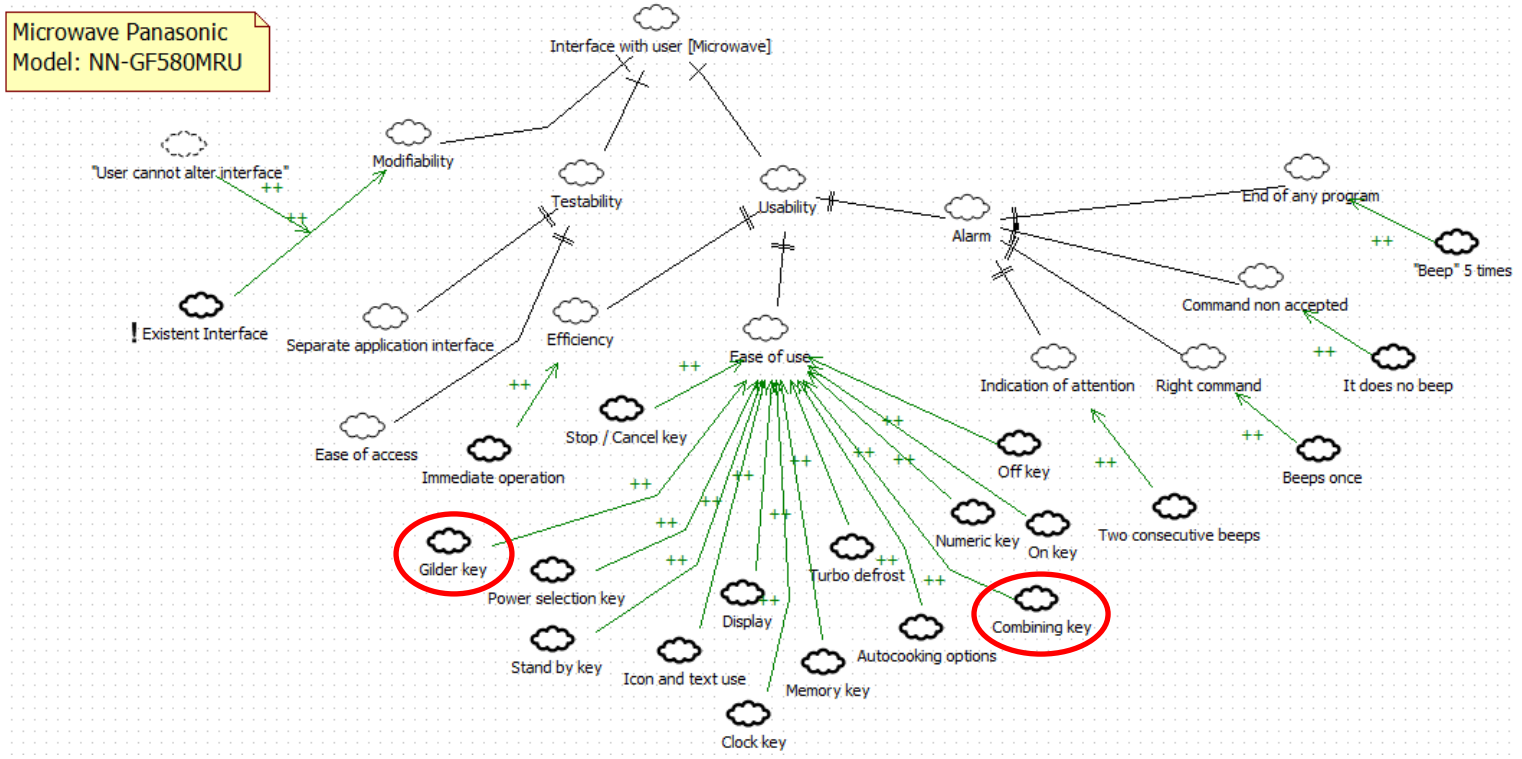

Figure 5. SIG Graph for "User Interface" requirement for microwave oven model NN-GF580MRU.

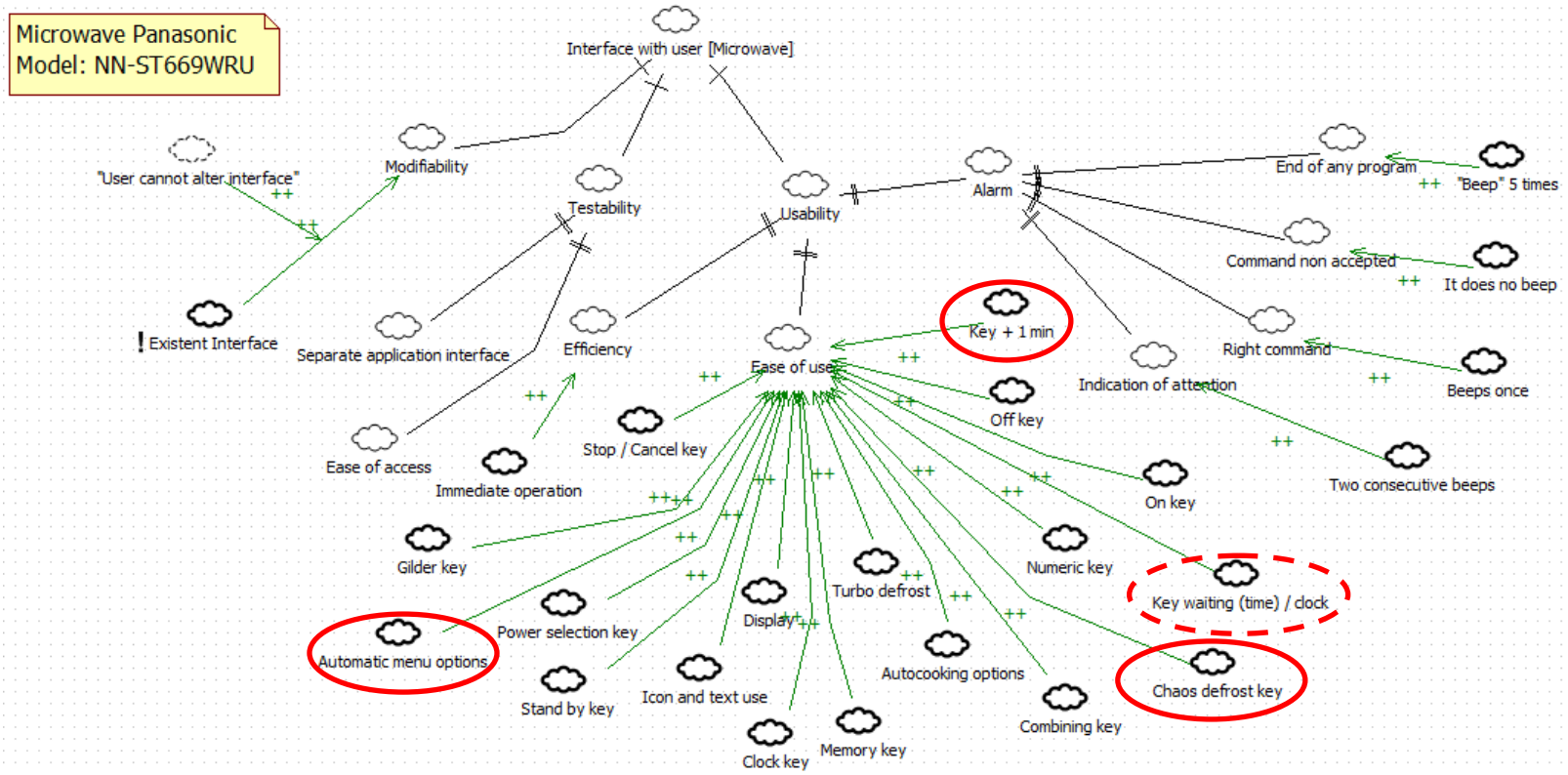

Figure 6. SIG Graph for "User Interface" requirement for microwave oven model NN-ST669WRU.