DOI: 10.26565/2304-6201-2020-47-02

УДК 004.047

Handling of expert knowledge in software product line development with usage of repertory grid method

R.O. Gamzayev, M.V. Tkachuk, D.O. Shevkoplias

Gamzayev Rustam *PhD, asc. professor, post-doctorate of the Department of Systems and Technology Modeling,*
V.N. Karazin Kharkiv National University, Kharkiv, Freedom Square 4, 61022,
e-mail: rustam.gamzayev@karazin.ua ;
https://orcid.org/0000-0002-2713-5664

Tkachuk Mykola *Dc. of techn. science, professor, head of the Department of Systems and Technology Modeling,*
V.N. Karazin Kharkiv National University, Kharkiv, Freedom Square 4, 61022,
e-mail: mykola.tkachuk@karazin.ua;
https://orcid.org/0000-0003-0852-108

Shevkoplias Daria *student of the Department of Systems and Technology Modeling,*
V.N. Karazin Kharkiv National University, Kharkiv, Freedom Square 4, 61022,
e-mail: shevkoplyasdarya23@gmail.com*;*
https://orcid.org/0000-0001-5870-771X

The development of software product lines (SPL) is a promising way to create competitive software for the modern IT - market. Because of perpetual changes in user needs and the appearance of new software and hardware components for SPL, maintaining a certain level of variability at all stages of their full life cycle (FLC) becomes an actual problem. One of the most important stages is requirements engineering. To support this process it is advisable to utilize the methods of expert knowledge elicitation and analysis. Based on the recognized multilevel approach to SPL engineering proposed by K. Pohl, the research goal is to develop an approach to correct specification of variability features at the initial stages of SPL FLC. The analysis of methods to extract and process the expert knowledge is carried out, their classification is presented, and the method of repertory grid (RG) is chosen for the further usage. It allows analyzing and structuring expert knowledge in a multidimensional information space and makes it possible to consider various contextual factors that affect relevant requirements for the target SPL. The essentials of the RG method are considered and its formal description is elaborated. The examples of its usage in designing "Smart House" systems which allow analyzing the technical, social and economic aspects of user requirements for software and hardware variability at the conceptual design stage of these systems are considered. As the result, a possibility to automate a process of a corresponding FODA-model creation is shown. That provides variability support at the next FLC stages of such systems.

Keywords: *expert knowledge, software products line, life cycle, variability, repertory grid method, "Smart-House" system.*

Обробка експертних знань в проектуванні лінійок програмних продуктів із застосуванням методу репертуарних сіток

Р.О.Гамзаєв, М.В. Ткачук, Д.О. Шевкопляс

Гамзаєв Рустам Олександрович *кандидат технічних наук, доцент, докторант кафедри моделювання систем і технологій,*
Харківський національний університет імені В.Н. Каразіна, м. Харків, Майдан Свободи 4, 61022

Ткачук Микола Вячеславович *доктор технічних наук, професор, завідувач кафедри моделювання систем і технологій,*
Харківський національний університет імені В.Н. Каразіна, м. Харків, Майдан Свободи 4, 61022

Шевкопляс Дар'я Олександрівна *студентка кафедри моделювання систем і технологій,*
Харківський національний університет імені В.Н. Каразіна, м. Харків, Майдан Свободи 4, 61022

Розробка лінійок програмних продуктів (ЛПП) є перспективним шляхом для створення конкурентноздатного програмного забезпечення на сучасному ІТ – ринку. За умов постійних змін потреб користувачів та появи нових програмно-технічних засобів для реалізації таких систем актуальною проблемою стає можливість забезпечення певного рівня варіабельності проектних рішень на всіх етапах їх повного життєвого циклу (ПЖЦ). Одним з найбільш складних з них є етап визначення початкових вимог користувачів до майбутньої системи і для підвищення ефективності цього процесу доцільно застосовувати методи виявлення та аналізу експертних знань у відповідній предметній області, для якої створюється та чи інша ЛПП. В статті, на основі аналізу загальновизнаного багаторівневого підходу до розробки ЛПП, запропонованого в роботах К. Поля (K. Pohl), сформульована основна мета дослідження: розробити підхід до коректного визначення властивостей варіабельності цільової системи вже на початкових етапах її ПЖЦ. Для цього проведено аналіз та запропоновано класифікацію методів видобутку та обробки експертних знань, і для подальшого застосування мотивовано обрано метод побудови репертуарних сіток (РС). Він дозволяє проводити аналіз та структурування експертних знань у багатовимірному інформаційному просторі, що дає можливість враховувати різні контекстні чинники, які можуть впливати на множину відповідних вимог до майбутньої системи. Розглянуті концептуальні засади методу побудови РС та запропоноване формальне теоретико-множинне визначення для його основних складових: елементів, конструктів та рейтингових оцінок. На прикладах з предметної області розробки систем «Розумний дім» з використанням такого підходу наведені конкретні приклади побудови РС, які дозволяють враховувати технічний, соціальний та економічний аспекти вимог користувачів щодо підтримки варіабельності програмно-апаратного забезпечення на етапі концептуального проектування цих систем. В результаті дослідження показана можливість реалізації автоматизованого процесу побудови відповідної FODA-моделі для забезпечення властивостей варіабельності на наступних етапах ПЖЦ систем цього типу.

Ключові слова: *експертні знання, лінійка програмних продуктів, життєвий цикл, варіабельність, метод репертуарних сіток, система «Розумний дім».*

Обработка экспертных знаний в проектировании линеек программных продуктов с применением метода репертуарных сеток

Р.О. Гамзаев, Н.В. Ткачук, Д.А. Шевкопляс

Гамзаев Рустам Александрович — *кандидат технических наук, доцент, докторант кафедры моделирования систем и технологи, Харьковский национальный университет имени В.Н. Каразина, г. Харьков, Площадь Свободы 4, 61022*

Ткачук Николай Вячеславович — *доктор технических наук, профессор, заведующий кафедрой моделирования систем и технологий, Харьковский национальный университет имени В.Н. Каразина, г. Харьков, Площадь Свободы 4, 61022*

Шевкопляс Дарья Александровна — *студентка кафедры моделирования систем и технологий, Харьковский национальный университет имени В.Н. Каразина, г. Харьков, Площадь Свободы 4, 61022*

Разработка линеек программных продуктов (ЛПП) - перспективный путь создания конкурентоспособного программного обеспечения для современного IT-рынка. Постоянные изменения в потребностях пользователей и появление новых средств разработки ЛПП требуют обеспечения свойств вариабельности на всех этапах их полного жизненного цикла (ПЖЦ). Важным этапом ПЖЦ является анализ начальных требований и для поддержки этого процесса рекомендуется использовать методы обработки экспертных знаний. На основе анализа многоуровневого подхода К. Поля к созданию ЛПП определена цель исследования: разработать подход к корректной спецификации вариабельности на начальных этапах ПЖЦ ЛПП. Проанализированы методы извлечения и обработки экспертных знаний, предложена их классификация и для дальнейшего использования выбран метод репертуарных сеток (РС). Он позволяет анализировать и структурировать экспертные знания в многомерном информационном пространстве и учитывает различные факторы, влияющие на требования к целевой ЛПП. Рассмотрены основы метода РС, дано его формальное определение и показаны примеры его использования для проектирования систем «Умный дом», что позволяет учитывать технические, социальные и экономические аспекты требований пользователей к изменчивости программного и аппаратного обеспечения на этапе концептуального проектирования этих систем. В результате показана возможность автоматизации процесса создания соответствующей FODA-модели, обеспечивающей поддержку вариативности на следующих этапах ПЖЦ таких систем.

Ключевые слова: *экспертные знания, линейка программных продуктов, жизненный цикл, вариабельность, метод репертуарных сеток, система «Умный дом».*

1. Introduction: Problem actuality and research goals

The current situation in Software Engineering (SE) is characterized by the growing number of sophisticated methodologies, advanced methods and powerful tools aiming to develop software products at a lower cost, in a shorter time, and accounting for their quality attributes. The development of software product lines (SPL) is a highly promising way to create competitive software for the modern IT - market. Because o of perpetual changes in user needs and the appearance of new software and hardware components for SPL design and implementation, maintaining an appropriate level of variability at all stages of their full life cycle (FLC) becomes an actual problem. The recognized multilevel approach to SPL engineering within its FLC was proposed by K. Pohl [1], and its graphical representation is shown in Fig.1.

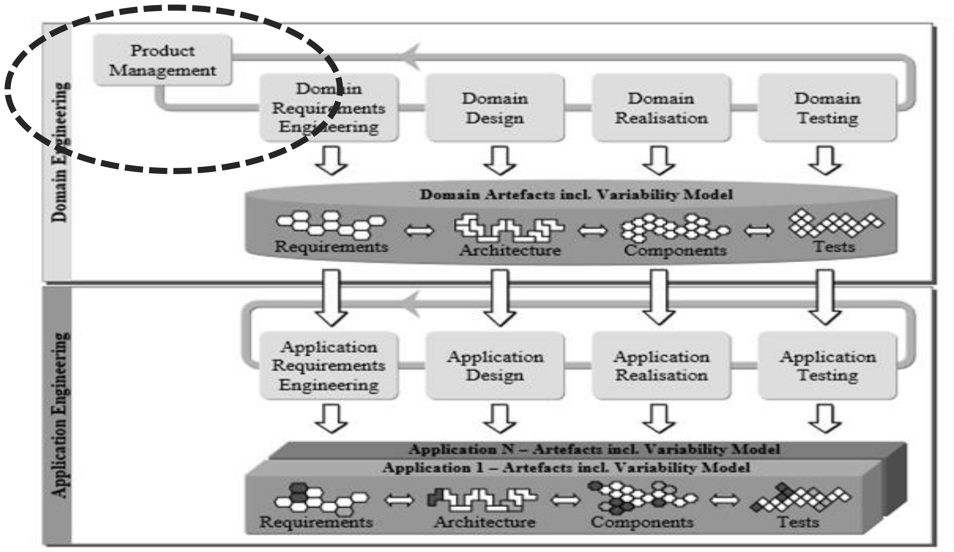

Fig. 1 The software products line engineering framework [3]

According to this engineering framework the FLC of any SPL includes a set of activities to produce the appropriate project artifacts which are structured at the 2 logical levels, namely (see Fig.1):

(I) Domain Engineering Level (DEL) provides a reusable operational platform and, therefore, it defines the *commonality* and the *variability* of future SPL components;

(II) Application Engineering Level (AEL) is responsible for the development of SPL-applications from the operational platform established at the DEL.

Each of these 2 levels: the DEL and the AEL, has a similar sequence of the several ordered project phases: *Requirements Engineering*, *Design*, *Realization* and *Testing*, and in the theory and practice of the modern SE there are a lot of appropriate development methods and CASE-tools to support them effectively [2, 3].

But at the DEL level there is a special project phase (see Fig.1): a phase of *Product Management* which is closely interconnected with a phase of *Domain Requirements Engineering*. Its main goal is eliciting and structuring all important user needs which have to be fulfilled by a target software product, especially, in a SPL to be developed. It should be mentioned that this phase is the most uncertain and weak-formalized one in FLC, and in order to support business analysts and software architects, who are supposed to make the project decisions, some interdisciplinary approaches are to be used (see e.g. in [4]). One of such approaches is a knowledge-based one [5], where a term *knowledge* can be formulated accordingly to the recommendations of the System Thinking World Community [6], namely: a *knowledge* is a collection of structured information objects and relationships, combined with appropriate semantic rules for their handling in order to get new proven facts about a given problem domain. That is why the main goal of the current research is to elaborate an approach to effectively handling the knowledge on the *Product Management / Domain Requirements Engineering* phases within SPL full life cycle (see Fig.1).

This paper is structured as follows: in the second section the variability issues for the one of important modern technical domains: development of smart-house systems (SHS) are discussed; in the third section some approaches to expert knowledge handling are overviewed and classified, and the method of repertory grid (RG) is chosen for the further consideration; in the fourth section the essentials of the repertory grid method are presented and illustrated with the examples in the SHS domain, and in the fifth section the usage of RG for building of a target feature-oriented model is shown. In the last section the conclusions and application for the further works are discussed.

2. The application area and problem statement: variability issues in "Smart-house" system (SHS) domain

In order to present our approach, we have decided to choose SHS which is rather complicated problem area, because the solutions that are developed within this domain usually combine software and hardware parts such as sensors, actuators, smart lighting systems, devices responsible for opening of doors/windows, etc. (Fig. 2). Hardware parts are developed by different vendors and use different protocols and, therefore, could require a lot of proprietary software applications. The system development under these conditions has the following potential drawbacks:

1. SHS management complexity is increased;
2. It becomes very complicated to implement some scenarios that interconnect devices from different vendors;
3. A number of different applications is required in order to work with different parts of the system.

It should be noted that most of the devices have quite similar functionality. For example, an alarm system could send a notification to the messenger, and a water sensor could do the same. In the same way some devices could share information about current conditions. That requires providing similar settings for different devices.

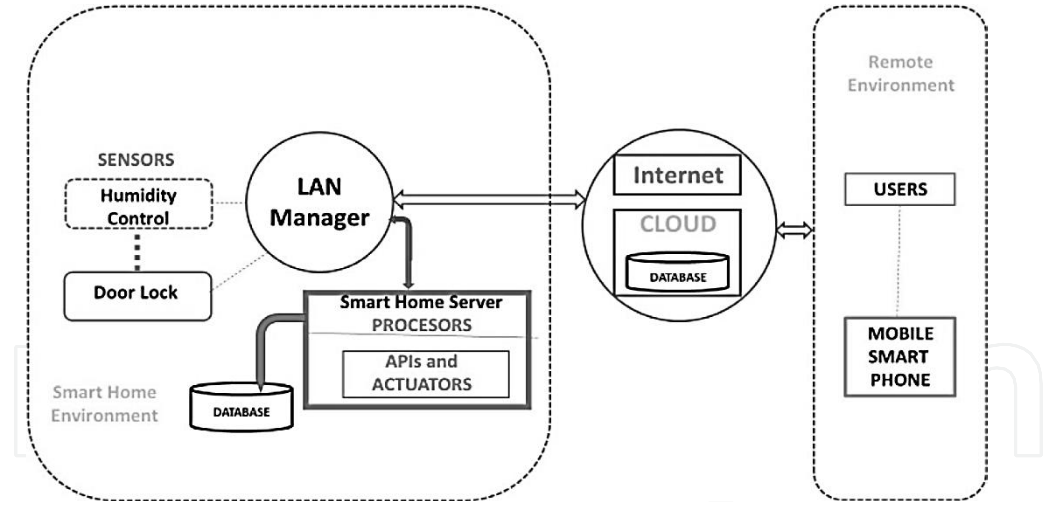

Fig.2 Modern SHS multi-level architecture [7]

There are the following SHS-specific features and problems:

- a lot of interconnected hardware and software components;
- all components have a lot of specific features and parameters;
- components operate in real-time mode;
- operation environment is dynamic and changeable;
- SHS has several groups of users;
- hardware components developed by different vendors;
- different protocols for communication.

After analyzing some of the abovementioned problems it becomes possible to merge them into the single integration problem at the software level. The main issue here is necessity to combine specific components of different platforms into one control system, with the possibility of centralized management of all kinds of equipment. The second issue can be expressed as the need to implement support for variability (or adaptability) of properties of software components, basing on the permanent presence of a set of different groups of requirements and their connection/dependence in the user group.

Based on these conditions and in order to formalize analysis process and synthesis of the SHS system structure, the corresponding initial domain model was built using the FODA notation [8], which is shown in Fig. 3, where the problem part of the system (hardware modules) is outlined with a bold line. The structure of this model contains corresponding sensors and actuators, and different subsystems. Sensors and actuators are controlled by special control unit - "Hub", and system engineers responsible for SHS maintenance, usually need a separate hub for each of the subsystems.

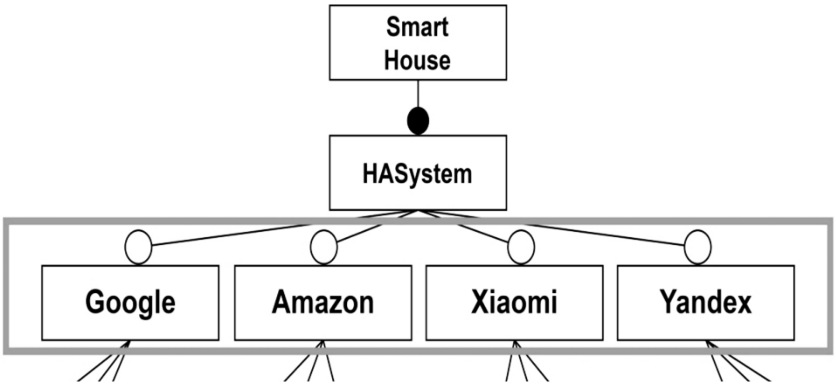

Fig.3 Technological variability issues in SHS

According to this first vision of the FODA-model (Fig.3), the integration problem is the necessity to use separate hub for each vendor, and, often, to use the proprietary applications to control the SHS elements due to the lack of integration and implementation of these functions into some common control system [8]. The second group is partially related to the first one, since a large amount of duplication of different functionality appears at different levels of the system, from the user interface to the devices themselves. When introducing variability support processes, the interface can be combined into a single application, due to the integration of the functionality of the selected components into one unit.

According to the problem mentioned above the main development objective is “How to design and implement configurable and adaptable complex software and hardware solutions, taking into account different users’ needs and system requirements?”

One of the approaches that makes it possible to omit the problem described above and to reach development objective is the SPL development [1, 3]. Any SPL includes methods for the variability management, the main idea is to manage configuration within a product family, plan the product development and elaborate a strategy for the further development.

The SHS SPL development process starts from the requirements elicitation and modeling the features for SPL. Achieved results are used to determine the similarity and variability in the SPL and define dependency in the system [8]. On the next stage of the SHS SPL modeling, previously defined variable and non-variable components become part of the feature model that defines a functional structure, similarity and reuse of the system elements. Dependencies and interfaces of such components are defined on this stage as well.

After domain analysis is finished and conceptual feature model is elaborated, it is used for the product development. The system architecture is build using variability components according to the model.

3. Some approaches to handling of expert knowledge and the motivated choice of repertory grid method

The concept of knowledge-based systems is focused on the processing of human knowledge that was received as a result of the experts’ actions in the professional domain [9]. An example of expertise in the SPL domain is the knowledge connected with the software architecture of such systems, conditions and metrics that should be considered, as well as, possible design/architecture patterns.

The difficulties in the knowledge elicitation process result from the following factors:

1. The influence of the experts’ experience or their personal opinion in the selected domain;

2. Absence of the theoretical methods to describe expertise;
3. Difficulty in the verbal expression of the knowledge.

Therefore, the main task could be formulated in the following way: to develop a control system that would facilitate the automated process of expertise elicitation.

Systems that could be used for knowledge elicitation could be classified by the expert methods used as follows [9]:

1. Systems based on Decision Trees;
2. Systems based on the psychological methods, such as Repertory Grid, Cluster Analysis, Multidimensional Scaling;
3. Systems that use models and methods for solving specific problem - Problem-Solving Methods;
4. Systems based on Case-Based Reasoning;
5. Inductive methods of knowledge elicitation, in which knowledge of expertise domain can be represented by examples;
6. Systems with combined methods and approaches for knowledge elicitation.

This classification is represented in Fig. 4:

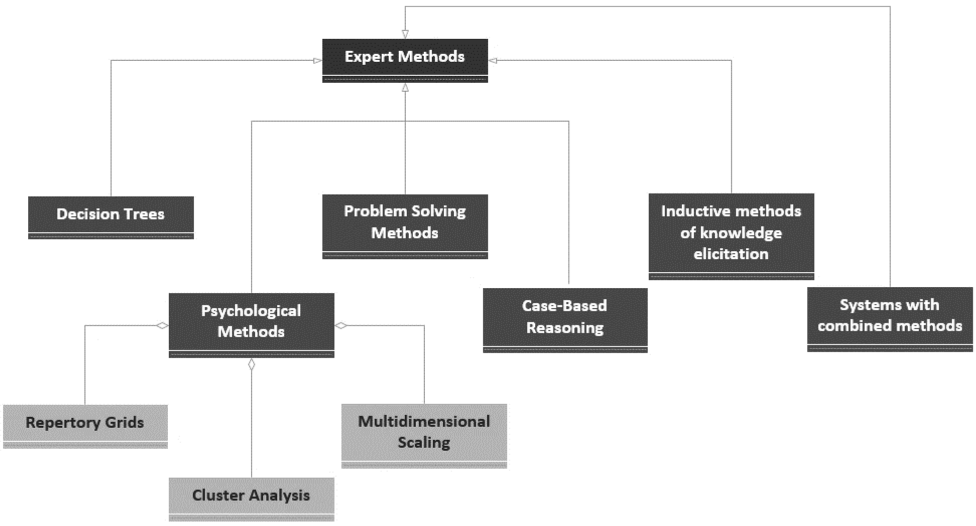

Fig.4 The classification of the expert methods based on their review given in [9]

We will overview some of these methods more closely.

Decision Trees method is the most commonly used and often applied to solve the problem of classification in machine learning. It is relatively simple method of data formalization that represents data as a directed acyclic graph consisting of nodes with rules and edges.

Repertory Grid method is an interview method based on Personal Construct Psychology theory initially designed by George Kelly. In some cases, this method is used together with a cluster analysis for the most effective handling of the expert knowledge. It is used to study personal and interpersonal systems of meaning, conceptual human structure (cognitive inclinations) in terms of interpersonal relationships by classifying a set of certain elements (e.g., other people) in terms of personal constructions [10]. The repertoire grid consists of the following components: elements, concepts and mechanisms that connect these elements and structures. The main idea of this method of identifying expert knowledge is to compare some elements according to the given characteristics.

Problem-Solving Methods are used to extract expert knowledge, build the appropriate database, identify missing knowledge and add/correct the database basing on the method of heuristic classification [9]. The problems solving method (heuristic classification) is performed in the following way:

1. Allocates for each failure or problem an explanation or possible alternatives;
2. Defines information for differentiation of these alternatives;

3. Select an approach to resolve the problem and plan the implementation of the best alternatives.

In this way the knowledge elicited from experts comes as a response to failures and complaints to be resolved. Thus, a multilevel network of problems and possible explanations is formed [9].

Case-Based Reasoning methods are based on the previous experts' problem solving practice and consist of the following stages [9]:

1. Eliciting one or more cases from expert practice;
2. Reusing and revising past knowledge and applying it to the new model, as well as, adapting it to the new situation in some cases;
3. Elicited knowledge is organized with a set of tables in order to consider all possible alternatives of reasoning and present the information as set of rules.

Inductive methods use a set of examples to represent knowledge about the expert domain. Such methods automatically build hypotheses about the connections and patterns of the studied domain. Inductive methods are classified according to the following criteria [9]:

- inductive rules learning,
- data-driven learning,
- model-driven learning,
- heuristic-based learning.

Systems that use a *combination* of different methods and approaches are the most powerful for expert knowledge elicitation, because shortcomings of one method are compensated by the advantages of others. However, such systems are often difficult to manage and maintain. There are following stages in such methods: knowledge extraction, task decomposition, information processing, step-by-step testing, data types integration, etc. [9].

For the further research the Repertory Grid Method is chosen, and its advantages for this purpose are discussed below.

4. Essentials of the repertory grid method: conceptual approach, formal definition and knowledge structured analysis with the examples in SHS domain

The main idea of the method of repertory grid (RG) is based on the assumption that a cognitive thinking of experts in some problem domain can help to decrease the process complexity and developer efforts by requirements engineering, especially, to determine appropriate variability features for the software products to be developed [10,11].

Any RG is built as a matrix of some dimension (usually, not higher than 3x3) with different contexts (represented as matrix columns) that influence system functionality in any way, and its characteristics (structured as matrix rows), that can vary depending on a specific problem domain. According to this vision an appropriate RG includes 3 sets of basic components (below they are shown on the example of SHS domain):

- a set of elements: they can be defined based on any domain expert interview which has to be divided into some semantic objects to be presented as RG columns, e.g.: *"Smart TV", "Smart light sensors" …*);
- a set of constructs: they have to reflect the alternative variable features of RG elements, and they are placed in the RG rows depending on the appropriate context, e.g. "*Smart TV should be turned on / turned off*"); "*Windows should be opened/ be closed*", *…;*
- a set of possible rating values: to evaluate the behavior of the system under the influence of certain elements the evaluation scale is defined (usually in the range from 1 to 5) where the lowest number of range represents the left bipolar construct, and the highest number of range is the right bipolar construct, e.g. the element "*Smart TV*" *should be turned on* (the value of 1) *or turned off* (the value of 5), depending on a given context.

Now we consider the usage of RG method for the handling of expert knowledge in the SHS development domain (see in section 2), basing on the approach elaborated in [N]. For this purpose, we propose a set-theoretical formalization of the RG – framework, therefore any RG can be represented as a tuple:

$$RG = \{E, C, R\}, \tag{4.1}$$

where *E, C,* and *R* are the set of elements, set of constructs and set of rating values, respectively.

As already mentioned, any RG is a special kind of matrix, and due to its multi-dimensional structure it is possible to represent some different aspects of an initial problem domain. Another advantage of RG method is a good visualization of expert knowledge handling. In the case of SHS development (as a typical social-technical system) these aspects (or dimensions) can be defined as [11]:

- social aspects (SHS has to be a user-friendly solution),
- economical aspect (project expenses should be reasonable for SHS owners),
- environmental aspect (any SHS has to be eco-friendly).

Taking into account these aspects the appropriate problem space (PS) for the conceptualization of user requirements can be specified using RG method, as shown in Fig. 5.

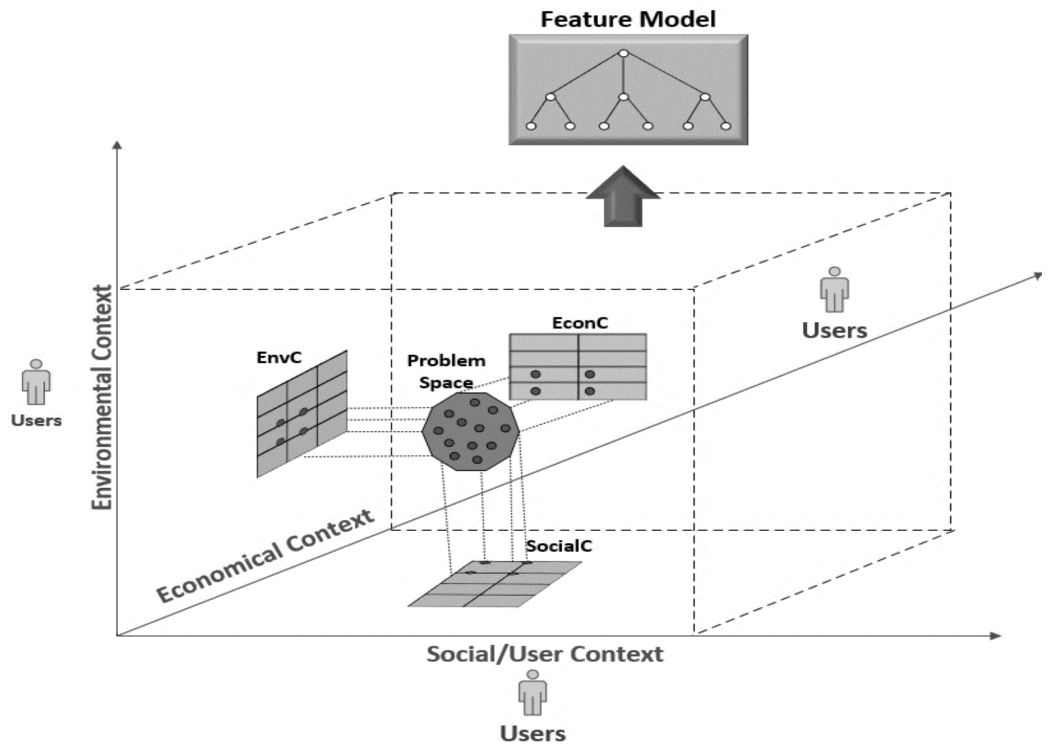

Fig.5 The conceptual scheme of RG method usage [11]

To continue our approach to formalization of the RG method, we can represent this space as a subset of the Cartesian product for the three sets defined in (1), namely:

$$PS(RG) \subseteq E \times C \times R \tag{4.2}$$

At the stage of SHS domain research stakeholders are user experts and the processes of knowledge elicitation, RG creation and complexity analyzation are accompanied by domain experts and knowledge engineers.

Given the fact that the above-mentioned aspects determine when and under what conditions one or another system function(s) should be used, RG can help to identify the impact of the selected function(s) on the behavior of the whole system, to reflect the commonality and variability between them.

From the interviews of domain experts in SHS design in terms of social context it is possible to elicit:

- elements: *«Sleeping», «Not at home», «Leaving house», «Reduce the expenses» …*;
- constructs: *«Temperature should be/doesn't be lower», «Soft music can be/needs not to be played», «Light should be turned on/off», «Windows should be opened/closed»…*;
- rating Values: in range of [1...5].

Fig. 6 represents the created RG based on elicited elements, constructs and rating values from users' interviews.

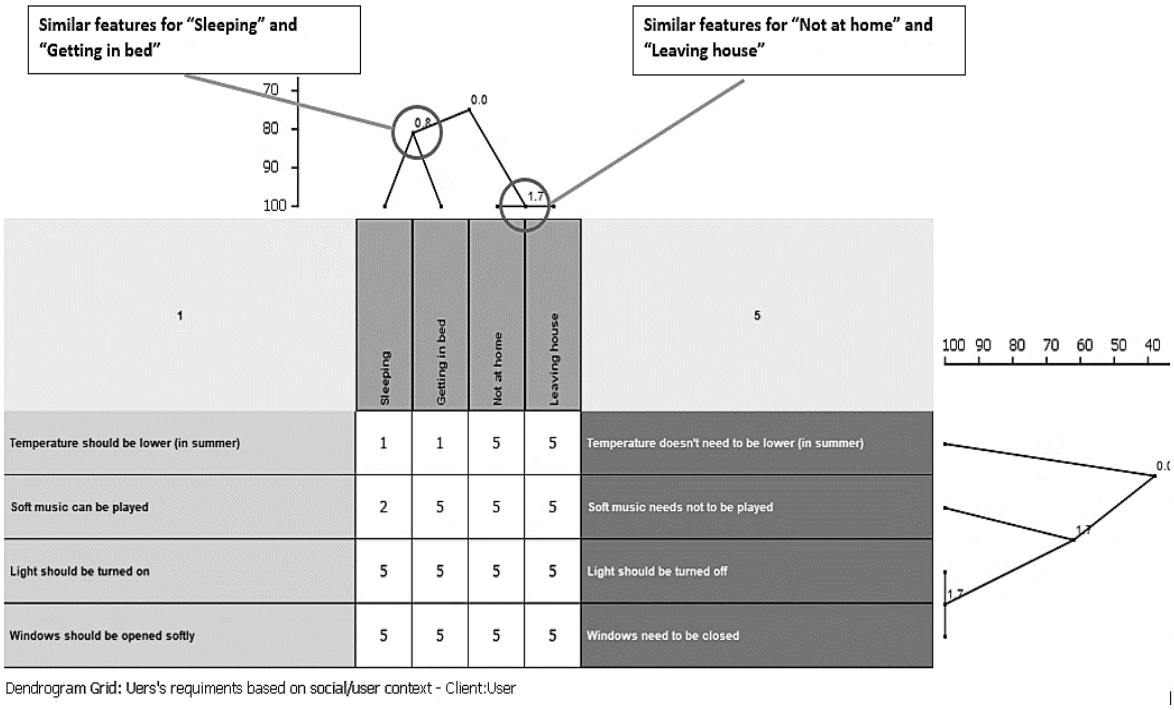

Fig.6 The structured knowledge analysis with usage of RG method

Through the elicited SHS properties and created RG dendrogram it is possible to analyze the common and variable points and reveal the complexity of the system. First, the estimations received from the interviews are analyzed. As shown on the Fig. 6, the construct "Light should be turned on/Light should be turned off" gets 5 for any context. This rating means that for all proposed contexts, function of "turned on light" is useless as it will never be used. The analysis of estimations for other elements occurs similarly. Thus, with RG dendrogram it is possible to determine which functions in the particular context are relevant and which are unnecessary.

Next step of RG dendrogram analysis is detection of common and variable aspects. Based on the given RG, common features will exist for "Sleeping" and "Getting to bed" as they both, due to the context of "Temperature should be lower/ Temperature doesn't need to be lower", "Light should be turned on/Light should be turned off" and "Windows should be opened softly/Windows need to be closed", have the same rating values. Likewise, common features can be identified for "Not at home" and "Leaving house" as both of these contexts have the same rating value for all constructs. The variable point of this particular RG would be the function "Soft music can be played/Soft music needs not to be played" for the social context of "Sleeping", as the exact functionality that should be used during this time is depends on the user attitude or other possible circumstances.

5. The proposed approach to building of a target FODA-model for variability management

To continue our research of the usage of RG method in handling the expert knowledge output for effective variability modeling in SPL development, we propose to consider the requirements specification in the form of a user story (UrS). The UrS is a special form which represents initial user requirements in semi-structured notation [12], where the following syntax is to be used for user needs specification in a natural language:

$$UrS \; => \{As\; a\; < \text{"}User's\; role\text{"} >, I\; want\; < \text{"}Action\text{"} >\; so\; that\; < \text{"}Goal\text{"} >\}, \quad (5.1)$$

where the definitions *"User's role", <"Action">,* and *"Goal"* are the appropriate textual descriptions (usually a simple sentences) given in terms of a target problem domain.

The UrS approach is a recognized way to exactly describe the domain specifics in agile-software development, where possible changes in user requirements must be taken into account. With respect to our target problem domain: SHS development, some possible examples of UrS can be represented as follows:

UrS1: {As a “*SHS Renter”*, I want *“Light to be turned off* “ so that “*I can sleep/leave the house*”},

UrS2: {As a “*SHS Owner”*, I want *“Temperature to be lower”* so that “*I can cut the expenses*”},

UrS3: {As a “*SHS Installer*” I want “*Types of sensors to be less*” so that “*I can reduce time to configurate*”},

………

If now we compare the semantical meanings of the definitions {*“User’s role”, <”Action”>*, and *”Goal”*} used in the general syntax form (3) with respect to their examples in *UrS1, UrS2, UrS3…,* with the components of RG represented in formula (1), it makes clear, that the definition {*“User’s role”*} corresponds to the elements of a set *E*, and the definitions {*<”Action”>*} and {*”Goal”*} match to the elements of a set C respectively. In the formal way it can be defined as two mapping rules ρ and μ accordingly, and represented as:

$$\rho: E \rightarrow \{\text{“}User\text{'}s\ role\text{”}\},\ \mu: C \rightarrow \{<\text{”}Action\text{”}>, \text{”}Goal\text{”}\} \quad (5.2)$$

It has to be mentioned that in recent publications about UrS technique the possibility of their usage for conceptual domain modeling is emphasized [12, 13], and there are already some CASE-tools to support a mapping of UrS to appropriate domain models, e.g. to FODA-models [1,4]. That is why it is possible to consider the sequential usage of the formal definitions for RG and UrS given in (4.1, 4.2) – (5.1, 5.2) as a description of a *structured information base* for the elaboration of a computerized procedure to construct a target FODA-model. The example of such a model for SHS development is shown in Fig. 7, as the structured description of technological variability features to be provided in the target system (in comparison with its initial vision given in Fig. 3).

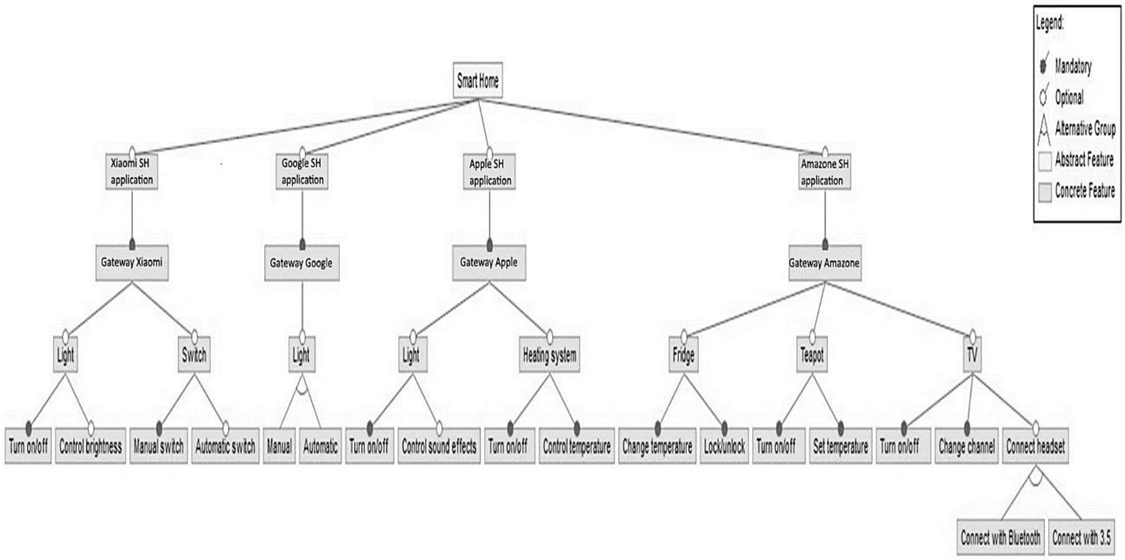

Fig.7 The example of a variability model in FODA-notation for SHS development

To implement an appropriate computerized procedure to automated creation of a target FODA-model using RG methods for expert knowledge handling it is also necessary to apply some CASE-tools for supporting this process [15], and the GridSuite is one of such tools. [16].

6. Conclusions and further work

In this paper the actual scientific and technical problem: the development of software product lines (SPL) with respect to variability issues at all stages of their full life cycle (FLC) is considered. One of the most important stages in FLC is the requirements engineering, and to support this process it is advisable to utilize the methods of expert knowledge elicitation and analysis. The review of some methods to extract and to process of expert knowledge has been carried out, their possible classification has been offered, and the method of repertory grids (RG) has been chosen for the further usage. The essentials of the RG method are considered and its formal definition is elaborated. The examples in the domain of smart house systems using this approach is considered which allows analyzing the technical, social and economic aspects of user requirements for software and hardware variability at the conceptual design stage of these systems. As the result, a possibility to automate a process of a corresponding FODA-model creation that provides an automated variability support at the all FLC stages for SPL has been shown.

For the next step we are going to develop the appropriate information technology to support the proposed approach using some already available CASE-tools and our own software solutions to be elaborated.

REFERENCES

1. K. Pohl, et al: *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer, 2005, 467 pp.
2. E. Evans, *Domain-Driven Development-DDD): Structuring of complex software systems: translation from eng*. – M. Williams, 2011, 448pp. [in Russian]
3. K. M. Lavrishcheva, et al. *New theoretical bases of technology production of software systems families in the context of generating programming: monograph* / In-t program. systems. NAS Ukraine, - K., 2011, pp. 277. [in Ukrainian]
4. M. Tkachuk, R. Gamzaev, I. Martinkus et al. *Towards Effectiveness Assessment of Domain Modelling Methods and Tools in Software Product Lines Development* // Enterprise Modelling and Information Systems Architectures – International Journal of Conceptual Modeling, Vol. 13 (2018), Germany, pp. 190-206.
5. J. Tiihonen, M. Raatikainen, V. Myllärniemi, et al. *Carrying Ideas from Knowledge-based Configuration to Software Product Lines* // Proceedings of the 15th International Conference on Software Reuse, Cyprus, June 5-7, 2016, pp. 55–62.
6. Official Web-site of System Thinking World community, 2004. [Online]. Available: http://www.systems-thinking.org/kmgmt/kmgmt.htm (Last accessed on 12.09.2020).
7. Saeed F, et al. *IoT-based Intelligent Modeling of Smart - Home Environment* // Journal of Sensor and Actuator Networks. 2019; 7(1).
8. R.O. Gamzayev, M.V. Tkachuk, O.T. Tovstokorenko. *Application of domain modeling methods to support software variability in the development of "Smart-Home" systems* // "Information systems and technologies" ICT-2020. Materials of 9th International scientific and technical conference September 17-20, 2020, Kobleve – Kharkiv, Ukraine (included in the conference program) [in Ukrainian]
9. Methods of knowledge acquisition automatization in expert systems: classification, current state, comparative analysis // International scientific-practical journal "Software products and systems". [Online]. Available: http://www.swsys.ru/index.php? page= article &id=1187&lang=.docs. (Last accessed on: 12.09.2020). [in Russian]
10. M. L. G. Shaw, B. R. Gains. *On the relationship between repertory grid and term subsumption knowledge structures: theory, practice and tools.* // Research and Development in Expert Systems, M.A.Bramer & R.W.Milne (Eds), CUP, 1993, pp. 125-143.
11. D. Sangeeta, L. Seok-Won. *From Requirements Elicitation to Variability Analysis Using Repertory Grid: A Cognitive Approach* // Proceeding of RE 2015, Ottawa, ON, Canada, 2015, pp. 46-55.
12. A. Gupta. *Generation of Multiple Conceptual Models from User Stories in Agile* // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, Live Studies Track, and Poster Track, Essen, Germany, 18-03-2019, published at http://ceur-ws.org.
13. R. Mesquita, A. Jaqueira, C. Agra, et al US2StarTool: *Generating i* Models from User Stories* // Proceedings of the Eighth International i* Workshop (ISTAR-2015), CEUR Vol-978.
14. G. Lucassen, F. Dalpiaz, J. Martijn, et al. *Improving Agile Requirements: the Quality User Story Framework and Tool. In: Requirements Engineering*, Vol. 21 (2016), pp. 383–403.
15. A. Curtis, T.M. Well, T. Higbee, et al. A*n Overview and Tutorial of the Repertory Grid Technique in Information Systems Research. In: Communications of the Association for Information Systems (CAIS),* vol. 23(3), pp. 37-62.
16. GridSuite: official We-site of the Repertory Grid Software. 2003-2020. [Online]. Avaliable: https://www.gridsuite.de/45623/41431.html (Last accessed on: 12.09.2020).

ЛІТЕРАТУРА

1. Pohl K., et al: Software Product Line Engineering: Foundations, Principles, and Techniques., Springer, 2005. 467 pp.
2. Эванс Э. Предметно-ориентированное проектирование (Domain-Driven Development-DDD): Структуризация сложных программных систем: пер. с англ. - М. Вильямс, 2011. 448 с.
3. Лавріщева К. М. та ін. Нові теоретичні засади технології виробництва сімейств програмних систем у контексті генерувального програмування : монографія / Ін-т програм. систем. НАН України, - К., 2011. 277 с.
4. Mykola Tkachuk, Rustam Gamzaev, Iryna Martinkus et al. Towards Effectiveness Assessment of Domain Modelling Methods and Tools in Software Product Lines Development // Enterprise Modelling and Information Systems Architectures – International Journal of Conceptual Modeling, Vol. 13 (2018), Germany. pp. 190-206.
5. Tiihonen, J., Raatikainen, M., Myllärniemi, V. et al. Carrying Ideas from Knowledge-based Configuration to Software Product Lines // Proceedings of the 15th International Conference on Software Reuse, Cyprus, June 5-7, 2016, pp. 55–62.
6. Official Web-site of System Thinking World community, 2004. URL: http://www.systems-thinking.org/kmgmt/kmgmt.htm (дата звернення 12.09.2020).
7. Saeed F, et al. IoT-based Intelligent Modeling of Smart - Home Environment // Journal of Sensor and Actuator Networks. 2019; 7(1).
8. Гамзаєв Р.О., Ткачук М.В., Товстокоренко О.Т. Застосування методів доменного моделювання для підтримки варіабельності програмного забезпечення в розробці систем «розумний будинок» // «Інформаційні системи та технології» ІСТ-2020. М А Т Е Р І А Л И 9-ї Міжнародної науково-технічної конференції 17-20 вересня 2020 р. Коблеве-Харків, Україна (включено до програми конференції).
9. Средства автоматизации приобретения знаний в экспертных системах: классификация, современное состояние, сравнительный анализ:[Електронний ресурс] // Международный научно-практический журнал «Программные продукты и системы». URL: http://www.swsys.ru/index.php? page= article &id=1187&lang=.docs. (дата звернення: 12.09.2020).
10. Shaw M.L.G., Gains B.R. On the relationship between repertory grid and term subsumption knowledge structures: theory, practice and tools. // Research and Development in Expert Systems, M.A.Bramer & R.W.Milne (Eds), CUP, 1993, pp. 125-143.
11. Sangeeta Dey, Seok-Won Lee. From Requirements Elicitation to Variability Analysis Using Repertory Grid: A Cognitive Approach // Proceeding of RE 2015, Ottawa, ON, Canada, 2015. pp. 46-55.
12. Gupta, A. Generation of Multiple Conceptual Models from User Stories in Agile // Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, Live Studies Track, and Poster Track, Essen, Germany, 18-03-2019, published at http://ceur-ws.org
13. Mesquita, R., Jaqueira, A., Agra, C. et al US2StarTool: Generating i* Models from User Stories // Proceedings of the Eighth International i* Workshop (ISTAR-2015), CEUR Vol-978.
14. Lucassen, G., Dalpiaz, F., Martijn J. et al. Improving Agile Requirements: the Quality User Story Framework and Tool. In: Requirements Engineering, Vol. 21 (2016), pp. 383–403.
15. Curtis, A., Wells, T.M., Higbee, T. et al. An Overview and Tutorial of the Repertory Grid Technique in Information Systems Research. In: Communications of the Association for Information Systems (CAIS), vol. 23(3), pp. 37-62.
16. GridSuite: official We-site of the Repertory Grid Software. 2003-2020. URL: https://www.gridsuite.de/45623/41431.html (дата звернення: 12.09.2020).

Надійшла у першій редакції 10.06.2020, в останній - 23.08.2020.