MODEL IN PROBLEM SPACE OF A LEGACY SYSTEM FOR THE SUPPORT OF DESIGN AND PRODUCTION OF RUBBER COMPOUNDS

Aleksandar Spasić*

Department of Information and Communication Technologies, Academy of Technical and Educational Vocational Studies Niš, Pirot Section

The fourth generation legacy system used for designing rubber compounds contains valuable information related to rules of business, knowledge, methods and results in compounding. In order to leverage the accumulated knowledge or design new applications related to compounding it is important to define a domain model based on the existing system used with remarkable success in the past twenty years. Due to the lack of both source and binary code, standard software reverse engineering methods could not be applied in this situation. In the process of manufacturing rubber products, the greatest attention is focused on a further understanding and improvement of the development of new rubber compound formulations, design technology and improvement of the existing ones. A basic problem that needs to be resolved is what kind of information is needed about a particular rubber compound. The problem of building an elastomer formulation from the ground up is complex and can easily entail thousands of choices by the compounder. This problem is not a trivial one, and it can only be resolved on the basis of available science-based elastomer technology as well as software engineering methods and models. Modelling in problem space is used as a research method in this paper. Model of problem space (MOPS) deals with creating and understanding of the problem, primarily the problem that a potential user of the system faces. The modelling output transcends both data and code and results in a suite of visual models or diagrams created with use of Unified Modelling Language (UML).

Keywords: rubber compound formulation; model of problem space (MOPS); behavioural description; life cycle model; flow model; unified modelling language (UML).

Introduction

Legacy applications related to rubber compounding described in this are part of “Development”, a subsystem of the large information system (TEGIS) in “Tigar-Technical Rubber Factory” which is used in the Development Department for research and development of new products [1-4]. TEGIS, which consists of 17 databases and more than 70 programme applications, was awarded as a runner-up of most important national IT competition “Diskobolos”.

The primary aim of the Development subsystem was to facilitate the design of new products and to enable the exchange of important information related to the existing ones. During its lifecycle the subsystem dealt with the data related to hundreds of products, including more than two hundred standard and operating formulations. The accumulated knowledge of generations of technologists had been systematised and archived in databases. Our primary motivation in this paper was to find out how the knowledge, information and business intelligence of rubber compounding activities that had accumulated in a legacy system could be preserved. It is assumed that a legacy system can also be a valuable source of information for system analysts and designers of new similar systems, as well as an educational tool for new elastomer technologists.

A legacy system is an existing application system in operation. Although this definition is correct, there is a perception that legacy systems are old and outdated programs. Brodie and Stonebraker in [5, p3] state that, “a legacy information system is any information system that significantly resists modification and evolution”. Usually, these types of legacy systems tend to mirror the life cycle of software development. Software development is usually defined within a framework called “generations.” Most professionals agree that there are five generations which strictly follow the generations of programming languages.

According to this classification, the legacy system discussed in this paper is of the fourth generation. It means that “its programming languages are less procedural than third generation languages. In addition, the symbols are more English-like and emphasize the desired output results rather than how the programming statements need to be written. As a result of this feature, many less technical programmers have learned how to program using fourth-generation languages (4GLs). The most powerful
features of 4GLs include query of databases, code generation, and graphic screen generation abilities. Such languages include Visual Basic, C++, Powerbuilder, Delphi, and many others. Furthermore, 4GLs also include what is known as query languages because they contain English-like questions that are used to directly produce results by directly accessing relational databases. The most popular 4GL query language is Structured Query Language (SQL) [5].

Software Reverse Engineering (SRE) is, by its very nature, the process of analysing the existing applications and database code to create higher-level representations of the code [6].

When a software system is analysed, either in whole or in part, the practice is to extract the information about the design and implementation. Reverse engineering skills are also often used to detect and neutralize viruses and malware, as well as to protect intellectual property [7].

A typical SRE scenario would involve a software module that has worked for years and carries several rules of business in its lines of code. When the source code of the application has been lost, what remains is the “native” or “binary” code.

Unfortunately, the Technical Rubber Factory has undergone several transformations and reorganisations over the last few years. As a result, no code is available to the author, except for screenshots taken a long time ago along with short descriptions of applications. Rebuilding compounding business rules through domain model design is the only solution at the moment.

A compound is a mixture of base polymer(s), fillers and other chemicals which form the finished rubber material. More precisely, the term ‘compound’ refers to a specific blend of ingredients tailored to the specific characteristics needed to optimize the performance of a specific service. Rubber compounding involves the science and engineering of rubbers and rubber additives, such as processing aids, fillers, and curing agents, in specific proportions to obtain a uniform mixture that will have desirable physical and chemical properties to meet processing at low cost and end use performance [8, p.1].

The basis of a compound design is the selection of polymer type. There is a misunderstanding that an elastomer compound is made from a single ingredient and that it is the same for every different type of polymer. The fact is that a rubber compound can be a combination of 3 to more than 20 various elements and thousands of different formations. Reinforcing agents, such as carbon black, coloured pigments, curing or vulcanizing agents, activators, plasticizers, accelerators, anti-oxidants or anti-radiation additives may be added to elastomer by the compounder. There may be hundreds of such combinations. In a typical example there may be four candidate polymer types, ten candidate filler systems, and five types of plasticizer. The total number of combinations of just these three ingredients is 200, without even taking into consideration the variable levels. If only three levels of each candidate material are evaluated, the number of possible formulations is 5400. When the remaining ingredient types and levels are added, the complexity of the problem becomes truly immense [9]. Given such complexity, it is hard to blame the compounder for using the “Edisonian” (hit-or-miss) approach to formulation development that is most commonly used today. Edisonian approach, however, is typically costly and therefore often impractical due to the limited funding of research and development. Rubber compounding is a science of generating desired features and properties by fluctuating components’ ratios and compensating for the interactions. This is the reason why most compounders keep their rubber compositions private. It can take numerous alterations and hours of evaluation to perfect the mechanical features while maintaining good processing.

Experimental

Analyses of legacy system

The purpose of the “Development” subsystem was the fulfilment of explicit requirements of the JUS-ISO 9004 standard in part 10.1- “Planning for controlled production”, section 10.1.1- “Planning of production operations”. This subsystem originally contains 11 software applications that were interconnected horizontally and vertically (Figure 1).

![Figure 1. Subsystem “Development” – main menu](image)

The input data are those from the raw materials database. The database of raw materials is at the first level of filling databases. It contains data on groups and subgroups of raw materials and their codes, as well as data that unambiguously define each raw material used for formulations. These databases also contain data on the specific weights of raw materials and their prices. The raw material base is linked to the raw material specifications database where the required physicochemical properties of raw materials are defined.
Figure 2. The raw materials database working form

The application “Raw materials” (Figure 2) enables the input of data on groups and subgroups of raw materials, as well as the input of data on raw materials themselves. The application enables searches by numerous criteria.

Raw material specifications are documents defined by the Development Department. These are documents that define various physicochemical parameters that a certain raw material must meet in order to be procured on the market and to be used in the production of compounds. Because of their quantity, there was a need to create a database and an application that would manipulate it (Figure 3).

Figure 3. Raw materials specifications

The application “Raw materials” (Figure 2) enables the input of data on groups and subgroups of raw materials, as well as the input of data on raw materials themselves. The application enables searches by numerous criteria.

Raw material specifications are documents defined by the Development Department. These are documents that define various physicochemical parameters that a certain raw material must meet in order to be procured on the market and to be used in the production of compounds. Because of their quantity, there was a need to create a database and an application that would manipulate it (Figure 3).

Figure 3. Raw materials specifications

The “Formulations” application was made with the aim of facilitating the work of compounders with calculations needed to design the compounds and create a knowledge base that would allow a quick and intuitive search for compounds with certain properties.

Two basic types of formulations are defined as follows: standard formulations and production (operating) formulations. The difference is that the share of ingredients in the standard rubber compound formulation is given in phr (parts per hundred of resin) and in the production the formulation is given in kilograms. Multiple production formulations can be created from one standard formulation.

The application enables the entry of the type of formulation, the entry of formulations, the entry of physical and chemical properties of formulations and the definition of aging conditions (Figure 4). It is possible to generate an unlimited number of operating formulations based on a standard one.

Figure 4. Form for data entry related to the standard formulation

The application enables the input and search of rheological properties of compounds, descriptive properties of compounds, mixing regulations, as well as the input of operating formulations (Figure 4). The application allows searching by a large number of criteria, as well as the on-the-fly generation of numerous reports.

The data that uniquely identify standard formulations are their identification numbers. The application will not allow any other input until this information is entered. Data on the type of formulation and the production unit to which the formulation belongs are entered from the list. Data on the specific weight, the total amount and the price of the compound are those that the application calculates using the data from the raw materials database. If any of the ingredients does not have a defined specific weight, which means that the data on the specific weight of the compound will not be accurate, the application will warn about it. It is possible to enter another compound formulation as an ingredient that already exists in the database.

Through the form in Figure 5, the rheological properties of the compound defined before and after aging are defined. Aging conditions are entered from the lists.

Figure 5. Rheological properties of compounds work form
The form for entering descriptive properties of the compound is presented in Figure 6. Descriptive properties of the compound are entered from lists and descriptive grades range from insufficient to excellent.

Figure 6. Descriptive properties of the compound entering form

These data enable associative search of compound formulations according to criteria that depend on previously defined descriptive properties.

The machine which is used for mixing the compound is selected from the list of machines from the machines and tools database. The type of machine is a two-roller or a mixer.

After entering the parameters for machines, the application calculates the weight of the compound batch of the mixture and by choosing the button "Operating formulation" the application converts the formulation into kilograms by using data from the standard formulation.

Since from one standard formulation several operational formulations can be produced, the compounder-technologist may decide to make some changes in the ingredients (for example, the addition of paint) that will not significantly affect the properties of the compound.

Figure 7. The mixing machines parameters entering form

Results and Discussion

Method

Modelling in problem space is used as a research method in this paper. A model is, by its very nature, an abstraction of the reality. Software projects use modelling throughout the entire life cycle. A successful modelling needs to consider the areas in which modelling needs to take place. These modelling spaces are formally considered and discussed in [10]. Three different but related modelling spaces are defined: problem, solution and background. This classification provides a much more robust approach to modelling, as they differentiate the models on the basis of their purpose, i.e. whether a model is primarily created to understand the problem, to provide a solution to the problem, or to influence both of these purposes from the background, depending on organizational constraints and the need to reuse certain components and services. The modelling output in such software projects transcends both data and code and results in a suite of visual models or diagrams.

In the Unified Model Language (UML) projects, model of problem space (MOPS) deals with creating an understanding of the problem, primarily the problem that the potential user of the system faces. Though a business problem is the one usually being described, a technical problem can also be described at the user level in MOPS. In any case, the problem space deals with all the work that takes place in understanding the problem in the context of the software system, before any solution or development is attempted. Typical activities that take place in MOPS include documenting and understanding the requirements, analysing requirements, investigating the problem in detail, and perhaps optional prototyping and understanding the flow of the process within the business. Thus, the problem space would focus entirely on what is happening with the business or the user [11 -13].

The problem space will need the UML diagrams that help the modeller understand the problem without going into technological details. Here, the interest has been shown in the UML diagrams that help express what is expected from the system, rather than how the system will be implemented. These UML diagrams in the problem space are as follows [10]:

- Use Case diagrams—provide the overall view and scope of functionality. The use cases within these diagrams contain the behavioural (or functional) description of the system. It is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

- State machine diagrams—occasionally used for better understanding of the dynamism and behaviour of the problem. It is a significantly enhanced realization of the mathematical concept of a finite automaton in computer science applications as expressed in the Unified Modeling Language (UML) notation.

- Activity diagrams—provide a pictorial representation of the flow anywhere in MOPS. In MOPS, these diagrams work more or less like flowcharts, depicting the flow within the use cases or even showing the dependencies among various use cases.
Behavioural Description of the Process of Compound Formulation

The main objective of a behavioural description is to visualize how the user (represented by the actor) will interact with and use the system. This is done by showing the actor associating with one or more use cases and, additionally, by drawing many use case diagrams.

Modelling a behavioural description of the process of creation formulations for elastomer compounds will be considered in two distinct segments: preparation and creation of formulation.

Preparation

The main actors in the problem space of preparation stage are the Sales Department, the Development Department, the Procurement Department and the Engineering Department.

The use cases important for modelling in the problem space of preparation for compound formulation are as follows: Receive Customer’s Request, Analyse Customer Requirements, Enter Raw Material Data in Database, Enter Raw Material Specification Data in Database and Enter Machines & Tools Specification Data in Database. The Use Case diagram of preparation for compound formulation is shown in Figure 8.

**Use Case “Receive Customer’s Request”**

Short Description: The customer requests information from the Sales Department related to the possibilities of production of new product with given specification.

Actors: Sales Department.

Pre-Conditions: Customer sends an information request.

Post-Conditions: Request or item order is received. Important actors are informed.

Main Flow: (1) The Sales Department receives a customer request related to the possibilities of production of a desired product. (2) The Sales Department forwards the request to the Development Department. Use Case terminates.

**Use Case “Analyse Customer Requirements and Required Raw Materials”**

Short Description: Personnel from the Sales, Development, Procurement and Engineering Department analyse the customer requests and approve/reject the request.

Actors: Sales Department, Development Department, Procurement Department, Engineering Department.

Pre-Conditions: The Sales Department forwarded the customer request to the involved actors.

Post-Conditions: The request is approved and a production order is issued or the request is rejected.

Main Flow: (1) The Development Department analyses production possibilities (production cost, raw material availability, and production capacity and delivery time). (2) The Engineering Department analyses production tools costs, machines availability and their capacities. (3) The Procurement Department analyses the availability of raw materials and production tools. (4) Based on the analyses, the management approves the request and issues a production order [A1]. Use Case terminates.

Alternative flow: (A1) Based on the analyses, the management rejects the request and cancels the production order. Use Case terminates.

**Use Case “Enter Raw Material Data in Database”**

Short Description: Personnel from the Procurement Department assign ID, categories and subcategories and specific weight to the new raw material and enter the data in raw material database. If the raw material already exists in the database, they check the data and the available amount.

Actors: Procurement Department.

Pre-Conditions: Order for procurement of new raw material is issued.

Post-Conditions: Data related to specific raw material are added to database.

Main Flow: (1) The Procurement Department receives an order to procure specific raw material. (2) Data related to the raw material are entered into the database [A1]. Use Case terminates.

Alternative flow: (A1) Personnel check the existing data and the available amount. Use Case terminates.

**Use Case “Enter Raw Material Specification Data in Database”**

Short Description: Personnel from the Input Control Laboratory (part of the Procurement Department) assign specification data to the new raw material and enter the data in raw material database. If the raw material already exists in the database, they check the existing specification data.

Actors: Procurement Department (Input Control Laboratory).

Pre-Conditions: New raw material is acquired.
Post-Conditions: Data related to specification of given raw material are added to the database.

Main Flow: (1) Input Control Lab (part of the Procurement Department) receives an order for entering specification of the given raw material. (2) Specification data related to the raw material are entered into database [A1]. Use Case terminates.

Alternative flow: (A1) Lab personnel check the existing data and specifications. Use Case terminates.

Use Case “Enter Machine and Tools Specification Data in Database”
Short Description: Personnel from the Engineering (mechanical engineers) assign specification data for machine and/or tools and enter the data in database. If the machine already exists in the database, they check the existing specification data.

Actors: Engineering Department (Mechanical Engineer).

Pre-Conditions: New machine is acquired.
Post-Conditions: Data related to specification of the given machine are added to the database.

Main Flow: (1) The Engineering Department (mechanical engineer) receives an order for entering specification of the given machine. (2) Specification data related to the machine are entered into database [A1]. Use Case terminates.

Alternative flow: (A1) Engineer checks the existing data and specifications. Use Case terminates.

Use Case “Design Standard Formulation”
Short Description: Compounder creates the standard compound formulation.

Actors: Compounder (Technologist).

Pre-Conditions: Raw materials and raw materials databases are completed.
Post-Conditions: Standard formulation is designed and stored in the database.

Main Flow: (1) Compounder (technologist) uniquely identifies the standard formulation defining its identification number, and other input is unavailable until this information is entered. Compounder chooses the type of standard formulation (experimental or working one). (2) Compounder chooses from the raw material database the amount of an ingredient (in phr), its group and subgroup and the ingredient itself. [A1] (3) The data on the specific weight, the total amount and the price of the compound are those that the application calculates using the data from the raw materials database. Use Case terminates.

Alternative flow: (A1) Compounder chooses a previously designed compound as an ingredient.

Use Case “Design Operating Formulation”
Short Description: Compounder creates operating compound formulation.

Actors: Compounder (Technologist).

Pre-Conditions: Standard formulation is completed. Mixing Rules are defined.
Post-Conditions: Operating formulation based on one standard formulation is designed and stored in database.

Main Flow: (1) Compounder (technologist) uniquely identifies the operating formulation defining its identification number. Compounder chooses mixing rules for a two-roller [A1] (2) Operating formulation is calculated (ingredient weights) using the data from the standard formulation. Use Case terminates.


Use Case “Design Mixing Rules”
Short Description: Compounder creates mixing rules for the compound.

Actors: Compounder (Technologist).

Pre-Conditions: Standard formulation is completed. The machines and tools database is completed.
Post-Conditions: The mixing rule for a two-roller/mixer machine is created.

Main Flow: (1) Compounder defines the mixing rule for a two-roller machine (batch weight, rollers diameter and rollers length) [A1]. Use Case terminates.


Use Case “Define Rheological Properties of Compounds”

Short Description: Compounder creates rheological properties based on the results obtained from the laboratory.

Actors: Compounder (Technologist), Control Lab.

Pre-Conditions: Standard formulation is completed. Operational formulation is completed and the compound is produced. Control Lab tested rheological properties of the compound.

Post-Conditions: Rheological properties of the given compound are defined and entered in the database.

Main Flow: (1) Compounder defines rheological properties of the compound defined before and after aging. Use Case terminates.

Use Case “Define Descriptive Properties of Compounds”

Short Description: Compounder defines descriptive properties of the compound.

Actors: Compounder (Technologist).

Pre-Conditions: Standard formulation is completed. Operational formulation is completed and the compound is produced. Control Lab tested properties of the compound.

Post-Conditions: Descriptive properties of the given compound are defined and entered in the database.

Main Flow: (1) Compounder defines descriptive properties of the compound. Use Case terminates.

Life Cycle Model of the Compound Formulation

The life cycle model represented by the state machine diagram shows the entire behaviour of the object, as it changes its state in response to the messages it receives. The nature of the state machine diagram is considered to be dynamic-behavioural. “What happens at a certain point in time?” is the question answered by the state machine diagram representing the life cycle of the compound formulation and shown in Figure 10.

Figure 10. State Machine Diagram of the Compound Formulation Life Cycle

Modelling flow of the Compound Formulation

Modelling the flow of a certain process comprises capturing activities made of smaller actions. An action represents a single step within an activity where processing or data manipulation occurs in a modelled system. Activity modelling focuses on the manner of execution and the system behaviour flow, rather than on how it is assembled. Activity diagrams complement the use case diagrams by visually showing the internals of a use case.

Since activity diagrams show multiple threads, they can be used to optimize both business and system processes. This is because a group of processes running in parallel can be captured and modelled through multiple threads. Once they have been modelled, they can be refined and optimized. They also provide an opportunity to
document not only the flow but also the role responsible for that flow.

Activity diagram of the compound formulation flow is presented in Figure 11.

![Figure 11. Activity Diagram of the Compound Formulation](image)

### Conclusion

Creating formulations for rubber compounds is one of the fundamental activities in designing a new rubber product. Activities that make up this development presuppose the possession of multidisciplinary knowledge in various engineering disciplines, such as polymer technology, chemical engineering and mechanical engineering. Also, the already gained experience is a necessary condition for the successful creation of polymer compound formulations.

In order to enable archiving and sharing of knowledge, which has been accumulated for many years in a business system that deals with rubber processing, and in order to facilitate the design of new formulations, it is necessary to create databases and applications that will work with formulations for rubber compounds.

This paper presents a procedure for redesigning a legacy system that has been used successfully for more than twenty years. As the legacy system is a valuable source of data related to business rules, knowledge, methods and results of formulation development, whose source and binary code is not available today, it was necessary to re-design and analyse the system in the problem space. Model of problem space (MOPS) deals with understanding the problem, primarily the problem that the potential user of the system faces. Though a business problem is the one usually being described, a technical problem can also be described at the user level in MOPS.

This analysis is based on a functional specification that is derived from available screenshots, brief descriptions and the author’s memory of how the legacy applications operate.

The current work demonstrates that the UML provides an extensive set of tools to describe the process of designing formulations for rubber compounds, as well as the business model of the compounding. A business analysis of the problem space and behavioural description of a compounding workflow is done by using the use case diagrams. The life cycle of the main content – the compound formulation is defined by the state machine diagram while the flow and dependencies are analysed by using the activity diagrams for the stage of formulation design.

The challenge for the future is to make the model of solution space, as well as the model of background space. Also, the principles proposed in this paper can be applied to other similar technological fields and areas of work.

### References

[8] K.S.Sisanth, M.G.Thomas, J.Abraham, S.Thomas,
Izvod

MODEL U PROSTORU PROBLEMA NASLEĐENOG SISTEMA ZA PODRŠKU PROJEKTOVANJA I PROIZVODNJE GUMENIH SMEŠA

Aleksandar Spasić

Katedra za informaciono-komunikacione tehnologije, Akademija tehničko-vaspitačkih strukovnih studija Niš, odsek u Pirotu

Nasleđeni sistem četvrte generacije koji se koristi za podršku projektovanja receptura gumenih smeša sadrži dragocene informacije povezane sa poslovnim pravilima, znanjem, metodama i rezultatima izrade receptura. Kako bi se iskoristilo akumulirano znanje ili projektovale nove aplikacije vezane za izradu smeša, od značaja je definisati model domena zasnovan na postojećem sistemu koji se koristio sa izuzetnim uspehom u poslednjih dvadeset godina. Zbog nedostatka izvornog i binarnog koda, standardne softverske metode obrnutog inženjeringa u ovoj situaciji nisu mogle da se primenjuju. U proizvodnji proizvoda od gume najviše pažnje je usmereno na daljem razumevanju i unapređenju razvoja novih formulacija gumenih smeša, tehnologiji projektovanja i poboljšanju postojećih receptura. Osnovni problem koji treba rešiti je vrsta informacija koje su potrebne o određenoj gumenoj smeši kako bi se uspešno dizajnirala data receptura. Problem izrade elastomerskih receptura od početka može biti složen i tehnolog može da se suoči sa velikim brojem varijanti. Ovaj problem nije trivijalan i može se rešiti samo na osnovu dostupne naučno zasnovane tehnologije elastomera, kao i metoda i modela softverskog inženjerstva. U ovom radu se kao istraživačka metoda koristi modelovanje u prostoru problema. Model prostora problema (MOPS) bavi se razumevanjem problema, pre svega problema sa kojim se suočava potencijalni korisnik sistema. Izlaz iz modelovanja je hiperarhipski iznad podataka i programskog koda i predstavljen je skupom vizualnih modela ili dijagrama kreiranih upotrebom jezika za objedinjeno modelovanje (UML).