2019 12th International Conference on Computer and Electrical Engineering IOP Publishing
Journal of Physics: Conference Series **1457** (2020) 012012 doi:10.1088/1742-6596/1457/1/012012

Approach to user interfaces development based on semantic model of user activity

S A Belikova, Y I Rogozov, A S Sviridov, O V Shevchenko, A V Egorov, L V Koltunova

Southern Federal University, 1 Engels Street, Taganrog, Russia

corresponding author’s e-mail address: belousova@sfedu.ru

Abstract. In this paper there is proposed a method of user interface design, which is based on semantic model of user activity in the domain, which is formed by the end user himself in natural language, as a source data for design. This description is further transformed into a mechanism of action, the combination of which can be called a model. Also the paper contains the analysis of possible models used in model-oriented approaches to interface development, advantages and features of the proposed model are highlighted. Based on the model in the form of action mechanisms, the user interface of the information system is formed. The paper contains the general example of the process of user interface construction.

1. Introduction

Any system is made of any combination of interacting elements, which are systems themselves. Interacting elements can be people, components and aspects, as well as ideas and concepts, methods and approaches, processes and software, and used elements including computers and network devices, different vehicles and so on. There are many definitions of complex systems but all scientists agree that complex systems perform some functions (with the help of inner elements and interactions between them) to achieve missions and goals. Individual elements themselves cannot achieve the same goals and the whole is greater than the sum of the parts – here we observe the emergency principle of systems. In the lower level of abstraction we observe components and processes out of which the system is constructed, while in the upper level of abstraction we observe the phenomena of the emergency principle. In this paper it is considered that the role of system elements can play the instances of user activity, which form the functionality of the designed system. Elementary actions are the basic elements from which the system is constructed. Such elementary actions are integrated into the final system at the end. Such a final system is heterogeneous, resulting in heterogeneous user interfaces.

The paper presents the method of user interface development the main idea of which is usage of user actions in the domain. The rules of merging these actions into the whole are presented as well.

2. Problematic situations of user interface development

Development of complex systems and user interfaces as its part requires the collaboration between team members (arriving from different communities) with each other and between team members and users themselves. During the interface development process, we meet the difficulties connected with the correct understanding of the transmitted meaning from one team member to another and to the user at most. This situation arises because all design members are from heterogeneous domains, more precisely they think heterogeneously – within different concepts. So, the final interface can not fit the initial idea.

On the other hand, user interface development process is usually organized by a life cycle model describing and guiding activities from the initial idea to the final implementation and performance testing, as for example the waterfall model. The problem with this approach is that it is required correct and complete understanding of the complete user interface design project from the beginning, as correcting a mistake made in a previous phase is a difficult and expensive task. Or similar situation, when requirements to the interface are changing during the system usage, is also difficult and expensive while changing the interface. So, it should be proposed some new approach that can help the customers and developers to spend less time and money to the process of interface correcting.

To overcome these problems, it is suggested approach to user interface development that can help end user and development team to correctly understand each other, and at the same time that can help to simplify the process of managing the interface structure if it is necessary.

Today there is a widely spread approach to use different models for systems and interface design [3, 8, 1, 9]. Let’s consider these approaches and models more precisely.

3. Types of models used during user interfaces design

The proposed method can be attributed to model-oriented, since the description of the user's activity will be transformed into the structure of the mechanisms of user actions [5], which in turn will be the initial data for the interface structure design. In order to formulate the advantages and differences of the model in the form of action mechanisms, it is first necessary to consider what types of models are used in existing model-oriented approaches.

Task Model allows you to provide the structure and description of the tasks (actions) of the user that he can perform in the software system. [1, 3] Such models should reflect the content of user actions in the system: what should he do and why. There are two types of task models [9]. The first type aims to reflect the sequence of tasks and their components, and the second – the data streams that are used when performing tasks. [8]

In this case, the actions that the user performs in the subject area are not taken into account in this model, but only actions that directly relate to interaction with the system are considered. As a model of tasks for visualizing the structure of activity, two groups of methods are used. The first group consists of graphical ways of visualizing the activity process itself, for example, storyboarding, as well as the hierarchical sequence of performed tasks [9]. But this category does not allow to fully reflect the data with which the person works. In addition, such methods do not allow to unambiguously interpret the meaning embedded in them to the person who reads them, because there is no clear methodology for creating and describing such models as storyboards, and the hierarchical structure of tasks also does not allow to understand the full context of the actions. Another group of methods is aimed at reflecting the data streams used in performing tasks. Here there is only an attempt to reflect the process, the main bias is made on the reflection of the data structure used in this process. These methods include infographics, various methods of data presentation, ontology visualization methods [7].

The Dialog Model describes the structure of interaction between the user and the system: the structure of transitions from element to element depending on the action performed. As a model of dialogue, behavioral abstractions are usually used, for example, Petri nets, flowcharts, activity diagrams, UML state sequences. [6]

The Application Model contains the structure of interaction between the system logic and the user interface, as well as the type of data transmitted during this interaction.

Domain Model is a correspondence of the concepts of the domain and the concepts of application logic and interface. Contains concepts, objects, operations, describing the subject area. The form of representation of such models are the essence of the subject area with the attributes, as well as possible relationships and operations on them.

The Presentation Model contains a high-level view of the interface, including what elements the user interface consists of and how these elements should be presented to the user.

The Behavior model has a similar context with the dialogue model. Describes how the user initiates a dialogue with the system, including a description of the input data, the controls used. The Control model contains a list of functions or operations that can be called, as well as preconditions and postconditions of their call. The Environment model contains the cultural aspect of the interaction context. Usually presented in the form of descriptions in natural language. User model most often contains a description of user characteristics, such as level of knowledge, physical and psychological qualities. [8, 1, 9]

None of the above types of models contain explanations of the user's activity in the subject area, they are all at a lower level of abstraction and contain specific options for representing the interface and dialogue, therefore, they are focused on the developer, not on the user. In this case, the user may experience difficulties in checking the compliance of the functionality of the designed system, and the interfaces created in this way may not be sufficiently convenient, which will adversely affect the user performance (the consequences of a semantic gap problem [2, 6, 4]). The use of the model in the form of action mechanisms containing a description of the user's activity in the subject area, regardless of the system being developed, can be categorized as process-oriented approaches, has more opportunities to adapt the interface (and the system as a whole) to possible changes in the subject area and conditions of system usage by changing the description of the activity.

As mentioned above in this paper it is considered that instances of user activity play the role of system elements, and such parts user activity form the functionality of the designed system. Elementary actions are the basic elements from which the system is constructed. Such elementary actions are integrated into the final system at the end and they should be presented in some way. They are presented in the form of mechanisms of actions. Their structure can be found in papers [5] and combination of these mechanisms constitutes the interface model.

4. Semantic model of user activity

The proposed semantic model is a form of organization and presentation of knowledge about the subject area for which the user interface is designed. Semantic model is presented in the form of action mechanism.

The requirements for building a semantic model in the form of an action mechanism are as follows:

- the main functional person should be selected (for example, a doctor, a registry specialist, etc.), on whose behalf the description of the activity will be entered. This is necessary to further divide the interface into functional blocks;
- the description of the activity should begin with a description of large operations, each of which will be represented by a set of interrelated action mechanisms describing the process of performing the operation.

General requirements for the semantic model in the form of action mechanism:

- Completeness. All human actions in the subject area should be described, which should be expressed in the content of mechanisms through specific concepts, entities and relationships.
- Unity means that the content of the action mechanisms should not contradict the actual content of the subject area, namely, the same terms and concepts should be used, so that in future the interface based on this description would not confuse users and make inconveniences for them.
- The inclusion of multi-level concepts implies that the description begins with the identification of general categories (major operations), from which the transition gradually takes place to the specifying concepts and entities. Thus, there is a transition from the method of performing the task to the algorithms.
- Extensibility and flexibility. The structure and content of the action mechanisms can be changed and supplemented at any time in the design process.
- Connectivity implies interdependence and interconnection of action mechanisms through meaningful elements of the mechanism.

The main features of the action mechanism as a domain model are:
- visibility of construction, since the mechanism has a specific structure that determines the content of the mechanism;
- Visibility of the basic properties and relationships. The elements of the action mechanism are meaningfully and logically interconnected, and in general all the mechanisms are interconnected through the elements;
- the availability of a mechanism for research or reproduction, as well as the simplicity of research or reproduction is a very important property that allows the end user along with developers to participate in the creation of a domain model;
- saving the information contained in the original (with the accuracy considered in the construction of the hypothesis model) allows, when detailing, to obtain new information necessary to build the interface project. The structure of action mechanism is presented in fig. 1, the full description of it can be found in [5].

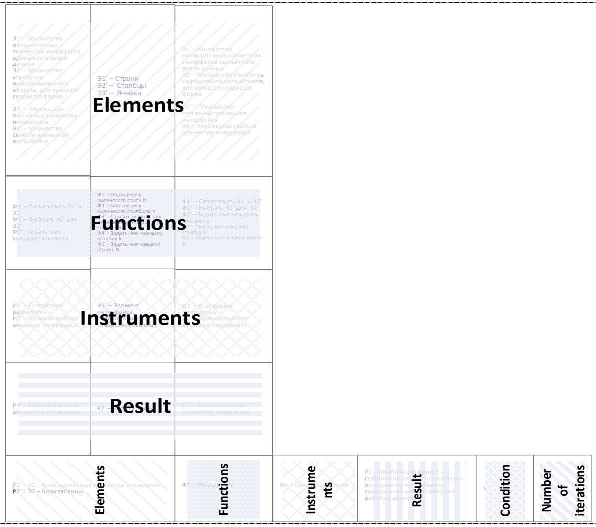


Figure 1. Structure of action mechanism that consists of elements, functions, instruments, conditions, number of iterations and results.

5. Approach to user interface development based on semantic model of user activity

The process of user interface construction begins with the semantic analysis of the description of user activity made by user himself. While analysing we, at first, have to determine the type of user action based on which later the type of form is chosen. The process of determining the type of actions is represented in the fig. 2. Depending on the nature of the user's real activity in performing a certain professional function, a type of form is selected that can assist the user in performing this function. And the structure of this activity influences the filling of the selected interface form type with concrete elements.

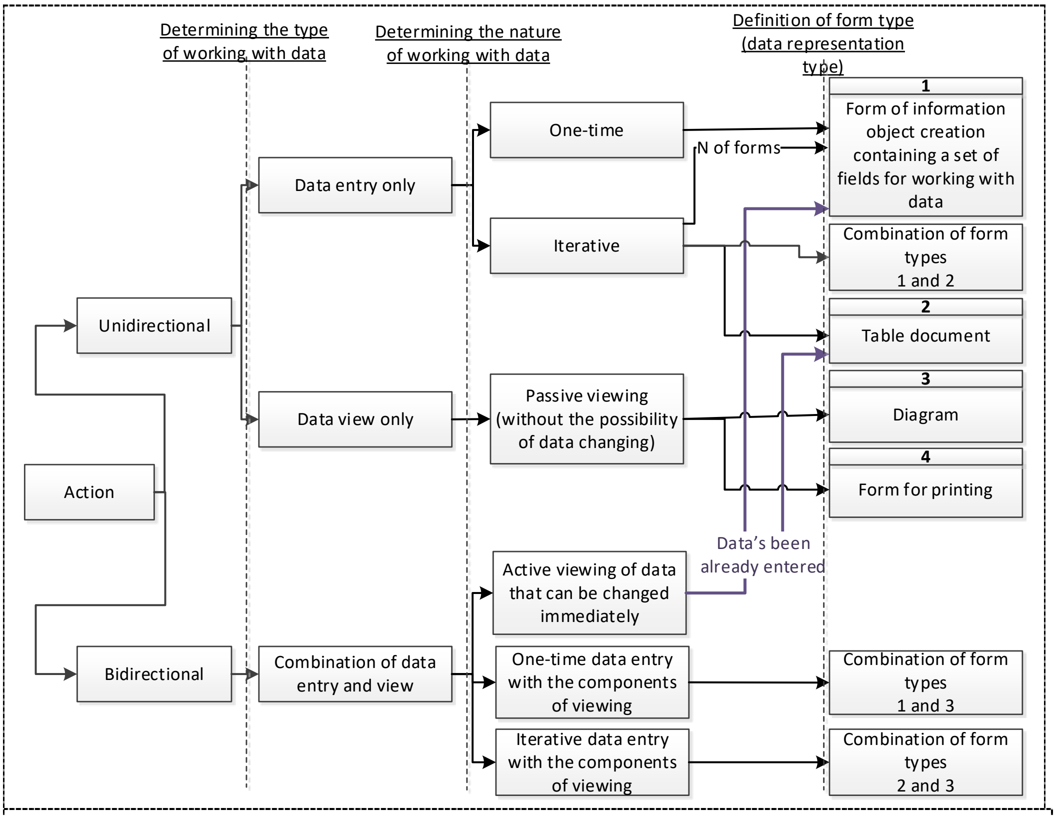

Determining the type of working with data
Determining the nature of working with data
Definition of form type (data representation type)
Action
Unidirectional
Bidirectional
Data entry only
Data view only
Combination of data entry and view
One-time
Iterative
Passive viewing (without the possibility of data changing)
Active viewing of data that can be changed immediately
One-time data entry with the components of viewing
Iterative data entry with the components of viewing
N of forms
Data's been already entered
1
Form of information object creation containing a set of fields for working with data
Combination of form types 1 and 2
2
Table document
3
Diagram
4
Form for printing
Combination of form types 1 and 3
Combination of form types 2 and 3

Figure 2. Sequence of interface form type selection

The main characteristic of the proposed method that helps to manage the interface structure is that if somehow to change the high-level description of actions presented with the mechanisms, it will be possible to get changes in the interface structure.

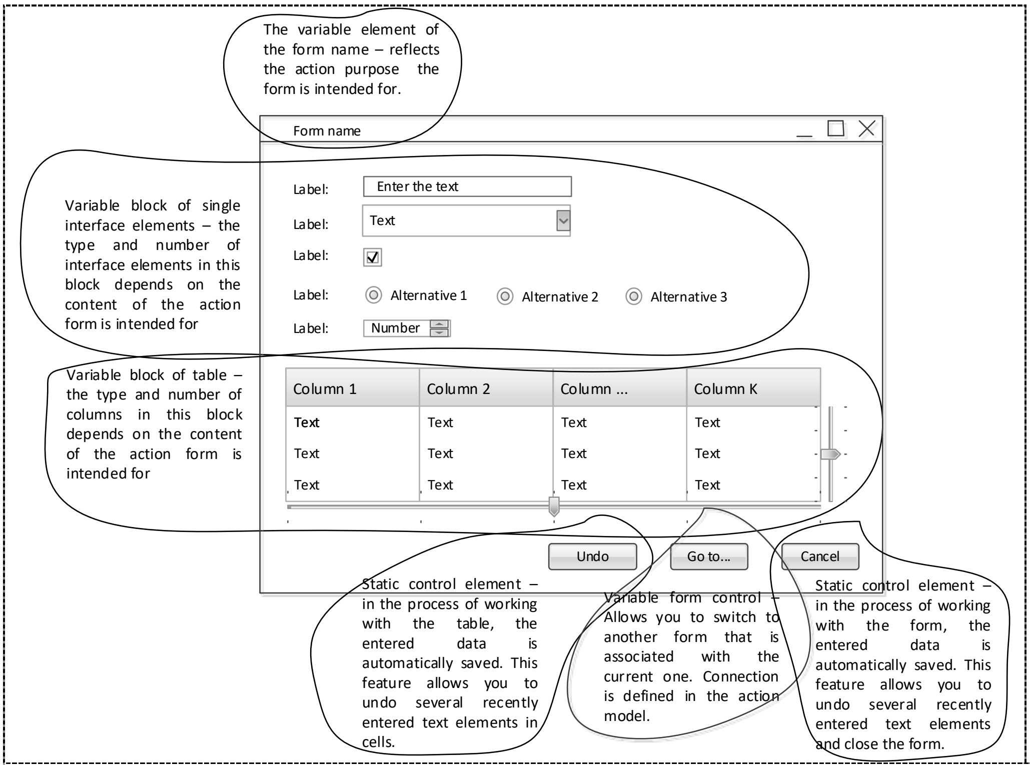
The interface will be a logically related set of interface forms filled with required controls. The communication logic of the interface forms is based on the connection of action mechanisms.

The initial data for the user interface design is the structure of the action mechanisms; the result is a set of related forms filled with related interface elements. Using a set of defined rules, information is extracted from the structure of the action mechanisms and the interface components are obtained.

The rules contain sections for the following interface components: composition of the interface forms, connection of interface forms, composition of the controls in each interface form, communication of controls in each interface form, properties of controls, placement of elements in the interface form, auxiliary controls, connection of controls with the data that will be entered / displayed using this element, connection of the control with the business logic of the system in whole.

The picture 3 shows the example of combination of form types 1 and 2. Some interface elements are static, and some are variable. Static elements are present always in every form of such type. The presence and content of variable interface elements depend on the content of user actions description.

The resulting interface form can be obtained if using the proposed set of rules, that collectively constitute method of user interface design based on semantic model of user activity. To organize the possibility of automated formation of the user interface, it is necessary to follow the rules of interface design based on the structure of the mechanisms of action.


The variable element of the form name – reflects the action purpose the form is intended for.

Variable block of single interface elements – the type and number of interface elements in this block depends on the content of the action form is intended for

Variable block of table – the type and number of columns in this block depends on the content of the action form is intended for

Static control element – in the process of working with the table, the entered data is automatically saved. This feature allows you to undo several recently entered text elements in cells.

Variable form control – Allows you to switch to another form that is associated with the current one. Connection is defined in the action model.

Static control element – in the process of working with the form, the entered data is automatically saved. This feature allows you to undo several recently entered text elements and close the form.

Figure 3. Typical form combining table element and set of single interface elements.

5.1. Determination of the interface forms composition.

Each action corresponds to a separate interface form. If an action serves to obtain some characteristic of another action, then a secondary (subordinate) interface form can be identified for it. However, if:

- there are no elements in action requiring data entry,
- the element is the result of the previous action,
- the action contains the functions performed by the system and requiring activation by the user,

then instead of identifying a new form, a button can be added to activate the action in the form that implements the action that results in the element.

5.2. Formation of the main elements of the interface.

5.2.1. Functions. The formulation of the function begins with a verb. If the function matches any predefined keyword, then suggest an interface element corresponding to this keyword. If there is no one-to-one correspondence between the function (keyword) and the interface element, then suggest a set of interface components that are suitable for this function. If the wording of the function does not match any of the predefined keywords, then it is necessary to prompt the user to add a new function to the keyword list and set the category of the corresponding interface component. Trace the relationship between function and result. If the link exists and the result needs to be displayed, then the control is placed – the button, for example.

5.2.2. Elements. If the element is not defined in the system and is not obtained from another action, then it is necessary to select the data entry element. To select a data entry element, the link should be traced between the function and the element.

5.2.3. Instruments. Check for a match with a predefined keyword. If a match is found and the keyword belongs to the control tools, then use this tool to define the properties of the interface element. If a match is found and the keyword refers to the rules defined outside the system (comparison rules, calculation rules, etc.), then use the instrument to specify the connections between the interface components.

5.2.4. Conditions. Since the action mechanisms are build based on text description of user activity in the domain, if the sentence contains “if”, the action should be performed only if the condition is true.

5.2.5. Number of iterations. Some actions need to be repeated several times, so this characteristic of mechanism should contain how many times the action should be repeated.

5.3. Defining connections between interface elements and between screen forms.

The properties of interface elements need to be made dynamically changeable, that is, to make their value dependent on the data with which the system works, then we get the effect of the data on the interface. Or, if we make the value of the properties dependent on other elements of the interface, we will get a connection between them.

6. Conclusion

The paper presents the method of user interface design, based on the description of the user's activity in the problem domain, which is formed by the end user. The proposed method is a methodological framework for creating user interfaces, the content of which can be objectified in the form of mechanisms of action at various levels. The main quality of the proposed method that helps to manage the interface structure is that if high-level description of actions in the form of mechanisms changes in some way, the interface structure changes as well.

Acknowledgments

The reported study was funded by RFBR according to the research project №17-07-00099 A.

7. References

[1] Vandervelpen C, Vandenbergh J, Luyten K and Coninx K 2004 D5.1 Model-Based User Interface. Development methods for the Design of Context-Driven *Adaptable Interactive Systems* 27
[2] Jonas J M 2018 Stakeholder integration in service innovation – a passive look from outside *Stakeholder Integration in Service Innovation* Springer Gabler, Wiesbaden
[3] Kajiyama T and Satoh S 2014 An interaction model between human and system for intuitive graphical search interface *Knowledge and Information Systems* **39** 41–60 https://doi.org/10.1007/s10115-012-0611-9
[4] Belikov A N, Rogozov Y I and Shevchenko O V 2019 Synthesis of the life cycle stages of information systems development *Advances in Intelligent Systems and Computing* **763** 331–37
[5] Belikova S A, Rogozov Y I and Sviridov A S 2019 Method of user interface design based on semantic approach *Advances in Intelligent Systems and Computing* **763** 311–18
[6] Abelein U and Paech B 2014 State of Practice of User-Developer Communication in Large-Scale. IT Projects. *C. Salinesi, I. van de Weerd (eds) Requirements Engineering: Foundation for Software Quality REFSQ Lecture Notes in Computer Science* **8396** Springer, Cham
[7] Herre H 2010 General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling *R. Poli, M. Healy, A. Kameas Theory and Applications of Ontology: Computer* Applications Springer, Dordrecht
[8] Crystal *et al* 2004 Task analysis and human-computer interaction *Proceedings of the Tenth Americas Conference on Information Systems* New York
[9] Trætteberg H 2002 Model-based User Interface Design 204