An Application of Conceptual Equations in Defining the Wordings of Project Tasks

Petr Sosnin[1]

[1] Ulyanovsk State Technical University, Ulyanovsk 432027, Russia
sosnin@ulstu.ru

Abstract. The paper deals with the use of the design thinking approach in creating the architectural views considering the motivational excitement and goal setting in designing the software intensive systems (SIS). In this case, any architectural view is formed in work with the system of conceptual equations, solving of which is based on the designer's reasoning of the abductive type. In conditions of automated design thinking, such reasoning helps to define wordings of the tasks, constructive relations binding motives, goals, and requirements integrated into the corresponding view. For all views included in the architectural description, these relations are useful to combine, visualize, and interpret as motivationally targeted view demonstrated which architectural decisions correspond to the intended goals.

Keywords: Architectural Modeling, Design Thinking, Goal, Software Intensive System, Viewpoint, Task Wording.

1 Introduction

Designing of any software Intensive system (SIS) is a unique process of operative combining the numerous and diverse project tasks, the greater part of which is unpredicted to be arising in the course of the corresponding project. For any of unpredicted tasks, it will need to build the wording considering the context of its appearing. Moreover, the wording of the unpredicted task (or new task) should be coordinated with the wordings of other tasks defined before. In such a work, it needs the order that is better to coordinate with the stepwise refinement process of work, beginning with the initial wording of the main task of the corresponding project. During this process, step by step, it will be formed levels of the grouped tasks.

Among these levels, the especial place occupies the first level of architectural tasks $\{Z_i^A\}$ that are targeted on understandable descriptions of the SIS from the viewpoints of diverse groups of stakeholders involved in the design process. When this level is built and tested, the obtained result as an integrated architectural description AD will play the role of a conceptual version of the SIS, reflecting its understanding as a

In: P. Sosnin, V. Maklaev, E. Sosnina (eds.): Proceedings of the IS-2019 Conference, Ulyanovsk, Russia, 24-27 September 2019, published at http://ceur-ws.org

wholeness, which will be demanded by stakeholders at all stages of developing the SIS. Therefore, coordinated wording of a set of architectural tasks is very important.

In this paper, we propose a way for coordinated wordings of a set of tasks $\{Z_i^A\}$ in conditions of its becoming So, this way is responsible for the initial wording of the main task of the project and subordinated tasks that are included in the project at the first level of the stepwise refinement process of the SIS architecture. So, this process captures a part of the conceptual design phase when tasks of the set $\{Z_i^A\}$ are solved only conceptually and only for their coordinated wording. Any conceptual solution of the task Z_i^A includes textual and graphical components that are combined in the whole expressed the intended understanding.

We objectify the proposed way in the instrumentally modeling environment WIQA (Working In Questions and Answers) that supports conceptual designing the SISs [1].

2 Preliminary bases of coordinating the architectural views

Professionally mature development of any modern SIS is unthinkable without the mandatory construction and operational use of an artifact AD. As a sample of verified structures and embedded understanding, AD plays a managerial role, providing the correspondence between this sample and the current state of the SIS in the design process. Should also be highlighted, this version is the first (earliest) representation of SIS as a wholeness, which can be tested to detect dangerous semantic errors.

Marked positives of AD were the reasons for intensive accumulating the experience of architectural modeling that was generalized in several standards among which it should be noted the standard ISO / IEC / IEEE 42010: 2011. This standard assumes that the AD is the system $S(\{V_j\})$ of “architectural views” $\{V_j\}$, based on corresponding “viewpoints” $\{VP_k\}$, each of which specifies “the conventions (such as notations, languages, and types of models) for constructing a certain kind of view. That viewpoint can be applied to many systems. Each view is one such application” [2].

Thus, in designing the certain SIS, at the conceptual stage of the work, responsible designers (fulfilling the role of the architect) must choose an appropriate set of views $\{V_j\}$ and develop them so, that they form a coordinated system $S(\{V_j\})$. Moreover, the main purpose of designers’ activity is to determine the basic requirements for the SIS and to express them in a visually understandable form.

Such work should be rationally organized and fulfilled. That is why, in designing the SIS, it needs to separate a subset of architectural tasks $\{Z_i^A\}$ among all tasks of the project. Then tasks of this subset should be solved so that they will be transformed into the system of tasks $S(\{Z_i^A\})$ with obviously indicated features as the certain system.

Since each of the architectural views is a systemic formation then the main features of the system $S(\{Z_i^A\})$ should be defined at the meta-level of the subset $\{Z_i^A\}$. For designers, the features of meta-level are better to represent so that they will have the managerial potential for developing the system S({Vj}). This case of organizing and creating the AD will lead to the necessity of searching the rational ways for defining of meta-level means that provide effective coordination of architectural views.

In our deep conviction, the coordination steps should be rationally intertwined with other actions of stepwise refinement in the process of solving the architectural tasks $\{Z_i^A\}$. Moreover, firstly, it needs to achieve coordinating of their wordings in conditions of sufficient substantiations.

This conviction is caused by our more the twenty years experience of studying the stepwise refinement in processes of solving the project tasks at the conceptual stage of designing the SISs. Main directions of this study are shown in figure 1, where it is highlighted that basic actions of the stepwise refinement were conducted using question-answer reasoning (QA-reasoning).

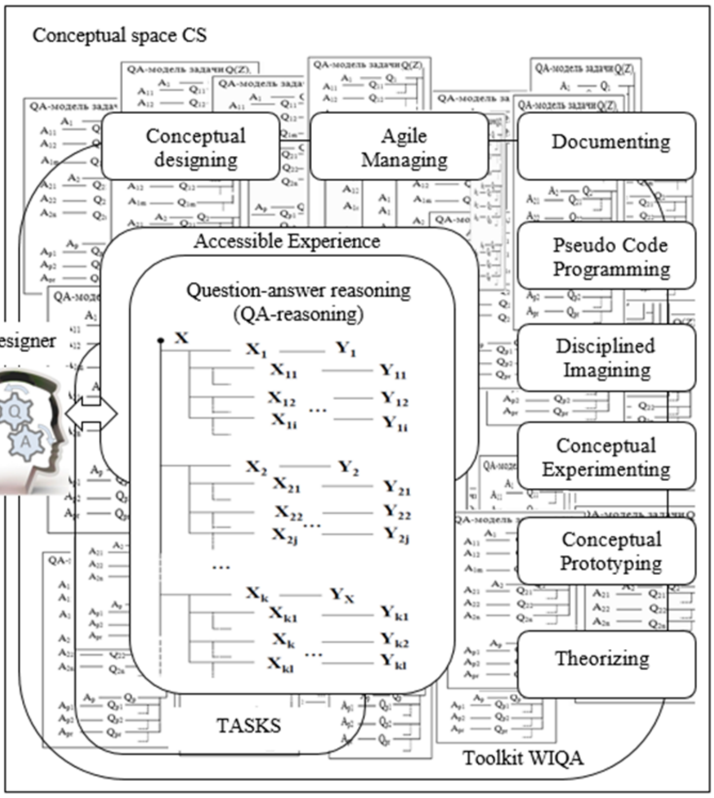

Fig. 1. Basic applications of QA-nets

In this scheme, one can view a visual tree of nodes, any of which has X or Y-type, but nodes can have not only hierarchical relations. They can be bound by common additional attributes modeling useful semantic relations. Thus, in the general case, the structure can have the form of the question-answer net (QA-net). In applications indicated on the scheme, QA-nets were used for modeling the different essences of the operational space of designing the SIS in their reflection on QA-memory of the toolkit WIQA. Moreover, units of these nets were interpreted as verbal “traces” of interactions of designers with an accessible experience and its models. Note that the features of our research are described in detail in monograph [3].

In any domain of our research, we developed the specialized means that (as subsystems) were embedded into the toolkit WIQA. In designing them, we applied the architectural approach, one application of which is presented in [4].

3 Conceptual equations

Let us return to the problem of coordinating the architectural views. Firstly, we focus on architectural modeling as a kind of activity in which the designers most fully manifest the features of design thinking. To underline these features, for the typical reasoning used by designers, Dorst [5] suggests using the expression

$$\frac{WHAT}{(thing)} + \frac{HOW}{(working\ principle)}\ leads\ \frac{RESULT}{(observed)} \tag{1}$$

which can be fitted on problematic situations arisen in designer's work. In investigated design situations, a researcher should define and insert adequate "constant' and/or 'variables' in this expression after which it can be interpreted as a specific 'equation.'

Such interpretation helped to Dorst in analyzing the following kinds of equations:

- DEDUCTION $WHAT\ +\ HOW\ leads\ ???$
- INDUCTION $WHAT + ???\ leads$ RESULT
- ABDUCTION 1 $???\ + HOW\ leads$ RESULT
- ABDUCTION 2 $???\ + ???\ leads$ RESULT

According to the conducted analysis, Dorst concluded that abductive types of reasoning define the nature of designing as a human activity in problematic situations. Moreover, for design practice, the most typical reasoning corresponds to conceptual equations of the fourth kind, which require the use of iterative reducing the uncertainties of 'unknowns' till their appropriate 'values.'

Such a way opens up the possibility for the reasoning of designers even in conditions

$$\frac{WHAT???}{(thing)} + \frac{HOW\ ???}{(working\ principle)}\ leads\ \frac{RESULT\ ???}{(observed)} \tag{2}$$

in which the designer iteratively sets intermediate values for two unknowns, calculating the third unknown in the conceptual equation to be solved.

Considering equations of all kinds are formed in different situations for diverse essences corresponded components of equations, but, in any case, they reflect cause-and-effect inferences that are intertwined in the design process. Among these inferences, solving the abductive equations occupy the principal place.

Let us assume that we chose conceptual equations for creating the initial wordings of project tasks. Such a decision requires determining the mechanisms that designers will use for working with similar equations. In our deep conviction, the role of these mechanisms can fulfill an appropriate version of implementing the design thinking approach (DT-approach).

Our conviction is caused by the following features of the DT-approach:

1. Steps of its application include wording for the task to be solved
2. For the potential solution of the task, the step of prototyping focuses on physical and conceptually algorithmic feasibility that must be considered in the wording.
3. The designer has the right to reformulating the task if there is a reason, for example, a necessity of coordinating with wordings of other tasks.

For such work, we developed and many times used our version of automated design thinking approach [4], the scheme of which is presented in figure 1.

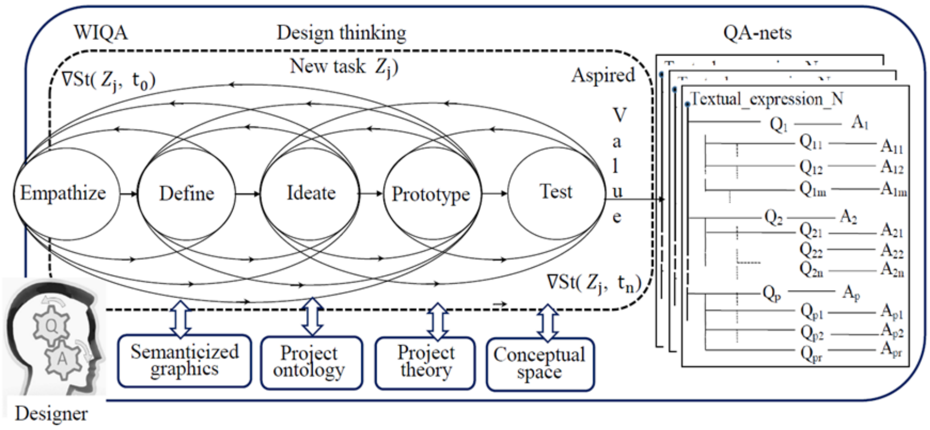

Fig. 2. The iterative process of design thinking

The scheme indicates that process of design thinking is implemented in the instrumental environment WIQA intended for conceptual designing the SIS in conditions when the designer can use such artifacts as Project ontology, Project theory and Conceptual space that are created in the course of designing in parallel with other designers' activity. The main attention of actions in the frame of the scheme focused on *formulating* the statement of the task and also on *representing, moving, evaluating, and managing* components of reasoning in forms of question-answer analysis. All these actions described in detail in the paper [4]. Additionally, actions include building the diagrammatic representation of the view (for example. the view V_j) and prototyping the solution of the task (for example, the task Z_j). So, the developed version of the DT-approach is the constructive way for solving the of conceptual equations that are considered above.

4 Features of Goal Setting in Developing the Architectural Description

As told above, in designing the SIS, wordings of the architectural tasks occupy the principal place. Namely, in the wording of these tasks, designers must define the basic goals and requirements of the SIS to be developed. In these actions, especially for an innovative project, designers try to express an arisen intention about achieving the conceived value. This intention implicitly concerns reasoning about the motives and goals of the project and their formulation or, by other words, it concerns the work, for example, with the following equation

$$\frac{Motives??}{(essences)} + \frac{HOW\ ?}{(goal\ setting)}\ leads\ \frac{Goals\ ???}{(intended)}$$

in which the quantity of question signs indicates on the different degree of uncertainty.

It should be noted, intentions, motives, and goals are typical concerns in architectural modeling, that prompt to bind with such modeling an appropriate view or views and a viewpoint or viewpoints. This paper devotes to defining of them, but, previously, we want to disclose some our positions that are based on the Vityaev's research [6], the content of which defines relations among motives and goals in the frame of the corresponding task and results of its solution.

In this publication based on the theory of functional systems (TFS, developed by P. Anokhin [7]), E. Vityaev describes features of the goal-directed behavior in conditions when a person is trying to solve the certain task. The discussed kind of behavior includes the following specific features:

1. Intrinsic and/or extrinsic stimulus (perceived by the person) lead to a motivational excitement that activates mechanisms of goal setting.
2. An initial definition of the intended goal "does not imply knowledge of how by what means and when it can be attained."
3. The goal cannot be attained without having a criterion of its attainment.
4. "Between the concepts of goal and result, the following relationship holds: the result is obtained when the goal is attained, and the criterion of its availability is "triggered." But when the goal is being set, we have the goal but not the result."
5. Extending the description of the goal by conditions in which the expected result can be (possible hypothetically) obtained, leads to a description of what is called the "task.

Below, these features and described understanding of the design nature will be taken into account in our reasoning that will concern the work with project tasks that help to create the architecture of the SIS.

5 Related Works.

This section, we start with clarifying the kinds of actions that the designer has to perform in design thinking in work with any new task. To do this, we will return to the study of Dorst who notes the set of such actions includes *formulating, representing, moving, evaluating and managing* of components of reasoning in the conceptual space of the corresponding project [3]. Noting these actions, he means that they are performed in conditions of solving the equations among which the abductive equations occupy a fundamental place.

Another observation of Dorst [8] is the start of the project with an abductive equation

$$\frac{WHAT?}{(thing)} + \frac{HOW\ leads\ to\ VALUE}{(frame)} \quad (4)$$

in conditions when designers have experience of designing that has similarity with a conceived project, and they can try to inherit the structural relationships between HOW and VALUE. For such inheritance, Dorst introduces the concept "framing," putting into it the following content "Framing is the key to design abduction. This is because the most logical way to approach a design problem is to work backward, as it were: starting from the only "known" in the equation, the desired value, and then adopting or developing a frame that is new to the problem situation." Similar cases also admit for the iterative search of the appropriate solutions, but at the level of reframing.

The important role of abductive reasoning in the design synthesis is discussed in [9] where its authors analyze the use of abductive transfers from a design intent to design targets, and to new design conceptions also. Additionally, they underline the possibility of an abductive diagnosis of violating design constraints or design axioms. The useful collection of patterns of abduction reasoning and inferences is considered and formalized in the paper [10].

Indicated type of equations suggests that their conceptual solutions are defined with the use of mechanisms of design thinking. Therefore, among related works, we mark publications that disclose some features of these mechanisms. Overview comparison of models and mechanisms at micro-, meso- and macro-levels of designing is conducted in the paper [11]. Several important mechanisms consider in the paper [12] that focuses on the evolutionary side of the explanatory design.

A very important group of related works includes papers devoted to relations between motives and goals. In the understanding of these relations, we follow the Goal Setting Theory Goal[13], some motivational aspect of which is constructively evolved in [14]. We took into account the way of the motivational design described in the papers [15] and [16].

All papers indicated in this section were used as sources of requirements in developing the set of means included in the motivational and goal-oriented viewpoint applied for building the motivationally targeted view on the SIS to be designed.

6 Goal-Oriented Architecture Modeling

The main intention of architectural modeling is to understand the SIS to be developed as a whole and as early as possible. Understanding is a naturally artificial process and its result, artificial part of which includes figuratively semantic scheme and coordinated verbal description. Strategic result of the necessary understanding is formed as an integrated complex of architectural views, each of which V_j can be considered as a result of solving the corresponding architectural task Z_j.

In this section, we present our version of attaining the necessary understanding that clarifies motivationally targeted view on designing the SIS. In the general case, work with any task Z_j begins with the conceptual equation

$$\frac{Motive\ (Z_J)??}{(essences)} + \frac{HOW1\big(Z_J\big)???}{(goal\ setting)}\ leads\ \frac{Goal\ (Z_J)?}{(intended)} \tag{5}$$

where, for simplifying the reasoning, we suppose that work with task Z_j is caused by one Motive(Z_j) and one Goal (Z_j).
Any goal is meaningful if there is a criterion whose application allows anybody to assess attaining the goal. Therefore, it needs to extend the equation (4) by adding the equation

$$\frac{Criterion\ (Goal\ (Z_J)??}{(essences)} + \frac{HOW2\big(Z_J\big)???}{(checking\ the\ attainment)}\ leads\ \frac{Goal\ (Z_J)?}{(intended)} \tag{6}$$

In its turn, needed checking the Goal (Zj) is possible if the task Zj is solved and the way of the solution is implemented as a minimum in conceptually-prototype version because only it opens the possibility for cause-and-effect understanding the solution of the task Zj. Thus, it needs to add one more equation

$$\frac{TaskZ_J??}{(essences)} + \frac{HOW3\big(Z_J\big)???}{(solving\ the\ task)}\ leads\ \frac{Result\ (Z_J)?}{(obtained)} \tag{7}$$

which together with the equations (3) and (4) forms a system of conceptual equations.

Note, components of these equations have conceptual nature, for example, component “Task Z_j” presents “wording of the task Z_j” in its current state. By other words, this component is a dynamic construction that is formed on the course of solving the indicated system of equations or conceptual solving the task Z_j.

In the process of solving, other components of the system of equations are also dynamic constructs that are formed in this process that leads to building the view V_j corresponding to the obtained Result(Z_j). The built view V_j integrates several important requirements $R_j=\{R_{jk}\}$, the composition of which correspond to the construct “Criterion (Goal(Z_j)).” Thus, the built view V_j can be presented by the following symbolic expression

$$V_j= S_j(Motive(Z_j)), Goal(Z_j), Criterion(Z_j), Result(Z_j)). \tag{8}$$

where any component of the expression is the conceptual object created in the course of solving the system of equation with the use of design thinking.

Let us assume that, in solving the main task Z* of the certain project of the SIS, for any view V_j of their system S({Vj}), appropriate conceptual equations were built and solved, in the result of which designers created the system of the following artifacts

$$\left.\begin{aligned} &V_1 = S_1(Motive(Z_1)), Goal(Z_1), Criterion(Z_j), Result(Z_1)) \\ &V_2 = S_2(Motive(Z_2)), Goal(Z_2), Criterion(Z_2), Result(Z_2)) \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \\ &V_J = S_J(Motive(Z_J)), Goal(Z_J), Criterion(Z_J), Result(Z_J)) \end{aligned}\right\} \quad (9)$$

where the set $\{Z_j\}$ includes architectural tasks $\{Z_j\}$ subordinated to the task Z*.

In our publication [4], we offered to bind with this system the integrated architectural view that we called "motivationally targeted view" (or shortly, MT-view) on the SIS. In the general case, the diagrammatic representation of the MT-view will have the structure that conditionally shown in figure 2.

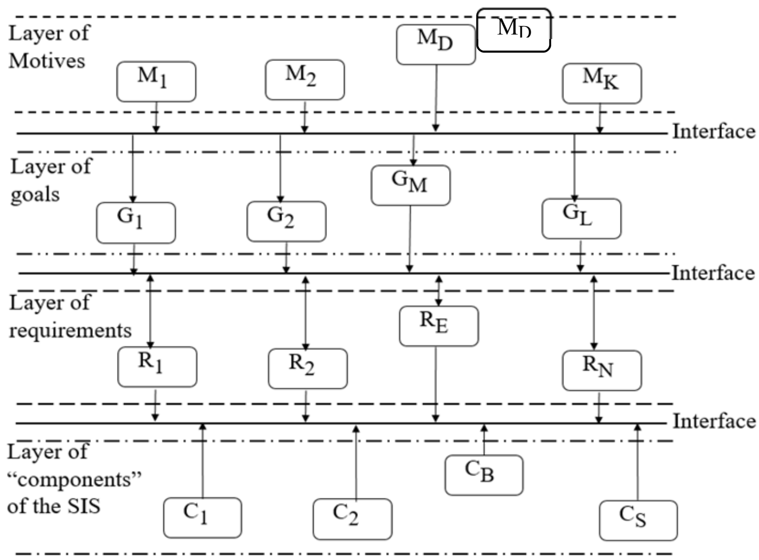

Fig. 3. Generalized structure of the motivationally targeted view

Any layer in this scheme presents components conditionally and in general. In the real MT-views, components of layers are combined in the net or tree. A similar way is used for combining the layers (in figure 2, this role is fulfilled by interfaces). What is especially important, after solving of all conceptual equations, components of any layer are strictly specified and expressed in understandable forms.

For the layer of motives, this means that the motivational design is implemented: internal and external motives are divided into groups; dominant motive M_D is highlighted; it is established cause-and-effects relations among motives and goals. In a set of goals, it is highlighted the main goal of G_M or their subset, and all relations of the subordinated types are defined and marked.

The layer of requirements (integrated into the AD) indicates, firstly, which criterions must be used for checking the attainment of goals defined in the layer above, and secondly, in which components of the SIS any requirement must be objectified. Thus, the motivationally targeted view reflects physical and conceptually algorithmic feasibility of the SIS that to be designed in accordance with initial intentions.

7 Life Cycle of the MT-view and the Corresponding Task Wording

Let us assume that for the main task Z*, the designer bound two views V_1 and V_2, for any of which the corresponding system of conceptual equation was built. In this case, it needs to decide how it will be solved the following system of systems

$$\left.\begin{array}{c} Motive(Z_1) + HOW1(Z_1)\ leads\ Goal\ (Z_1) \\ Criterion\ (Goal\ (Z_1) + HOW2(Z_1)\ leads\ Goal\ (Z_1) \\ Task(Z_1) + HOW3(Z_1)\ leads\ Rezult\ (Z_1) \end{array}\right\} \quad (10)$$

$$\left.\begin{array}{c} Motive(Z_2) + HOW1(Z_1)\ leads\ Goal\ (Z_1) \\ Criterion\ (Goal\ (Z_2) + HOW2(Z_2)\ leads\ Goal\ (Z_2) \\ Task(Z_2) + HOW3(Z_2)\ leads\ Rezult\ (Z_{2,}) \end{array}\right\} \quad (11)$$

If views V1 and V2 are independent, then equations (10) and (11) can be solved separately, but even in this case, they must be combined in the wholeness at least at the level of program coding.

When views are dependent, the designer should solve equations in coordination and such coordination will be defined by kinds of relations between tasks Z_1 and Z_2, for example, relations between Goals or Criterions or Results. Thus, wordings for both of the task should be formed in parallel by switching between them. Moreover, verbal traces of such relations must find its expression in the wording of the main task Z* obtained in solving the following system of equations

$$\left.\begin{array}{c} Motive(Z^*) + HOW1(Z^*)\ leads\ Goal\ ((Z^*)) \\ Criterion\ (Goal\ ((Z^*)) + HOW2\big((Z^*)\big)\ leads\ Goal\ ((Z^*)) \\ Task(Z^*) + HOW3(Z^*)\ leads\ Rezult\ (Z^*) \end{array}\right\} \quad (12)$$

That is why, constructive design thinking is the appropriate way for wordings of all tasks indicated above, but for coordination of them, the architect needs switching between “lines” of working with systems of equations (10), (11) and (12), between “lines” of applying the mechanisms of design thinking as it is generally shown in Fig. 4.

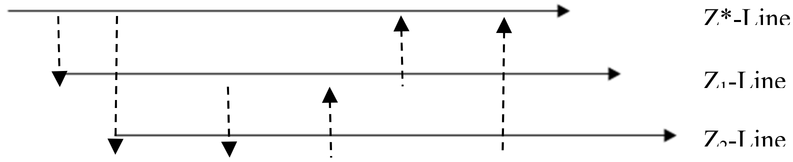

Fig. 4. Lines of generating the wordings of the tasks

As a result, will be a time, when, in the life cycle of these lines, for any task will be formed its initial wording. In the general case, the reasons for the switching can be caused by the necessity of changing the built initial wordings that will wordings, and it will require for rational change management.

Let us note the described feature of coordinating should be considered in building any group or groups of the dependent views. Therefore, designers should discover relations among views as early as possible and include their verbal traces in initial wordings of the corresponding tasks.

8 An Example of Coordinated Wording of the Project Tasks

For wordings and corresponding architectural views, it is principal to provide their understanding. Suppose that a decision has been made to include in the toolkit WIQA a subsystem that provides an understanding of the architectural description of the SIS to be designed. By other words, there is an intention to build (and embed in the toolkit WIQA) the subsystem that supports understanding of designers and other stakeholders in the course of developing the architectural description in the process of designing the certain SIS.

As told above, to evolve such intention till the state of the wordings of a set of tasks defining the architectural views of the subsystem it needs to apply the appropriate version of the DT-approach, but in any version, the first step of working requires to formulate the important requirement of the potential users to this subsystem.

The role of this requirement can fulfill the following discourse:

> **D1(Z*).** ***The subsystem must provide to embed the appropriate means in conceived components of the architectural description of the SIS to be designed, and these means must activate and increase the effectiveness of the natural mechanisms of understanding in interactions of stakeholders with these components.***

It needs to mark that words used in the discourse D1 must find their reification (materialization) in the process of designing or its results. Therefore, in the WIQA-environment, their values should be defined in the constructive form with the use of QA-analysis and reasoning. During such work, it will be revealed and clarified the following moments:

1. Natural understanding is based on the mental activity of the right and left hemispheres of the human brains.

2. By the nature of designing, this activity concerns intellectual guessing, mental imagery, mental experimenting, dialog processes of consciousness and their registering outside of brains (firs of all, in the computerized environment).
3. In this activity, understanding is responsible for the coordination of figurative and verbal forms both inside and outside of brains.
4. Means of objectifying the understanding outside of the brain must support registering and perceiving the architectural and cause-and-effect forms of understanding.

Question-answer processing these clarifying, estimating their physical and conceptually algorithmic feasibilities has led us to the following initial wording of Z*-type task for the discussed subsystem:

> ***Z*. It needs to develop the subsystem that provides to embed the appropriate means activating and increasing the effectiveness of the natural mechanisms of understanding in interactions of stakeholders in conceived components of the architectural description of the SIS.***
>
> ***By using the ontological and figuratively semantic maintenance, the subsystem must support registering outside of the brain and perceiving by the stakeholders the architectural and cause-and-effect forms of understanding in real-time interactions with them.***
>
> ***The subsystem must be embedded in the toolkit WIQA so that interactions with the accessible experience will be the base of all activities applied in the constructive work with understanding.***

Stepwise refining this task has led us to subordinated tasks Z_1 aimed at the ontological support of understanding, and the task Z_2, solving of which provides figuratively semantic maintenance of processes indicated in the wording of the task Z*.

The decision to apply the ontological maintenance was caused by the necessity of the controlled use of the project language, and especially in cases when understanding is very obliged.

Such a requirement correspnds to the following discourse:

> ***D1(Z_1). For the correct and controlled formation of the project language, embedding semantics in the generated text and graphical constructs, as well as for detecting and preventing semantic errors in them, it is necessary to form and use the project ontology.***

As for the discourse D1(Z*), the QA-analysis of the text D1(Z_1) helps to clarify the following details for words used in this discourse:

1. Ontological structuring of the surrounding is the base of human perception.
2. Concepts (notions) that are accumulated in the project ontology can be interpreted as a source of normative units helped in "measuring" or expressing the semantic values of textual and graphical constructs.
3. Program forms of concepts help in automated achieving the coordination between verbal and figurative descriptions of described situations or events.

Question-answer analysis of the discourse D1(Z_1) and marked details open the possibility for formulating the following wording of the task Z_1:

Z_1. It needs to develop the complex of means that is responsible for the ontological maintenance of conceptual action, including the processes of understanding in situations and events when it will be necessary.

The functional potential of this complex must be oriented on an appointment of the appropriate concepts (and their names) to the applied verbal and graphical constructs, discovering and preventing the semantic errors in the descriptions of constructs

Ontological and figuratively semantic maintenances must be complementary in conceptual solving of the project tasks.

More details, we will disclose the work with the system of conceptual equations (11) describing the figuratively semantic maintenance in processes of understanding. Necessary actions were conceived for supporting the mental imagination in conceptual experimentation applied in design thinking [4]. By other words, in this process, mental imagination is a source of intellectual "guessing" generated by the right hemisphere of the brain, and any result of such abductive guessing is necessary to register in the graphical form reflected its structure as a wholeness. That is why means of the semanticized graphics (SG) can help the designer in solving the conceptual equations (11). In their solving, we will orient on the building the MT-view, the generalized scheme of which was presented in figure 3.

Developing of these means is governed by the following motive:

M_D. In design thinking, explicit graphical registering the mental imagery will facilitate to increase the effectiveness of understanding activated when it is necessary for the designer.

Among possible ways of registering the appeared mental imagery, we chose those which is based on appropriate forms of programming that help to imitate the process of creating the result of registering ant its testing. Therefore, it was intended to realize the SG as a graphical editor oriented on creating the block-and-line schemes with features described just above. It should be noted that any of these schemes can be interpreted as an appropriate view of what the designer needs to understand.

This decision was expressed by the following main goal:

G_M. Means of the SG must help the designer to register the process and result of any act of understanding in observable and programmable forms that confirm the wholeness of the built block-and line scheme.

Understanding is a naturally artificial process and its result, as told just above, can be implemented with the use of appropriated views any of which (similarly architectural views) should be presented by the corresponding block-and-line scheme with a coordinated verbal description. Moreover, according to the goal G_M, any view on understanding must be accessible the designer in the appropriate program form. Such intention has led us to the following subordinated sub-goals:

G_1. For supporting the architectural forms of understanding, it is rational to apply the pictorial programming that helps to bind any block-and-line scheme with the program of its drawing for the repeatable activating the process and result of such form of understanding.

G_2. To provide the iterative coordination between the graphical side of the view and its verbal description, it is rational to apply the declarative programming (more exactly, Prolog-like description) that helps to discover errors and inconsistencies for their correcting in graphical and symbolic components of the artificial side of understanding.

G_3. To express cause-effect manifestations of understanding, it is rational to apply the model-driven approach to programming oriented on the use of UML.

Goals are defined correctly only if there are criteria for testing their achievement. As stated above, the role of such criteria can be assigned to requirements whose fulfillment confirms that the intended effects are observed. In real practice, some requirements can be reasons while effects are consequences.

In the considered case, awaited effects must be bound with a constructive expression of achieving the necessary understanding, the result of which is registered by the set of chosen views (block-and-line schemes and their descriptions). In developing the SG, achieving the necessary effects were caused by the following essential requirements:

R_E. To provide the programmable effects in the graphical components of views on understanding, it is necessary to extend the pseudocode language embedded in the WIQA toolkit so that it will support pictorial, declarative, and model-driven forms of programming.

R_1. For any element placed in the workspace of the graphical editor, means of pictorial programming must provide assigning an indexed link, the activation of which helps to switch to another scheme or call a program unit or switch to a specific application outside the WIQA toolkit.

R_2. For any textual unit and its semantic net, declarative programming must provide their automated iterative coordination with using the transition to Prolog-like description for its subsequent checking on errors and inconsistencies.

R_3. To objectify an algorithmic side of understanding that embeds in UML diagrams, it needs to provide their model-driven transformation to program codes helping to discover semantic errors and demonstrate dynamics of the corresponding process.

In the course of formulating the declared motive, goals, and requirements it was built the following statement of the task Z_2:

Z_2. It needs to develop the complex of means that provides creating the diagrammatic schemes of architectural views and other useful diagrammatic schemes in understandable forms.

In conceptual designing, the certain SIS, the functional potential of this complex must be oriented on supporting such intellectual activities as guessing, mental imagination, and conceptual experimenting in solving the project tasks.

Figuratively semantic maintenance and ontological support must be complementary in conceptual solving of the project tasks.

The described conceptual solution of the task Z_2 corresponds to the MT-view, the simplest scheme of which is shown in figure 5.

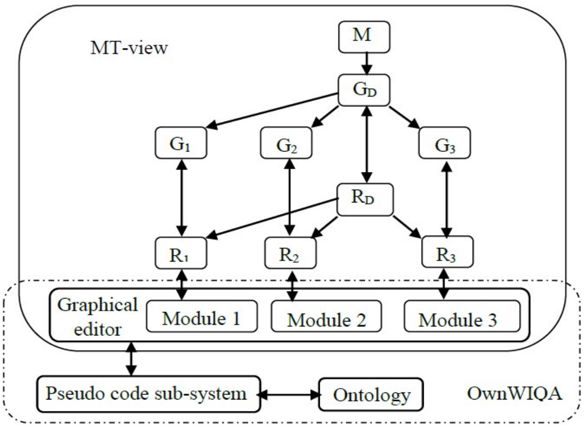

Fig. 5. Example of motivationally targeted view

The choice of the simplest scheme is caused by demonstrating aims of this section. The SG was developed two years ago, and the final version of its MT-view was richer.

In the version, that is presented in figure 5; it is reflected only the kernel of the SG – its three modules providing the basic modes of creating and using the graphical editor. Each of these modes is responsible for the materialization of the corresponding requirement.

Additionally, the view indicates its dependence from the project ontology. Any label on the scheme that is drawn in the editor and any notion applied in the scheme description must be checked on the correspondence of the project language, and, therefore, on the correspondence of the project ontology. Furthermore, the module 2 is responsible for logical checking the applied textual units after their transformations in Prolog-like descriptions that must be corresponded the project ontology also.

The scheme in figure 3 presents not only the example of the MT-view but also the basic components of the motivationally and goal-oriented viewpoint for developing any MT-views. More detailly, this viewpoint includes means for implementing the following actions:

- Preparation to drawing the MT-view (automated design thinking for solving the conceptual equations);
- Visualization for using the MY-view (graphical editor, project ontology, a complex of means for pseudo code programming, the library of program utilities for the SG-transformations and transitions).

If these actions are included in conceptual designing the certain SIS, they will facilitate to usefully objectifying the motivational and goal-oriented basis of the corresponding project in the visualized and understandable form.

9 Conclusion

Appearing the new project tasks is the primary reason for the uniqueness of any designing process. Among these tasks, architectural tasks occupy the principal place. They are not only new; their wordings must be coordinated. For creating the initial states of wording and providing necessary relations among them, we suggest using our version of automated design thinking. Moreover, above we applied to steps of design thinking for generation several wordings that focus on a concern of understanding

In the paper, we also demonstrate the purposefulness of including the MT-views in the practice of architectural modeling the SIS. In several real projects, we have tested our version of automated design thinking in its application for building the MT-views. This version is implemented in the instrumental environment WIQA intended for conceptual designing the SIS in conditions when the designer can use means of the Semanticized Graphics (SG-complex) and such artifacts as Project ontology, Project theory, and Conceptual space that are created on the course of designing in parallel with other designers' activity.

The paper includes the example of the MT-view that corresponds to the development of the SG-complex, extending the potential of the WIQA toolkit. This component is intended for the architectural expression of understanding when it is necessary. Such necessity is especially important when it needs to create the architectural descriptions or, by other words, when it needs to create a system $S(\{V_j\})$ of architectural views. Any of this view expresses the embedded understanding. MT-view is one of them.

References

1. Sosnin P.: Experience-Based Human-Computer Interactions: Emerging Research and Opportunities", IGI-Global, (2017).
2. Standard ISO / IEC / IEEE 42010: 2011, Available at https://www.iso.org/standard/50508.html
3. Sosnin, P.: Automated Design Thinking Oriented on Innovatively Personified Projects Communications in Computer and Information Science, 920, pp. 312-323, (2018).
4. Sosnin P., Shumilov S., Ivasev A.: An Architectural Approach to the Precedent-Oriented Solution of Tasks in Designing a Software Intensive System. Lecture Notes in Computer Science, vol 11623. Springer, Cham, (2019)
5. Dorst, K.: The Nature of Design Thinking, in DTRS8 Interpreting Design Thinking, In Proceeding of Design Thinking Research Symposium, pp. 131–139, (2010).
6. Vityaev, E.E.: Purposefulness as a Principle of Brain Activity. In: Nadin M. (eds) Anticipation: Learning from the Past. Cognitive Systems Monographs, Vol. 25. Springer, Cham, pp. 231-254. (2015).
7. Sudakov, K.V.: The theory of functional systems: General postulates and principles of a dynamic organization. Integrative Physiological and Behavioral Science, Vol. 32(4), pp. 392-414, (1997)
8. Dorst, K.: The core of 'design thinking' and its application, Design Studies Vol. 32 (6), pp. 521-532, (2011).

9. Lu, S. C.-Y., Liu, A.: Abductive reasoning for design synthesis, USACIRP Annals - Manufacturing Technology 61, pp. 143–146, (2012).
10. Schurz, G.: Patterns of abduction Volume 164 (2), pp. 201–234, (2008).
11. Wynn, D.C., Clarkson, P.J.: Process models in design and development. Research in Engineering Design. Vol. 29(2), pp.161–202, (2018).
12. Nguyen, L., Lang, D., van Gessel, N., A. K. Beike, A. K., Menges, A., Reski, R., Roth-Nebelsick, A.: Evolutionary Processes as Models for Exploratory Design. Biomimetic Research for Architecture and Building Construction. Biologically-Inspired Systems, Vol. 8. Springer, Cham, pp. 295-318, (2016).
13. Yurtkoru, E. S., Bozkurt, T., Bektas, F., Ahmed, M. J., Kola, V.: (Application of goal setting theory. PressAcademia Procedia (PAP), Vol. .3, p.796-801, (2017).
14. Locke, E. & Latham, G. (2006). New directions in goal-setting theory, Association for Psychological Science, vol. 15, no. 5, pp. 265-268, (2011).
15. Lunenburg, F.P: Goal-setting theory of motivation, International Journal of Management, Business, and Administration vol. 15, pp. 1-6.
16. Keller, J.M.: Motivational Design for Learning and Performance The ARCS Model Approach, New York: Springer, (2010).

∗ This work was supported by the Russian Fund for Basic Research (RFBR), Grant #18-07-00989a, Grant # 18-47-73001r-a, and the State Contract №2.1534.2017/4.6