



TOOLS FOR CONCEPTUAL-ALGORITHMIC PROTOTYPING OF PROJECT SOLUTIONS IN SOFTWARE INTENSIVE SYSTEMS DESIGN

Ali Hamzah Obaid ¹, Khansaa Azeez Obayes Al-Husseini ²

^{*1,2} Babylon Technical Institute, Al-Furat Al-Awsat Technical University, 51015 Babylon, Iraq

Abstract:

This article describes the development of a design tool that provides the ability to manage quickly create prototypes of design solutions, including graphical elements and software components that are connected together. At the same time, this tool should be universal and suitable for use in various projects.

Keywords: Software Intensive Systems (SIS); Information Technology (IT); Working in The Question-And-Answer (WIQA); Question-Answer (QA).

Cite This Article: Ali Hamzah Obaid, and Khansaa Azeez Obayes Al-Husseini. (2018). "TOOLS FOR CONCEPTUAL-ALGORITHMIC PROTOTYPING OF PROJECT SOLUTIONS IN SOFTWARE INTENSIVE SYSTEMS DESIGN." *International Journal of Engineering Technologies and Management Research*, 5(10), 117-122. DOI: 10.5281/zenodo.1491903.

1. Introduction

In the process of designing modern software-intensive systems (SIS), it is often necessary to make experiments, choosing the most suitable variant of design solutions from several possible ones. To solve the problem of choosing the most suitable design solution, an approach called prototyping is used.

This approach leads to including some tasks such as developed prototypes management, prototype testing and choosing in developers' workflows. A number of time resources needed for design process depend on project tasks solving effectiveness. Consequently, design success depends on the effectiveness of prototyping tasks solving. In this way, there is a need for an instrument that provides an effective solution for prototyping tasks. This paper describes the software solution providing prototyping support including visual modeling of prototype schemes and providing integration with existing design support tools in the question-answering environment. This allows both executions of prototype solutions in order to assess their effectiveness, and including work related to solving prototyping tasks in the overall design workflow.

2. Design Support Tools

As it was said in the introduction to this work, the prototyping facilities are designed taking into account their orientation to the question-answer environment. Such environment is the WIQA design environment, developed at the Department of "Computer Science" of Ulyanovsk State Technical University.

Own WIQA is the software complex including question-answering (QA) memory and a set of design tools, that uses it. As part of the Own WIQA environment, there is pseudo-code interpreter for LWIQA language. Also, it contains a graphical editor using which the developer can create interfaces in the form of Block-and-line schemes. This editor allows the developer to bind various pseudo-code programs and implement a mechanism for selecting a user of one of them. Since the components of the software being developed are pseudo-code programs, this graphics editor is an appropriate mechanism for implementing the interfaces of the prototyping tools being developed. WIQA tools include projects in the form of sets of QA-models. Also, it presents developers groups in the form of organization structure. A task in the form of QA-model presents its detailed content allows use it as the task solving workflow. In the workflow building process, there is an association between design tasks and designers appears.

In this case, the front of the team's work is formed. Each participant in the project process receives a set of tasks to solve a certain period of time, which form the task queue for the solution. This capability allows the implementation of prototype solutions as WIQA tasks. Also, the WIQA environment includes a tool for graphical visualization and editing of diagrams, which involves communication with the question-answer implementation of tasks. This tool allows implementing a graphical representation of prototypes in the WIQA environment.

3. Task of Prototyping Tools Design

The prototype of a design task solution is a verifiable composition of parts of this solution that are interconnected by suitable interfaces. Executable prototypes should be programmed and executed in the search mode to control the actions enclosed in them. Suppose that a prototype is necessary to check the procedure for which there is already an algorithmic idea of the solution, and a detailed algorithmic procedure scheme and program code is missing. In this case, to make sure that the idea is right (it is correct) and it is necessary to develop and test the prototype with the algorithmic scheme that corresponds to the idea. So, in the proposed approach, prototyping is understood as the creation of a pseudo-code for a certain algorithmic solution and its execution to verify the correspondence of a conceived idea.

The central place in the prototyping system is occupied by prototypes. The prototype of a design task solution is a verifiable composition of parts of this solution that are interconnected by suitable interfaces. Executable prototypes should be programmed and executed in search mode to control the actions enclosed in them.

On the basis of the decisions made in the previous section, generalizing them, we formulate the problem of development prototyping system.

- 1) To develop a tool complex for creating and exploring prototypes for preliminary verification of project design solutions in order to prevent semantic errors.
- 2) In developing the structure and functionality of the complex, use methods and tools for conceptual design and prototyping, oriented to the executable pseudo-code, supplemented by extensions of the C# language.
- 3) The created complex should provide expansion of tool-technological environment WIQA.

4. Architecture of Prototyping System

The prototyping system is a set of components that run in a WIQA environment. Since one of the most important features of this system is its close integration with the WIQA design environment, the most effective way to present information about project risks is to map them to WIQA question-answering memory. The prototypes set of the design solution is a graphical diagram including elements that are references to the implementation of prototypes.

The next figure shows the architectural model of the prototyping system:

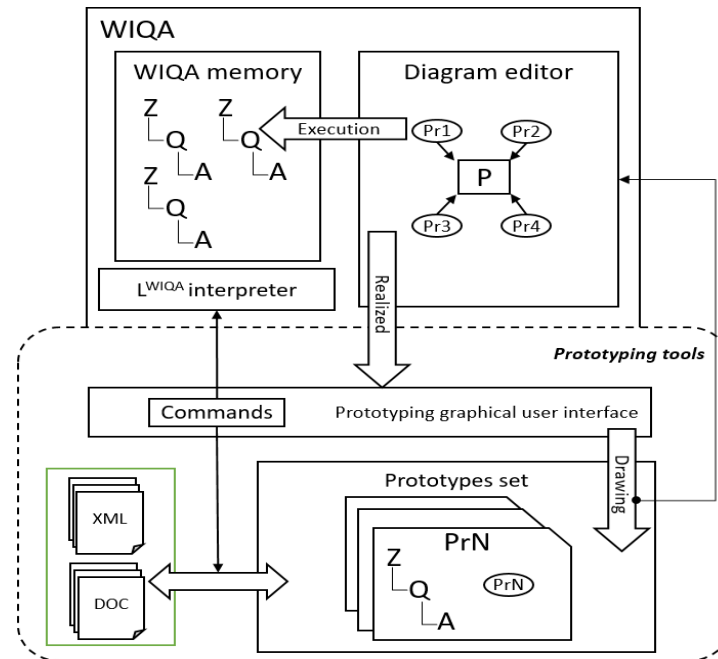


Figure 1: Architectural model of prototyping system

In accordance with this model, the prototyping work is performed in accordance with the following procedure:

- 1) The developer builds prototypes using the language LWIQA.
- 2) Using the chart editor built into the WIQA environment, the developer adds their graphic image, establishing a connection between it and the pseudo-code implementation of the prototype.
- 3) The person responsible for checking the prototypes opens the graphic representation of the prototypes of the project, selects the desired prototype on it and proceeds to its execution in the interpreter of the pseudo-code programming language.
- 4) After checking the prototypes, the participant performing the design process makes a mark on the diagram about the suitability of using a particular prototype.

An important feature of the implementation of graphical user interfaces in the WIQA diagram editor is the absence in this editor of the separation of the interface editing mode and the execution mode. Thus, the user gets the ability to use already functioning UI elements in one editor mode, and at the same time to edit it, add new graphic primitives to the page. The following figure shows a diagram of the interface forms of the prototyping system.

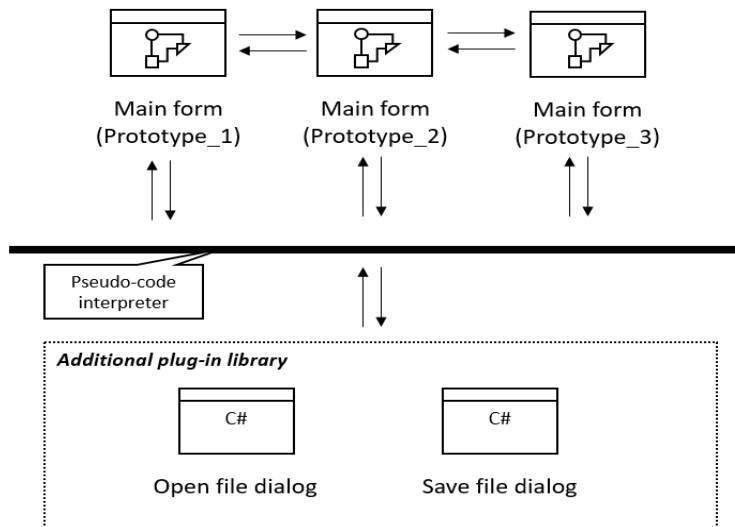


Figure 2: Diagram of interface forms of prototyping system

The following figure shows the template of the main prototype form:

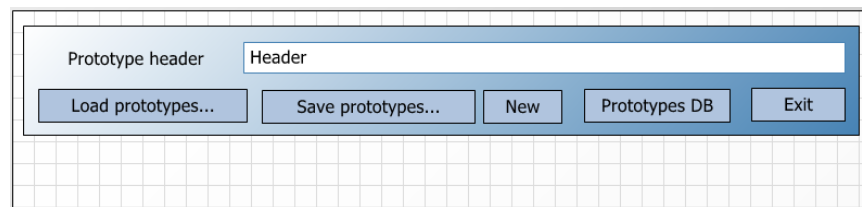


Figure 3: Main user interface of prototyping system

This interface contains a number of buttons, for each of which a pseudo-code handler is implemented. This handler is executed as a task stored in the query and response memory of the project. This view allows you to integrate prototyping tools into any WIQA project by importing these pseudo-code views from an XML file.

These pseudo-code handlers are implemented as tasks in the WIQA question-response memory and implement the basic functionality of the prototyping system. Consider, for example, the procedure for loading prototypes "Load prototypes." Figure 4 shows this procedure in the question-answer tree:

```

8.2.4. Загрузить прототипы
├─ Q 8.2.4.2. &diagfile& := OPENFILEDLG("Файлы диаграмм с прототипами (*.nsp)|*.nsp|Все файлы (*.*)|*.*", "nsp")
├─ Q 8.2.4.3. &eventfile& := OPENFILEDLG("Файлы связи (*.xml)|*.xml|Все файлы (*.*)|*.*", "xml")
├─ Q 8.2.4.5. DD_Load(&diagfile&)
├─ Q 8.2.4.6. DD_LoadEvents(&eventfile&)
└─ Q 8.2.4.7. FINISH
    
```

Figure 4: «Load prototype» pseudo-code procedure

5. Prototyping Methodic

Since the development of the prototyping system is focused on its implementation in a WIQA environment, the main mechanism of expansion of which is pseudo-code programming, the most

advantageous representation of the algorithms of its components is the presentation of them in the form of techniques that can easily be described as pseudocode. The prototyping system from the position of the methodologies was divided into two main parts - the methods of working with the prototype as such and the methodic of working with the library of prototypes.

As an example, consider the methodic of creating / editing a prototype by the user. The following is a fragment of it:

1. If the prototype is ready, complete editing, proceed to the methodic of preserving the prototype
2. If it is needed to create a pseudo-code prototype function
 - 2.1. Create a pseudo-code implementation prototype function
 - 2.2. Debug the pseudo-code implementation of the prototype function
 - 2.3. In the prototype graphical interface, add a graphic element for the new function
 - 2.4. Draw the relationship of this element with other elements
- ...

6. Conclusion

In the design process, developers have to carry out solutions to design tasks, for many of which there are several possible solutions, from which it is not always possible to choose the most appropriate option in advance. To select the most appropriate solution, developers build a set of prototypes, from which they subsequently select the most appropriate option by analyzing the performance of a particular prototype. The task of prototyping is one of those tasks that have to be solved in the design process, and therefore the efficiency of the design process depends on the effectiveness of their solution. The implementation of prototyping software makes it possible to make the prototyping process manageable, thereby increasing both the efficiency of working on prototypes and the effectiveness of all the work of designers.

References

- [1] Ali Hamzah Obaid. Development of a system of tools for conceptual-algorithmic prototyping in solving design problems in the development of systems with software. Master's dissertation, Ulyanovsk State Technical University, 2017.
- [2] Obaid, Ali Hamzah, Tools for conceptual-algorithmic prototyping in solving design problems in the development of systems with software, Interactive Systems: Problems of Human-Computer Interaction Ulyanovsk: USTU, 2017,25-27 P.
- [3] Obaid, Ali Hamzah. Information hiding techniques for steganography and digital watermarking. UDC 681.518 (04) INTERACTIVE SY STEMS: Problems of Human-Computer Interaction. – Collection of scientific papers. - Ulyanovsk: USTU, 2015.- 306 p. 2015
- [4] Al-Husseini, Khansaa Azeez Obayes. Risk Management Tools in The Design of Automated Systems. INTERACTIVE SYSTEMS: Problems of Human - Computer Interaction. – Collection of scientific papers. – Ulyanovsk: USTU, 2017. – 290 p.
- [5] Khansaa Azeez Obayes Al-Husseini. Methods and Means of Risk Management in the Design of Automated Systems. Master's dissertation, UISTU, 2017.
- [6] Khansaa Azeez Obayes Al-Husseini, Information security in the field of technical development and information, Interactive Systems: Problems of Human-Computer Interaction Ulyanovsk: USTU, 71-80p.2015.

- [7] Khansaa Azeez Obayes Al-Husseini / Ali Hamzah Obaid, Development of Risk Management Tools in Question-Answering Based Software Design Environment, International Journal of Computer Science and Mobile Computing - IJCSMC, Vol. 7, Issue. 6, June 2018, pg.165 – 174.
- [8] Sosnin P.I. Pseudo-code control of work flows in the design of automated systems. 2012.
- [9] Sosnin, P.I. Software flow control in computerized environments (monograph) / P.I. Sosnin, Yu.A. Lapshov, SN Larin, V.A. Maklayev // ed. P.I. Sosnina. - Ulyanovsk: UISTU, 2014. - 308 c.
- [10] Sosnin P., Cognitively Informed Intelligent Interfaces: Sys-tems Design and Development, IGI Global, 2012, pp. 157–176
- [11] Romodin, M.Yu. Pseudo-code programming in the conceptual design of databases / M.Yu. Romodin, Yu.A. Lapshov, P.I. Sosnin // Automation of management processes. - 2013. - №2 (32). - C. 94-100.
- [12] Ghosh S. Enhance PMBOK® by Comparing it with P2M, ICB, PRINCE2, APM and Scrum Project Management Standards / Ghosh S., Forrest D., DiNetta T. et al. // PM World Today - Jan 2012 - Vol. 14 Issue 1, Special section - p1-77.

*Corresponding author.

E-mail address: alimk_iq@ yahoo.com/Khansaa_aziz@ yahoo.com