

Development of a Prototype Case-Tool for the Company's Website Development in the Environment BPSIM.DSS

Andrey A. Tarasiev
IRIT-RTF; Ural Federal
University named after the
first President of Russia
B.N.Yeltsin
Yekaterinburg, Russian
Federation
andrew4800@mail.ru

Margarita E. Philippova
IRIT-RTF; Ural Federal
University named after the
first President of Russia
B.N.Yeltsin
Yekaterinburg, Russian
Federation
rituly_22@mail.ru

Igor O. Sitnikov
IRIT-RTF; Ural Federal
University named after
the first President of
Russia B.N.Yeltsin
Yekaterinburg, Russian
Federation
i.o.sitnikov@urfu.ru

Olga P. Aksyonova
IRIT-RTF; Ural Federal
University named after
the first President of
Russia B.N.Yeltsin
Yekaterinburg, Russian
Federation

Abstract

In this paper, we describe the development of a scenario for the typed process of developing a company's website in the BPSim.DSS environment, as well as designing and developing templates for the CASE-tool. Within the framework of work, the corresponding diagrams in DFD and UML notations were constructed, and the adequacy of these models was analyzed. The next step is to add the BPSim system to the code generation module of the design web templates.

1 Introduction

At present, the complexity of information systems is growing rapidly, regardless of the purpose for which they are created and on which platforms are oriented. Over the past few years, there has been a trend that most information systems are implemented in the form of web-applications (websites, portals, systems, etc.).

Obviously, new developments in this area are gradually complicating these systems, both in terms of technical aspects and the complexity of design and development. Accordingly, the issue of optimization and automation of the development of such information systems becomes topical.

Typical design patterns can be used to solve this issue. When developing web-applications, the principle of templating in various modules of the system is actively used.

However, usually these templates are libraries, packages, frameworks, templating engines, etc., while design templates, aimed at the development process itself, are not so widely used.

Thus, the goal of the work was the development of scenarios for the typed development of information systems of various types, as well as the creation of design templates for the CASE tool with the aim of further developing the code generation module for the BPSim.DSS system.

BPSim.DSS is a software solution designed to support decision making. It includes a number of packages designed to solve a variety of tasks, including Multi-Agent modeling, CASE tools, and so on. BPSim.SD is used as the built-in CASE-tool. The main advantage of this tool is that unlike most analogues such as Power Designer, Borland Together Designer, IBM Rational, CA Erwin Modeller, it includes various modules necessary for modeling in various notations, self-documentation, reporting, and so on. In other words, BPSim.SD implements the integration of the structural and object-oriented approach. This solution consists of three subsystems: creation of DFD (data flow diagrams), UML diagrams and user interface modeling [3].

Among the main types of applications, the following systems:

- The company's site;
- Internet showcase;
- Information portal;
- Multi-user project management system;
- Multi-Agent system;
- Messengers and chat bots.

In the framework of this paper, the development of the company's website is considered.

For the most comprehensive description of the area when designing information systems, the most successful is the use of object-oriented. The basic standard in this area is UML.

UML (English Unified Modeling Language-unified modeling language) - according to one of the definitions - in the field of modeling processes, in particular in the field of modeling of production processes, business processes, system design, processes for the development of various systems, software and so on, as well as descriptions and mapping of organizational structures [1].

UML is an open standard that uses visual graphical designations to design an abstract system model. This language is used primarily to identify, visualize, design, and document systems [1].

Because of this, to achieve the goal, it was decided to create DFD and UML diagrams that exhaustively describe this area. Some data driven [6-7] and development approach [8-9] are used for software development.

Because the detailed distribution of user roles in a real project team is not so important for creating CASE tool templates, simplified entities were used for modeling.

2 Formulation of the problem

In this paper, we consider the development of the site development script with the help of BPSim.SD package, for further application and expansion of the functionality of this CASE-tool. It should be noted that there are many approaches to the organization of the site architecture. In the first approximation, you should consider the process of creating an application in the MVC architecture.

MVC (Model-View-Controller) - application data separation scheme, user interface and control logic into the three separate component module: model, view and controller - in such a way that the modification of each component can be independently of each other.

This choice is due to the fact that this standard can be most clearly represented using visual modeling tools. In addition, to date, MVC remains the most widely used technology.

In order to achieve greater visibility models, as well as enhance their versatility, it was decided to consider the system in terms of the following actors (agents): developer, designer database, user interface designer (front-ender).

It should be noted that in the real system, the roles presented interact much more closely and more complexly, in addition, the process itself has a cyclical structure, tied to the activities of individual agents. However, to create primary templates such a detailed consideration is superfluous. In the future, with the continuation of the work, more detailed research and modeling is planned.

3 Diagram development

3.1 DFD modeling

Identify the subject area, identify the actors (agents), determine the structure of the system, and identify dependencies and relationships.

To do this, first of all, we need to model the model in DFD notation. The environment BPSim.SD allows you to do this immediately when creating a new project. At the same time, the subject area can be viewed from different points of view, after which the data models can vary. If we consider the system solely as a process, then the data flow diagram can look like this (Figure 1).

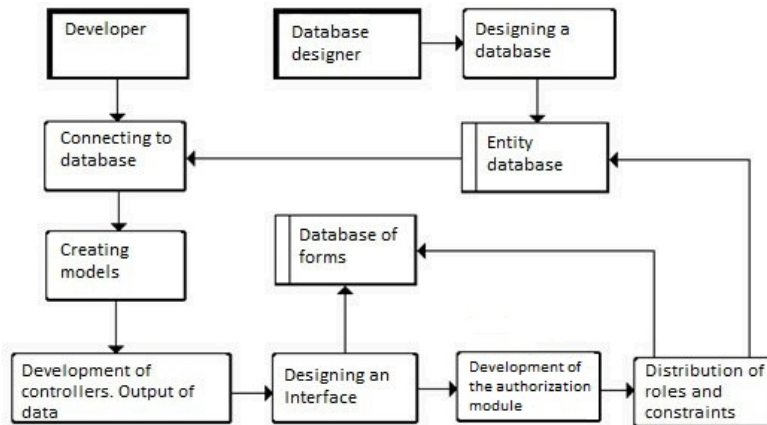


Figure 8. Data flow diagram

BPSim.SD environment allows to automatically convert DFD into a Use Case Diagram (Figure 2).

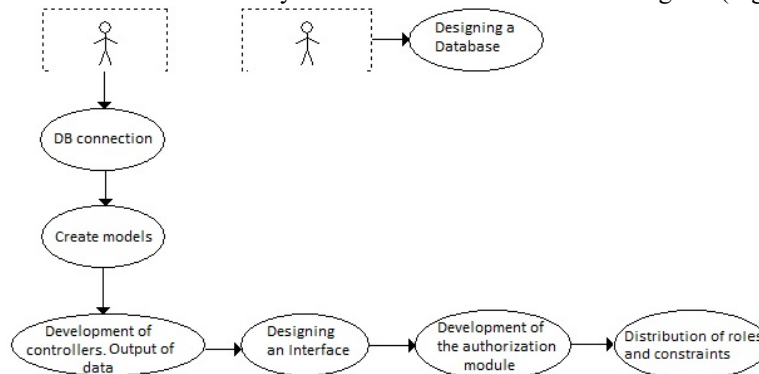


Figure 9. Use Case Diagram

However, this approach is not correct in this case, because to create effective design patterns, it is necessary to consider the system not only descriptively, but also constructively. To describe the processes, it is more reasonable to use IDEF notations or individual UML diagrams.

Thus, it is more reasonable to describe the data flows in the diagram by defining agent methods through the main processes, which can then be decomposed (Figure 3).

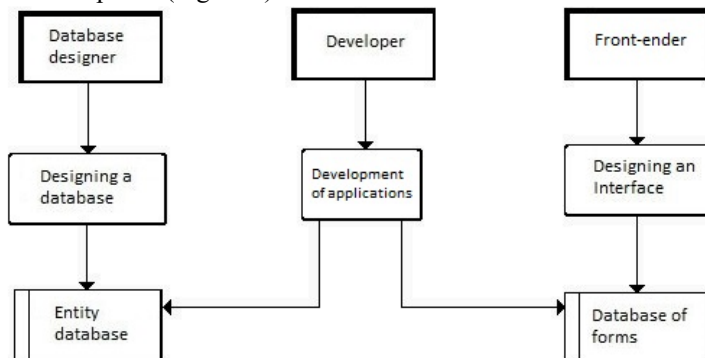


Figure 10. Data flow diagram of subject area

Similar to the previous step, the BPSim.SD environment allows you to automatically convert DFD into a use case diagram (Figure 4).

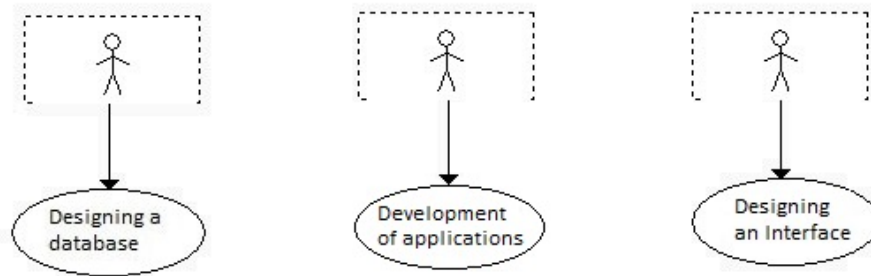


Figure 11. Use Case Diagram

In this model, the following agents are considered: the developer, the database designer and the interface designer (front-end). It is obvious that the designer describes and develops the structure of the database, the front-end implements the design of the user interface (web pages), and the developer develops the application logic, and also combines the results of the work of the other two actors.

When developing templates for a CASE tool, is necessary to use databases to store information about the structure of the application being developed, its database, visual forms of the user interface, and so on [4].

3.2 Creating UML diagrams

The next step is to implement a constructive description of the system using elements of visual modeling, an object-oriented approach [2].

3.2.1 Class diagrams

Class diagrams are used to describe the general internal structure using a constructive approach.

In the given subject area the structure of classes can be presented as follows. Classes are needed for storing the database entities of the developed site and their attributes, as well as visual forms of the user interface, including separate content blocks and their types of html tags (Figure 5).

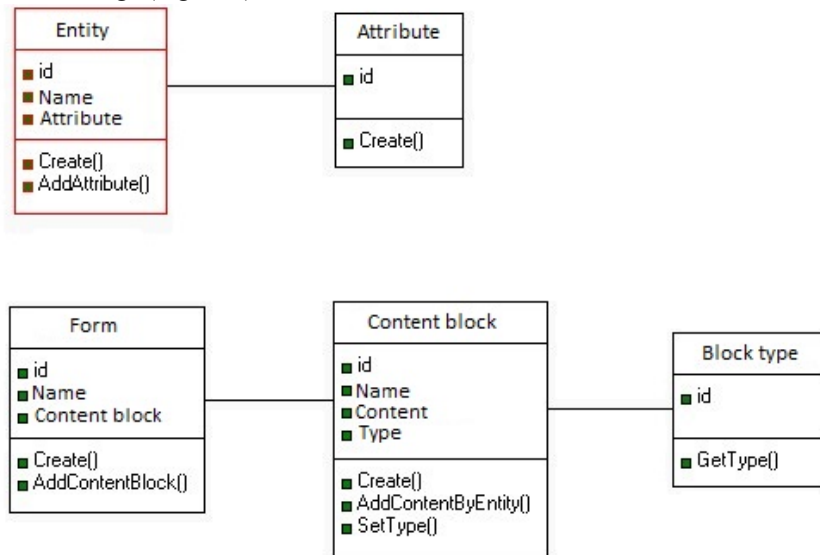


Figure 12. Class diagram

When designing this model, the properties of classes and methods that can be applied later were described.

After creating a class diagram and use cases, the BPSim.SD system allows you to create sequence diagrams to complete the work with the expert system.

3.2.2 Sequence diagrams

With the help of sequence diagrams, UML can describe processes occurring in the system from the point of view of the interaction of actors and business entities described in the class diagram.

The design of these diagrams allows us to define in detail the system and work out a general scenario for the development of the site in terms of describing the interaction of agents. Thus, it becomes evident the need for a multi-agent design approach in the future [5].

The sequence diagrams themselves represent the interaction of the agent-developer, database designer and user interface, business entities implemented in the system and boundary forms-the interfaces to the interaction with the CASE-tool (Figures 6 - 9).

The first diagram demonstrates interaction with the business entities of the system alone, while the subsequent diagrams reveal the multiagent principle of interaction of actors.

When implementing the information system on these diagrams, a product can be created that is aimed at unifying and simplifying the development process. Integration of this product with the existing CASE-priority allows you to extend its functionality.

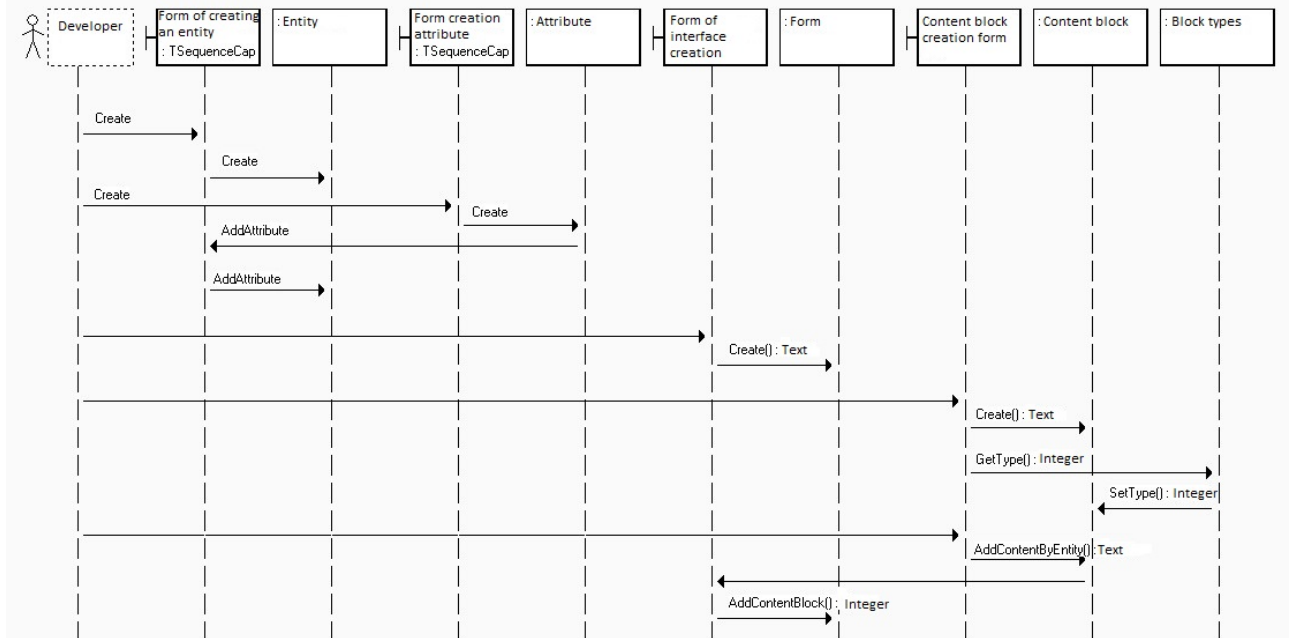


Figure 13. Sequence diagram for the developer in general

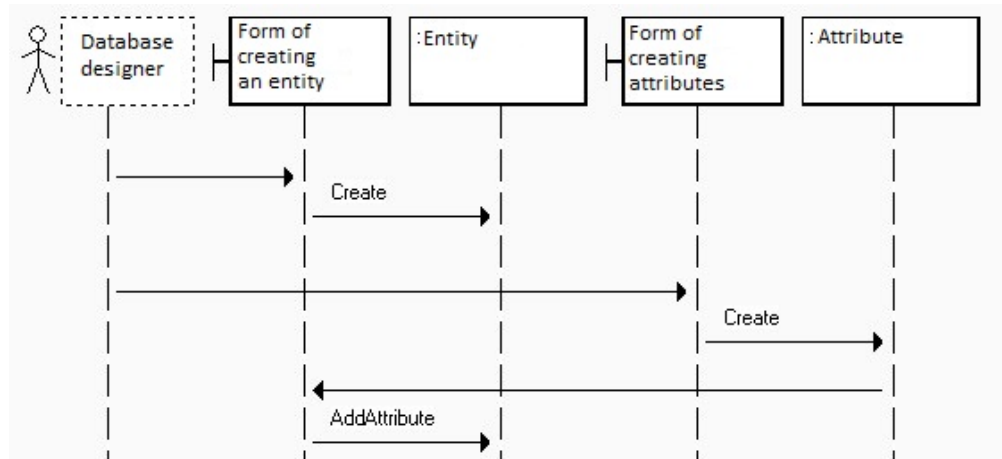


Figure 14. Sequence diagram for the database designer

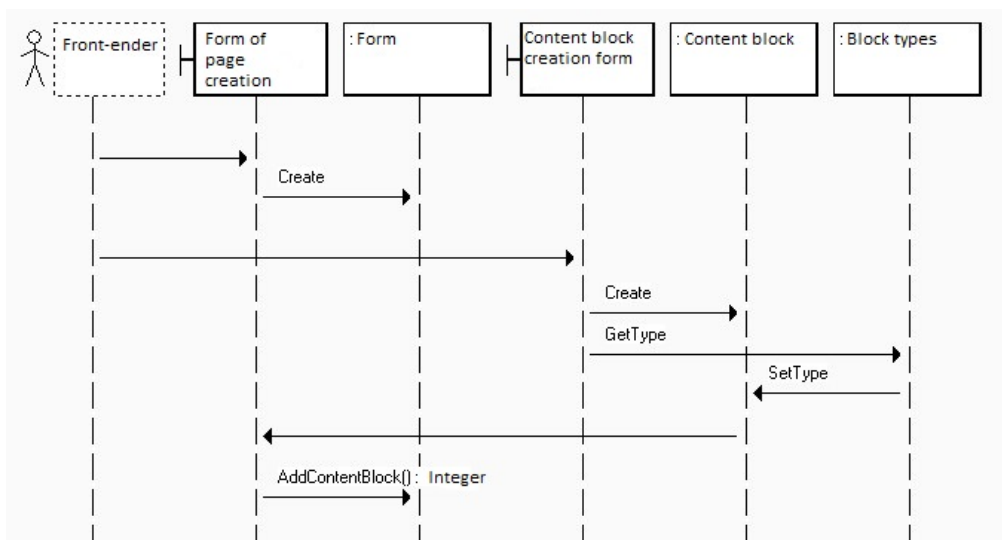


Figure 15. Sequence diagram for the interface designer

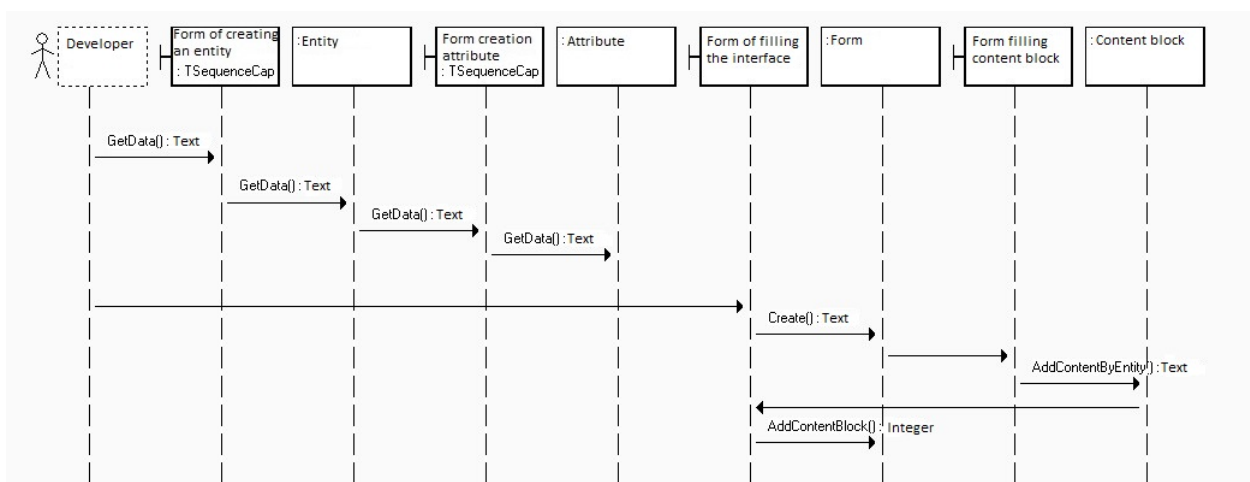


Figure 16. Sequence diagram for the developer

4 Conclusion

Thus, within the framework of the work done, a basic study of the site development script for the CASE-tool was carried out. For this purpose, diagrams have been created in various notations that describe the given subject area, on the basis of which design patterns can be created in the future. These diagrams include DFD, use case diagrams, classes and sequences.

In the future, it is planned to continue work in this area, complete the design in the area under consideration, and consider the related ones that were listed.

The results of the work will be applied to extend the CASE-tool BPSim.SD, and in the future will allow to developing information systems oriented to web-solutions, using design patterns and visual modeling.

Acknowledgements

The work was supported by Act 211 Government of the Russian Federation, contract № 02.A03.21.0006.

References

1. K. Larman. Application of UML and design patterns {St. Petersburg, 2002
2. H. Buch. Object-Oriented Analysis and Design with Applications {Moscow, 2010
3. I. A. Spitsina, K. A. Aksenov. Multiagent method of analysis and synthesis of information systems {Ekaterinburg, 2017
4. K. A. Aksyonov, E. A. Bykov, O. P. Aksyonova. Development and application of software engineering solution BPSim.SD. Proceedings - UKSim-AMSS 7th European Modelling Symposium on Computer Modelling and Simulation, 321-325 {November 2013.
5. K. A. Aksyonov, E. A. Bykov, O. P. Aksyonova. Applications of simulation-based computer aided software design solution using BPSim.SD Proceedings - UKSim-AMSS 7th European Modelling Symposium on Computer Modelling and Simulation, 326-331 {November 2013.
6. Kuleshov, S. V., Aksenov, A. J., Zaytseva A. A.: Software-Defined Data Formats in Telecommunication Systems. Series: Advances in Intelligent Systems and Computing. 575 , 326–330 (2017)
7. Alexandrov, V. V., Kuleshov, S. V., Zaytseva, A. A.: Active Data in Digital Software Defined Systems Based on SEMS Structures. Logical Analysis of Data and Knowledge with Uncertainties in SEMS – Smart Electromechanical Systems, Studies in Systems, Decision and Control. Gorodetskiy, A. E. (ed.). 49, 61–69 (2016)
8. Yudin, E. B., Yudina, M. N.: Calculation of number of motifs on three nodes using random sampling of frames in networks with directed links. 2017 Siberian Symposium on Data Science and Engineering (SSDSE), 23–26. DOI: 10.1109/SSDSE.2017.8071957
9. Borodin, A., Kiselev, Y., Mirvoda, S., Porshnev, S.: On design of domain-specific query language for the metallurgical industry. In: Proceedings of 11th International Conference BDAS: Beyond Databases, Architectures and Structures: Communications in Computer and Information Science, pp. 505–515 (2015)