Adaptive User Interfaces for Intelligent Systems: Unlocking the Potential of Human-System Interaction

Mikhail Sadouski, Pavel Nasevich, Maksim Orlov, Alexandra Zhmyrko
Belarusian State University of
Informatics and Radioelectronics
Minsk, Belarus
Email: sadovski@bsuir.by, fallenchromium@gmail.com, orlovmaksimkonst@gmail.com, aleksashazh@gmail.com

***Abstract*—The paper analyzes the capabilities of computer systems and the level of development of tools for interacting with them (user interfaces). Based on the analysis, an approach to the design of adaptive user interfaces of intelligent systems based on the OSTIS Technology is proposed. The semantic model of such interfaces, proposed earlier, has been clarified and extended, the proposed architecture of such systems is given. Adaptive user interfaces designed on the basis of the proposed approach will provide new scenarios of user interaction with computer systems.**

***Keywords*—adaptive user interface, intelligent systems, user interface of ostis-systems**

I. Introduction

In the modern world, people use computer systems of various purposes daily. A key component of such systems that directly influences their efficiency is the user interface, which in a broad sense is a set of tools that provide interaction between a person and the system.

A large part of the cost of developing computer systems is in the design, testing, and development of the user interface [1].

A poorly designed user interface limits the potential of the system by increasing the threshold of entry and reducing the efficiency of interaction with users or making some interaction scenarios impossible [2].

With the development of information society, the need of users for computer systems capable of solving various classes of problems, including tasks that are difficult to formalize, has led to an increased pace of development of computer technologies, the creation of a large number of models, methods and tools for the design and development of computer systems, including intelligent computer systems with increased requirements for interoperability, component compatibility and flexibility of scenarios of interaction with the user because of their ability to self-learning and solving complex problems, as well as problems, in the initial data and algorithms of solution of which there is an influence of non-factors.

However, even computer systems that we use on a daily basis are severely limited in their functionality due to the limited means of interaction with these systems. There is a mismatch between the current level of development of user interfaces and the problem-solving capabilities of computer systems.

Each user has unique needs and the application of adaptive user interfaces for intelligent systems becomes essential. The ease and flexibility of dynamically changing user interfaces based on user tasks that are not predetermined in the design of the system becomes key and allows for greater potential for interaction with intelligent computer systems.

This article analyzes the capabilities of computer systems and the level of development of tools for interacting with them (user interfaces).

The user interface is considered in this context as the language of communication between the system and the user, together with the means for such communication, emphasizing the content of the interaction more than the specific technical aspects of the implementation of the interaction. This analysis is based on a description of the historical development of computer systems (and, as a consequence, the development of the many classes of problems that computer systems solve).

Based on the analysis, an approach to the design of adaptive user interfaces of intelligent systems based on the OSTIS Technology is proposed. The proposed [3] semantic model of such interfaces is clarified and extended, and the proposed architecture of such systems is given. Adaptive user interfaces designed on the basis of the proposed approach will provide new scenarios of user interaction with computer systems.

II. Analysis of existing approaches to solving the problem

A. Challenges of modern user interfaces

The challenges of modern user interfaces that cause users to fail to take advantage of the full potential of computer system problem solvers can be categorized as follows:

- A mismatch between the user's skills and means of interacting with the system and the actual means of interaction provided by the system. An example is an over-complicated user interface for inexperienced users, or a user interface that does not take into account that the user is fully informed about the algorithm for solving a problem and takes time and attention away from unnecessary explanations. This leads to the fact that the user cannot predict the navigation through the user interface — his cognitive load increases, the time required to solve the problem increases [4].
- Mismatch between the task and environment for which the user interface was designed with the actual task and environment of the system. This refers to the case where the system's problem solver is capable of solving a broader and more general class of problems than the system's user interface allows, since only in such a case can the user interface alone be said to be the limiting factor in the applicability of the system — [5].
- Lack of user's ability to make changes to the interface for integration with other user interfaces and/or systems. The impossibility of building an integrated working environment (i.e. a new user interface, qualitatively different from the multitude of interfaces designed to solve each of the sub-tasks of the complex task) makes it impossible to build scenarios for automatic integration of computer system subsystems and severely limits the user's ability to integrate computer systems, because without the ability to change the user interface, the program behavior can only be changed programmatically [6], [7].

There are a lot of examples of functionalities limited on the user interface side, which are nevertheless technically realizable, but we will limit ourselves to a few significant ones:

- Lack of integration of system components: your word processor does not prompt you to open the last quarter's financial report, even though you were emailed it yesterday and are scheduled to check it on your electronic calendar today.
- Lack of automatic adaptation to the environment: the online store does not offer you to make a pickup to the branch you are closest to at the moment.
- Lack of ability to personalize the user interface: your fitness app does not allow you to fully customize the start screen with health metrics that are irrelevant to you and with workout suggestions that do not match your goal and access to fitness equipment.
- Solving complex tasks: for an average user the interface for automated solution of a multi-step task is not standard and is always available. For example, to process a call or a letter from a customer, enter relevant information from the letter into the project accounting system, assign a person with the necessary competence to be responsible for the project and send the customer a letter with the contact of the person responsible for the project.

It is worth noting that for each of these examples it is possible to find a counterexample: a computer system that did take into account the described use case, but it is worth recognizing that in general the described limitations exist in the vast majority of computer systems.

B. Overview of the evolution of computer systems and their user interfaces

Let us consider the development paths of computer systems and their user interfaces - the major milestones, and what they were driven by (discoveries or user needs).

Over the last 60 years, a large number of approaches to user interface design have been developed. The approaches focused on different directions, such as "which side of the interaction is primary" (the emotional state of the user, the purpose of the system, feedback from the user, the design of the program that implements the algorithm for solving the problem) and "how to convey the meaning of the objects of the user interface" — confrontation of metaphorical and idiomatic approach [8], [9].

At the same time, the paradigms used to implement the user interface have evolved. For example, the transition from a model where each interaction has an underlying function call to perform the process and provide feedback to an object-oriented interface where the UI elements correspond to the properties of the system entities (reactive approach).

Currently, the approach when the user interface is described by a set of states and transitions in the algorithm of problem solving is popular (i.e. the paradigm of so-called "wizards", which leads the user step-by-step from the need to its resolution). Each of the approaches has a place, but a special place in the design of user interfaces for intelligent computer systems, based on the dynamic nature of the tasks that can be set before it, takes the paradigm of interface design based on the problem being solved and the algorithm for its solution. To date, there are no comprehensive means of building adaptive user interfaces (taking into account the environment, the characteristics of the user and his device), while dynamically taking into account the information about the algorithm for solving the problem.

C. Conclusions

Based on the representation of the user interface described in I, we can conclude that the problems described in II-A are the result of a methodological error in the design of interfaces. At the moment, the factors affecting how the user interface should look like are taken into account by the developer at the time of creation of the computer system and are laid in its program code in the form of ready-made structures created under the influence of the model of communication with the user, the model of the environment, the model of the user and the proposed algorithm for solving the problem. Often this modeling process does not lead to an optimal result. In addition, computer systems tend to receive new functions within their life cycle, which entails updating the user interface, which in turn leads to repeated manual design of the user interface taking into account all the factors described above.

User interfaces of computer systems are suboptimal largely because the toolkit for creating user interfaces with tools to solve the problems described above is not sufficiently refined, versatile, or widespread. In order to create a complete interface development toolkit, it is necessary to describe the mechanisms of system-human communication, in particular, to define the classes of transmitted messages in order to build a model of human communication based on the task facing the system. It is also necessary to describe what components of the user interface implement certain messages between the system and the user; describe relevant properties of the user and the environment, and automate the rules by which the user interface should change in the presence of certain properties of the environment or the user. Thus, formalize the experience of experts in building user interfaces and turn it into an automated algorithm. Such a tool would have the ability to generate an interface for systems with dynamic problem formulation, adapt and personalize the interface to the user and the environment.

III. Proposed approach

A. General principles of building adaptive user interfaces of intelligent systems

An ontological approach based on the semantic model proposed in [3] is suggested to build adaptive user interfaces of intelligent systems. This approach assumes:

- "lexical" interface description — a description of the components from which the interface is formed;
- "syntactic" description of an interface — rules for forming a correct interface from its components;
- semantic description. Knowledge about what entity the displayed component is a sign of. The semantic description also includes the purpose, scope of application of the interface components, description of the user's interface activity, etc. The semantic description also includes the purpose, scope of application of the interface components, description of the user's interface activity, etc.

The following ontologies need to be implemented within the approach:

- ontology of problems and algorithms for their solution;
- ontology of user interfaces of intelligent computer systems;
- ontology of components of user interfaces of intelligent computer systems;
- ontology of the context of use of user interfaces of intelligent computer systems;
  - ontology of users of intelligent computer systems;
  - ontology of intelligent computer systems users' actions;
  - ontology of the environment of intelligent computer systems;
  - ontology of devices of intelligent computer systems;
- ontology of external information constructs;
- ontology of incoming and outgoing messages of intelligent computer systems.

In addition to the proposed ontologies, it is necessary to develop tools for automatic interface generation and editing of its model to enable interface modification during operation and tools to support the design of user interfaces.

The above means are also proposed to be realized with the help of the above ontologies, which allow to fully describe in the knowledge base of the intelligent system both the interface itself and the principles of its interaction with the user, as well as the mechanisms of adaptation of the interface depending on the user and the environment. It is important to note that algorithms for solving user tasks are also proposed to be formed dynamically in the knowledge base of the system.

The application of this approach requires a basis in the form of intelligent systems design technology, which allows solving complex problems and whose components integrate with each other and have a synergistic effect. In turn, the application of the ontological method of building adaptive user interfaces within the framework of such technology will make intelligent systems practically applicable for a large number of classes of tasks.

The technology that meets these requirements is the OSTIS Technology. Intelligent systems developed according to the OSTIS Technology are called ostis-systems [10].

The advantages of the OSTIS Technology are as follows:

- unification of different types of knowledge with the help of SC-code;
- ontological approach to knowledge structuring;
- availability of a variant of realization of the platform of interpretation of semantic models of intellectual systems (sc-models);
- problem solver is based on a multi-agent approach in which agents interact with each other solely by specifying the actions they perform in a shared semantic memory;

- library of reusable knowledge base components and problem solvers;
- the possibility of semantic representation of logical formulas and statements;
- availability of a variant of implementation of the interpreter of logical models of problem solving.

The OSTIS Technology allows to implement the user interface in the simplest and most efficient way due to the use of automation tools for the design of user interface components, as well as due to the library of reusable components of ostis-systems, which significantly reduces the time of interface development [11].

The main ostis-system within the OSTIS Ecosystem is the OSTIS Metasystem [12]. The OSTIS Metasystem allows to automate the development of ostis-systems and their components, including user interfaces [13].

The internal universal language of knowledge representation is SC-code [14]. With the help of SC-code the interface of ostis-systems itself and the programs and rules that are used when using the interface are recorded in a unified way. That is, using SC-code, it is possible to write the interface, and also the program which can not only implement some business logic of system, but also the program which will change the interface itself which is used at interaction with the user. Through the use of SC-code and the principles of the OSTIS Technology it is possible to realize the adaptability of user interfaces in the simplest and most flexible way.

To implement the user interface as a subsystem with its own knowledge base and problem solver, it is necessary to develop the listed family of user interface ontologies, as well as a collective of agents for the functioning of the user interface:

- non-atomic sc-agent of interpreting sc-model of user interfaces of ostis-systems;
- non-atomic sc-agent of generation of sc-model of user interface of ostis-systems;
- non-atomic sc-agent of adaptation of sc-model of user interface of ostis-systems to a particular user and external environment;
- non-atomic sc-agent of interpretation of user actions; user interfaces of ostis-systems.

B. *User interface module for ostis-systems*

The implementation of the user interface of ostis-systems is based on:

- extensibility without the need to change the logic of the module (for personalization and adaptation on the part of the user and for styling and fine-tuning on the part of the system developer);
- maximum simplification for developers to add a new user interface platform (other than the current web-oriented);
- ensuring the simplest possible integration of the UI module into existing systems;
- ensuring that the final interface is comparable in speed to user interfaces developed by traditional means;
- support for a high level of abstraction for operating user interface objects, thus not limiting developers using any means to communicate with users (including virtual reality helmets or interaction with real world objects).

In this regard, the following architectural decisions have been made:

- to implement the functionality of translating the sc-model of the user interface into a specific platform-dependent markup on the side of the ostis-systems problem solver;
- to implement the functionality of user actions interpretation on the side of ostis-systems problem solver;
- to implement a mechanism to track updates to the user interface model and pass only the changed part of the markup to the user interface implementation platform;
- to use a web-oriented platform as the first supported platform for user interface implementation;
- not to use web primitives to describe interfaces, but to limit ourselves to higher-level concepts and relations, such as size, algorithm of mutual arrangement, decomposition of a user interface element (an example of the description of the "Button" component is presented in Figure 1);
- to provide the possibility of applying the rules of user interface adaptation defined declaratively in the knowledge base of ostis-systems.

The general architecture of the module is presented in Figure 2.

The order of interaction can be described as follows:

- the user interacts with the user interface of the ostis-system to solve the task he/she needs;
- information about the interaction is transferred to the knowledge base of the system;
- initiation of a team of agents to solve the user's task is performed;
- results of the user's task solution are stored in the system's knowledge base;
- the user interface agent collective is initiated;
- user interface agents modify the interface model in the knowledge base based on the received information about the result of the performed task as well as the current usage context;
- as a result, adaptation is performed - the user interface model in the knowledge base of the system is changed;
- changing the user interface model in the knowledge base of the system leads to an updated visualization of the user interface.

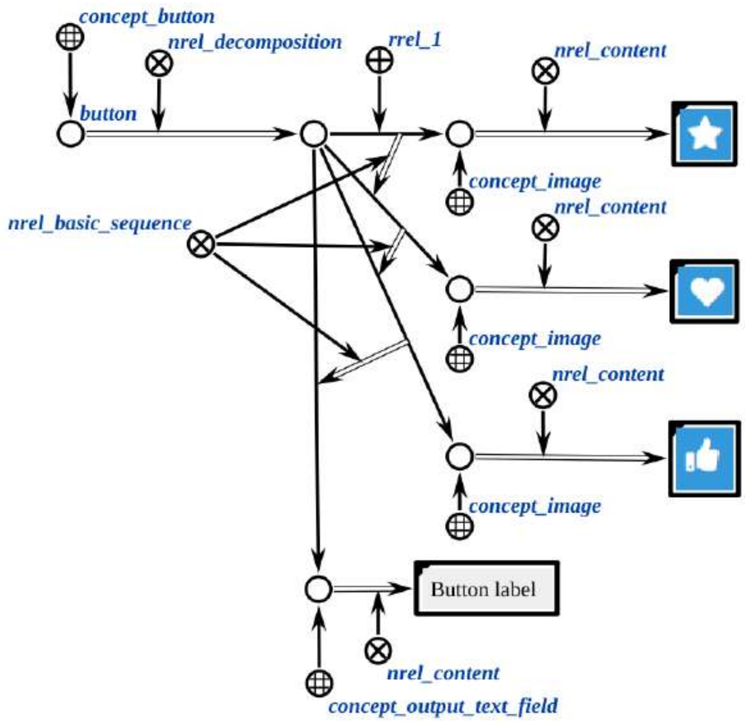

Figure 1. Example of button decomposition

Description of user's action takes place in the knowledge base of the system. An example of formalization of a single button press is shown in Figure 3. The sequence of click-based UI updates is shown in Figure 4.

The algorithm for updating the UI after a user action involves the following steps:

1) The user performs the action of pressing the button.
2) The user interface captures information about the user's action in the system's knowledge base.
3) When this information appears, the user action interpretation agent (UIActionAgent) is initiated.
4) This agent searches the knowledge base for an internal system action that should be executed as a result of pressing a button. The search is based on information about the class of the user action and the UI component for which it was initiated.
5) Having received a template for creating an internal action in the system's knowledge base, UiActionAgent initiates its execution.
6) An internal system action is performed. If necessary, the user interface model in the knowledge base is changed.
7) An UpdateModelAgent call is executed to update the user interface model.
8) For each modified component of the user interface, the agent of forming a visualization of this component for the used platform is called.
9) The modified part of the user interface is redrawn.

The key components of the module are:

- User interface sc-model generator. It is responsible for compiling requirements for the current state of the user interface based on the solution step of the current task. Based on the solution algorithm, the generator finds a set of input information received from the user and a set of elements on which it is necessary to give feedback. Further, the generator specifies the types of received information and feedback, searches for the most general classes of user interface components that solve the problem of input and output of information for this particular task and creates/modifies the sc-model of the user interface containing all the necessary elements.
- SC-model adaptation agent. It is responsible for adding the properties necessary for interpretation to the user interface components (e.g. size, color of elements, their location, etc.) based on the system settings, knowledge about the environment and rules of adaptation to the environment, knowledge about the user and rules of adaptation to certain user characteristics.
- Interpreter. It converts a specific user interface sc-model into an external language compatible with the user interface implementation platform, so that the user interface can be recreated on the platform or the state of an existing interface can be updated.
- User action interpreter. It obtains information about user interactions with the interface and triggers the system to perform the action described in the user

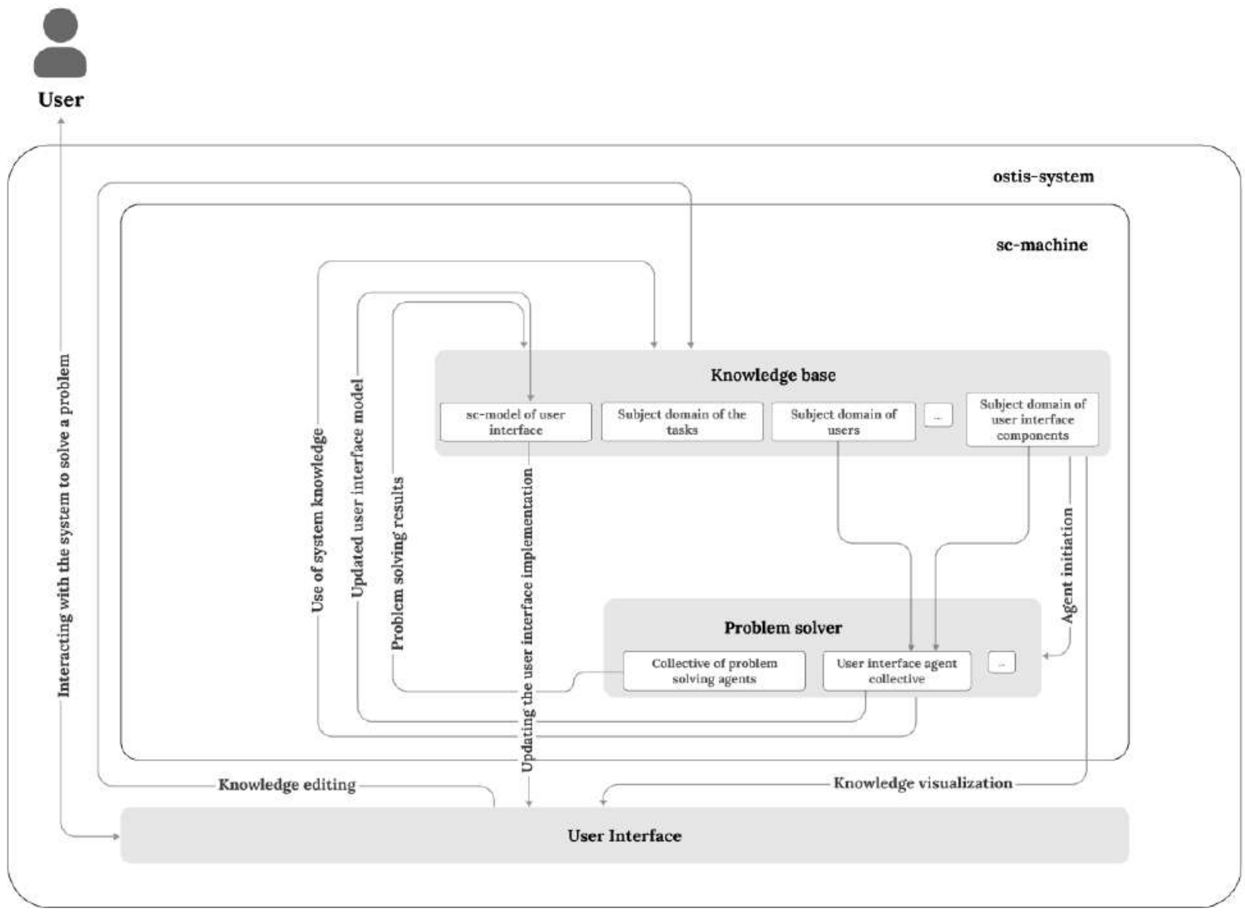

Figure 2. User interface architecture

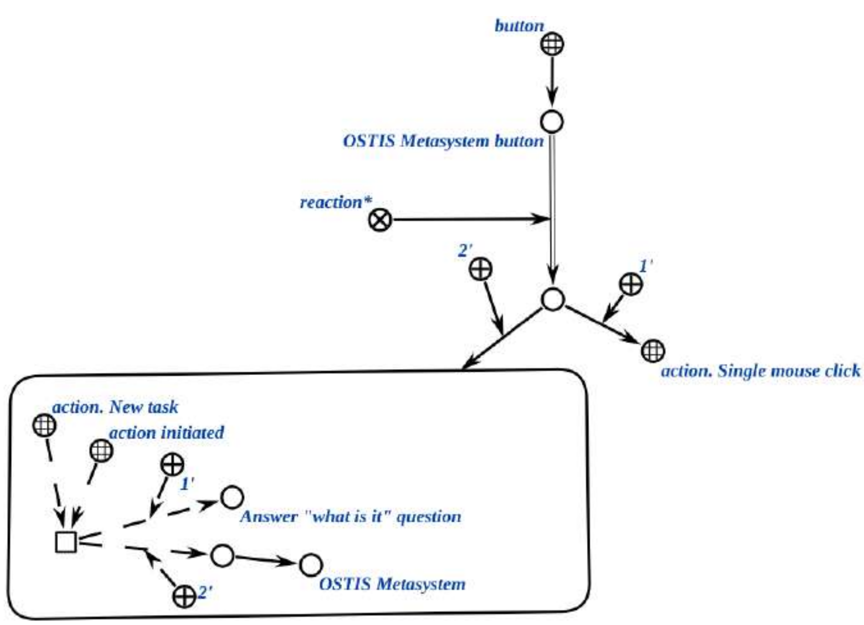

Figure 3. User action example

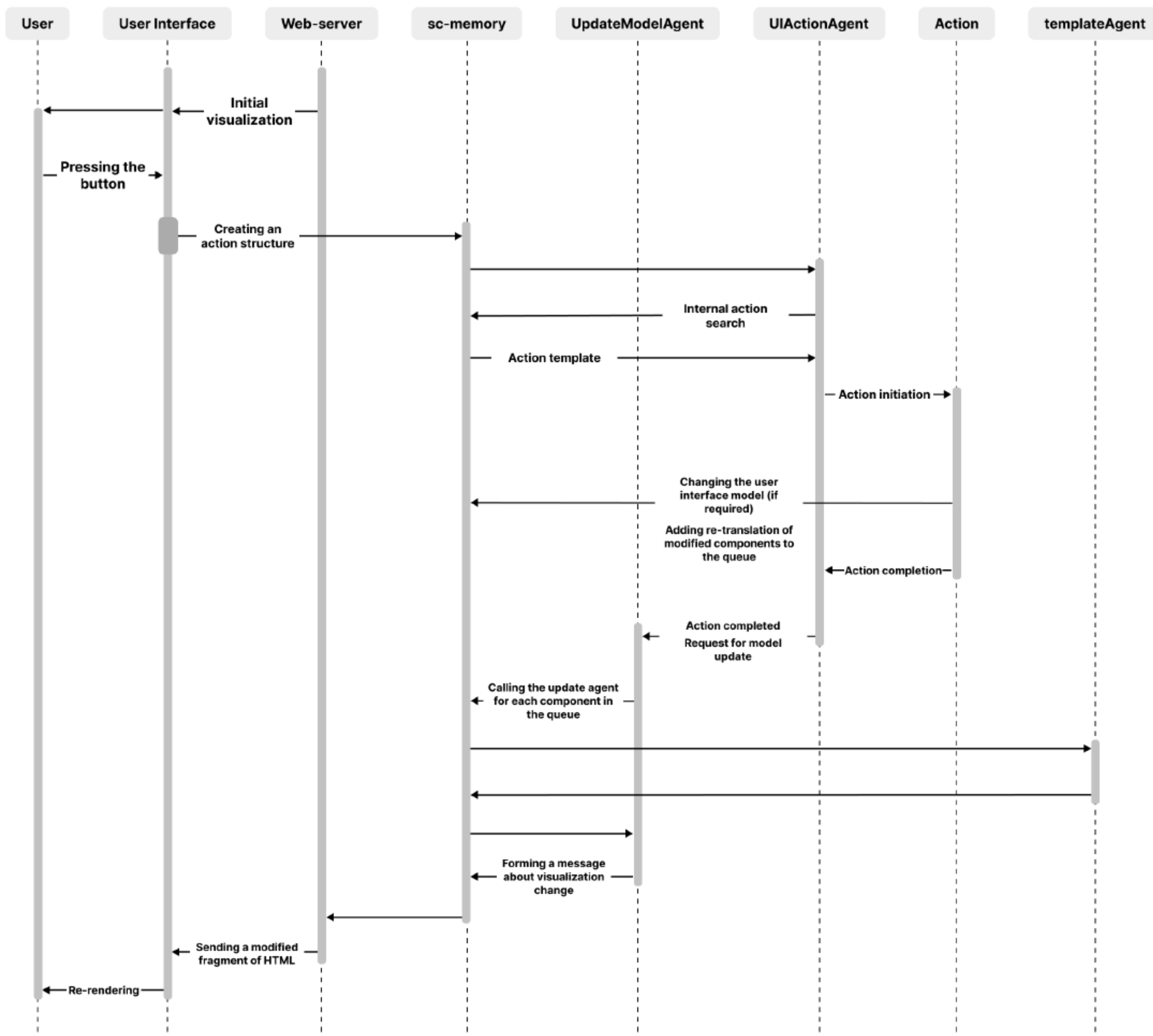

Figure 4. Sequence diagram for updating the user interface based on user interaction

interface component model as a reaction to one or another type of interaction.

The user interface implemented according to the specified architecture will be generated on the basis of its model in the knowledge base, adapted to the needs of users, and dynamically changed depending on the tasks to be solved.

IV. Conclusion

The paper analyzes the capabilities of computer systems and the level of development of tools for interaction with them (user interfaces). Based on the analysis, an approach to the design of adaptive user interfaces of intelligent systems is proposed to provide new scenarios of user interaction with computer systems.

To apply this approach, the OSTIS Technology is used, which allows solving complex problems and whose components integrate with each other and have a synergistic effect. In turn, the application of ontological approach based on the semantic model of building adaptive user interfaces within the framework of the OSTIS Technology allows making intelligent systems practically applicable for a large number of classes of tasks. The paper refines and extends the previously proposed semantic model of adaptive user interfaces of intelligent systems and presents their proposed architecture.

Acknowledgment

The authors would like to thank the scientific collectives of the departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and Brest State Technical University for their help and valuable comments.

References

[1] (2024, February) Data-driven ui: unlimited power. [Online]. Available: https://mobius-piter.ru/en/2018/spb/talks/v96lokugwe8cwggio8ois/

[2] B. A. Myers and M. B. Rosson, "Survey on user interface programming," ser. CHI '92. New York, NY, USA: Association for Computing Machinery, 1992, p. 195–202. [Online]. Available: https://doi.org/10.1145/142750.142789

[3] M. E. Sadouski, "Semantic models and tools for designing adaptive user interfaces of intelligent systems," *Informatics*, vol. 20, no. 3, p. 74–89, Sep. 2023. [Online]. Available: http://dx.doi.org/10.37661/1816-0301-2023-20-3-74-89

[4] S. Abrahão, E. Insfran, A. Sluÿters, and J. Vanderdonckt, "Model-based intelligent user interface adaptation: challenges and future directions," *Software and Systems Modeling*, vol. 20, no. 5, p. 1335–1349, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1007/s10270-021-00909-7

[5] A. Wolff, P. Forbrig, A. Dittmar, and D. Reichart, "Linking gui elements to tasks: Supporting an evolutionary design process," vol. 127, pp. 27–34, 01 2005.

[6] D. Shunkevich, "Agentno-orientirovannye reshateli zadach intellektual'nyh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132.

[7] N. Zotov, "Semantic theory of programs in next-generation intelligent computer systems," in *Open semantic technologies for intelligent systems*. BSUIR, Minsk, 2022, pp. 297—-326.

[8] V. N. Lukin, A. L. Dzyubenko, and Y. B. Chechikov, "Approaches to user interface development," *Program. Comput. Softw.*, vol. 46, no. 5, p. 316–323, sep 2020. [Online]. Available: https://doi.org/10.1134/S0361768820050059

[9] M. G. Shishaev and V. V. Dikovickij, "Tekhnologiya sinteza adaptivnyh pol'zovatel'skih interfejsov dlya mul'tipredmetnyh informacionnyh sistem," *Trudy Kol'skogo nauchnogo centra RAN*, no. 5 (24), pp. 101–108, 2014.

[10] V. Golenkov, Ed., *Tehnologija kompleksnoj podderzhki zhiznennogo cikla semanticheski sovmestimyh intellektual'nyh komp'juternyh sistem novogo pokolenija [Technology of complex life cycle support of semantically compatible intelligent computer systems of new generation]*. Bestprint, 2023.

[11] M. Orlov, "Control tools for reusable components of intelligent computer systems of a new generation," *Open semantic technologies for intelligent systems*, no. 7, pp. 191–206, 2023.

[12] K. Bantsevich, "Metasystem of the ostis technology and the standard of the ostis technology," in *Open semantic technologies for intelligent systems*, ser. Iss. 6, V. Golenkov, Ed. BSUIR, Minsk, 2022, pp. 357–368.

[13] A. Zagorskiy, "Principles for implementing the ecosystem of next-generation intelligent computer systems," *Open semantic technologies for intelligent systems*, no. 6, pp. 347–356, 2022.

[14] V. Golenkov, N. Guliakina, and D. Shunkevich, *Otkrytaja tehnologija ontologicheskogo proektirovanija, proizvodstva i jekspluatacii semanticheski sovmestimyh gibridnyh intellektual'nyh komp'juternyh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint [Bestprint], 2021.

АДАПТИВНЫЕ ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ: РАСКРЫТИЕ ПОТЕНЦИАЛА ВЗАИМОДЕЙСТВИЯ "ЧЕЛОВЕК-СИСТЕМА"

Садовский М. Е., Насевич П. Е.,
Орлов М. К., Жмырко А. В.

В работе проведен анализ возможностей компьютерных систем и уровня развития инструментов взаимодействия с ними (пользовательских интерфейсов). На основе анализа предложен подход к проектированию адаптивных пользовательских интерфейсов интеллектуальных систем на основе Технологии OSTIS. Уточнена и расширена предложенная ранее семантическая модель таких интерфейсов, приведена предлагаемая архитектура таких систем. Проектируемые на основе предложенного подхода адаптивные пользовательские интерфейсы обеспечат новые сценарии взаимодействия пользователей с компьютерными системами.

Received 01.04.2024