NoSQL Storage for Configurable Information System
The Conceptual Model

Sergey Kucherov, Yuri Rogozov and Alexander Sviridov
Department of System Analysis and Telecommunications, Southern Federal University,
44 Nekrasovsky St., Taganrog, Russian Federation

Keywords: Data Storage, NoSQL, Configurable Information System.

Abstract: A basis for creation of configurable information systems and data warehousing for them is the using of a single form of representation - base abstraction for action synthesis. The base abstraction for action synthesis (BAfAS) involves a description of information system as an user's activity, rather than as individual components. Configurable information system's data storage should be focused on storage not isolated facts, but on types of user actions. Implementation of these action types in turn will lead to obtain facts. Purpose of the paper is to propose the model of configurable information system's storage, which is adequate to a model of the system and the conceptual model of storage described with NoSQL-technologies.

1 INTRODUCTION

Be advised that papers in a technically unsuitable form will be returned for retyping. After returned the manuscript must be appropriately modified. The transition to a new generation of information systems (IS) (Rogozov and Degtyarev, 2014a) - configurable IS (CoIS) (Garcia and Goldszmidt, 2007), differ both in architecture (Rogozov and Degtyarev, 2013) and the principles of operation (Sviridov, 2014), causes the appearance of another data storing principles and ways of its implementation (Kucherov, Rogozov and Sviridov, 2014).

Setting aside technical and methodological features of development processes an IS can be considered as a reflection of user actions performed using other means, but with the same result (Rogozov, 2013). We considered classical approaches to a design of flexible software systems (Rossi, Guevara and Enciso, 2010), and formulated the following abstract scheme of an IS design process from the "main abstraction" (Rogozov, Sviridov and Belikov, 2013) viewpoint (Fig. 1).

Initial state of a domain is a performing certain actions by user. First step in transition to an information system is describing user actions in the form of business process model that is fixing his actions.

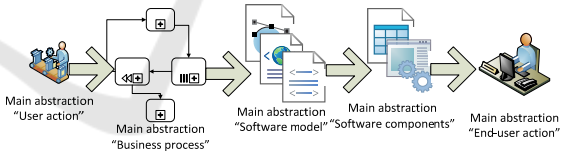

Figure 1: Abstract scheme of the classical IS design process.

Depending on design concepts representation of business processes can vary, but in any case there is the first step to avoiding the usual to an user "action" abstraction. Next step is representation of a business process as a set of abstractions describing an information system, which is capable to performing the process. Usually, the set abstractions at the modelling stage is the structural (IDEF, DFD) or object (UML) model. At the design and implementation stage abstractions are interface, business logic and data, which is absolutely not comparable with user's "action" abstraction. Despite the fact that as a result we change user action (performed manually) to end-user action (performed by an information system), the form of presentation varies significantly. This variety creates a lot of problems, and the main problem is misunderstanding between an user and a developer. The reasons are lies in application of classical analysis methods to nonclassical problem for them (Lazarev, Rogozov and Sviridov, 2014).

In our opinion, the only abstraction that can

Kucherov S., Rogozov Y. and Sviridov A..
NoSQL Storage for Configurable Information System - The Conceptual Model.
DOI: 10.5220/0005499802790287
In *Proceedings of 4th International Conference on Data Management Technologies and Applications* (DATA-2015), pages 279-287
ISBN: 978-989-758-103-8

represent IS in adequate to a domain and to an user form is the base abstraction for action synthesis (BAfAS) (Rogozov, 2014a). BAfAS is the fixed set of characteristics with variable content (Rogozov and Sviridov, 2014a). Changing content determines the assignment of actions, while the fixed structure of an action representation (characteristics: element, function, tool, result) allows to use one abstraction throughout the process of creating an IS - from a study of subject area, prior to an implementation and an operation of the system (Rogozov and Sviridov, 2014b). This, in turn, will require different methods of system analysis and synthesis based on the "action" abstraction. The paper introduces the concept of configurable information system's storage based on the "action" abstraction and NoSQL technologies.

Configurable information system should primarily provide the flexibility to customizing, extending and changing functionality with minimal effort (Kucherov, 2013). In this context, there is necessity to shift from the traditional IS partitioning to "interface", "business logic" and "data" to a common base abstraction (Rogozov, 2014b). Base abstraction should be the unified way of an IS representation. For many years there was the problem of misunderstanding between developer and end-user (Kucherov, Lipko and Shevchenko, 2014), caused by substitution of concepts in the implementation of the system: the end-user requires automation of specific actions or his entire activity, while the developer operates mentioned above aspects - "interface", "business logic", "data" (Kucherov, 2013).

BAfAS allows to simulate the user activity, but not parts of an information system, reproducing the user's actions (as it does a lot of existing methodologies). This is the principally new viewpoint on information systems (Rogozov, 2013). To describe an end-user's activity BAfAS uses a simple set of characteristics - elements, functions, tools and results. Regardless of the detalization level this set stays fixed and represents the shape. Filling the BAfAS shape with content (specification of elements, functions, etc.) is a process of construction an action. Actions could form the activity by general characteristics, by binding through "result = element" or by nesting in each other.

The task different from data storing is arising (Kucherov, Samoylov and Grischenko, 2014). There is necessity to storage of action types, filling content and implementation of which will allow to obtain traditional data facts (i.e. stored named values). Action types and actions (as a reflection of the user work with information system), always consist of a fixed set of characteristics and have enhanced binding possibilities (Kucherov, Sviridov and Belousova, 2014) laid down in the BAfAS. The set of stored action types and actions is not simply the result of the user work with IS, but also the IS configuration. Filling the structure of action type with content is the process of configuring an IS. Performing of action and remembering this process is a reflection of the user work with IS. The configurable information system's storage should have the next number of new properties:

- Shifting the focus in storage: key aspect is action type (reflects configuration of the IS) and the action itself (lead to obtain the fact), not some fact taken separately (result of action);
- Expansion of the connection's significance between stored objects. In the classical approach the relationship between instances of stored data objects means their mutual affiliation. In the case of storing actions connection determines the sequence of steps in automated end-user activity.

The possibility of using widespread data storage technologies for obtaining of the above properties requires analysis. In the paper we represent CoIS storage concept that are based on the BAfAS and NoSQL-technologies. The action storing model and requirements to CoIS storage is formulated in the first section. The second section presents the results of data storage technologies analysis in case of their application to solve described problem. The third section describes a conceptual model of CoIS storage.

2 THE MODEL OF ACTION REPRESENTATION IN CONFIGURABLE INFORMATION SYSTEMS

The data model (the most close in meaning term) includes three aspects (Codd, 1981) - data structure, data manipulation rules and tools to ensure integrity. At the stage of formulating action storage conception in CoIS we will focus on the structural aspect. The use of classical approaches for creating configurable information systems invariably leads to complication turnkey solutions from a technical point of view (Shaw and Garian, 1996.). Despite the apparent simplicity of operation configurable systems, their support without the involvement of technical experts is difficult.

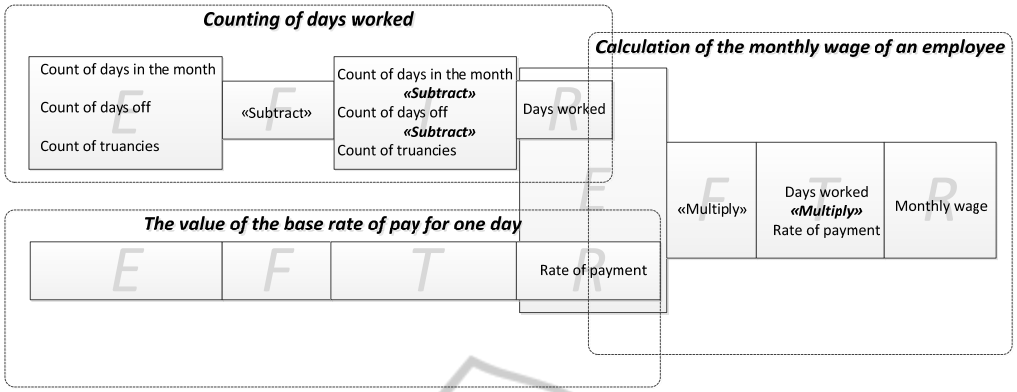

Figure 2: Action type "calculation of the monthly wage of an employee".

One reason for the complexity of created solutions is no single view (abstraction) on an IS. Consequence of this is multiple interfaces to make an adequate state of system components (e.g., ORM-layer, resulting a relational database adequate to the object model). The system should be presented in a unified manner to resolve this issue. This manner could be a base abstraction, which is a reflection of user's actions. Base abstraction should be used to present both logic of interaction with user, stored action results and configuration of the system itself. The actions that can be performed, but does not include values of characteristics represent the system configuration. Actions that have been performed with certain values of characteristics - represent the result of user work with the information system (Rogozov, 2013).

Consequently, the model of action representation in configurable information system, being adequate to unified BAfAS model, should use the action as a basic storage unit. From the structural aspect in the model of action representation can be distinguished: action type, action (performed), constant.

Action type is a reflection of the user's work unit, expressed by:

- used elements:
- applied functions;
- tool that regulates rules of applying functions to elements;
- the result that can be obtained by performing the action.

Result corresponds to the purpose of action implementation. It should be noted that the action type has only specified the expected outcome in the form of performing goals, while the value of the result appears after the action implementation. Action type is the basic unit of storage and representation in configurable information system. An example of the user's action type can be "calculation of the monthly wage of an employee" (Fig. 2).

As elements the action type uses information on number of days worked (which in turn is the result of Action type "counting the number of days worked) and information about the cost of one working day (which is the constant - action type with non-specified elements, functions, and tools, but with specified result value). As functions the action type uses the mathematical operator "multiplication". The tool in this case is a multiplication rule, included in one of the system modules.

In case of representing a system as actions erased a clear separation on the "data", "business logic" and "interface". The IS model becomes more adequate to picture of the real world. Described action type can be repeatedly reproduced to calculation of the monthly wage. In such context, the action type is much like the concept of the function in a programming language, but unlike function the action is not a fragment of code which access data and GUI across multiple interfaces. Action is independent stored element, which can be configured and used as a source of values.

Action is a result of performing the action type at any point in time, containing specific values of all characteristics, including the value of result (Fig. 3). In the context of above described example, the action can be a particular employee wages for a particular month. BAfAS will allow to store not only the wage value, but also its production process. This largely enhances the warehousing possibilities in terms of maintaining the integrity and historicity changes.

Constant is a special case of the action type, which does not have the content of elements, functions and tools, the combination of which allowed us to obtain the result value (Fig. 2).

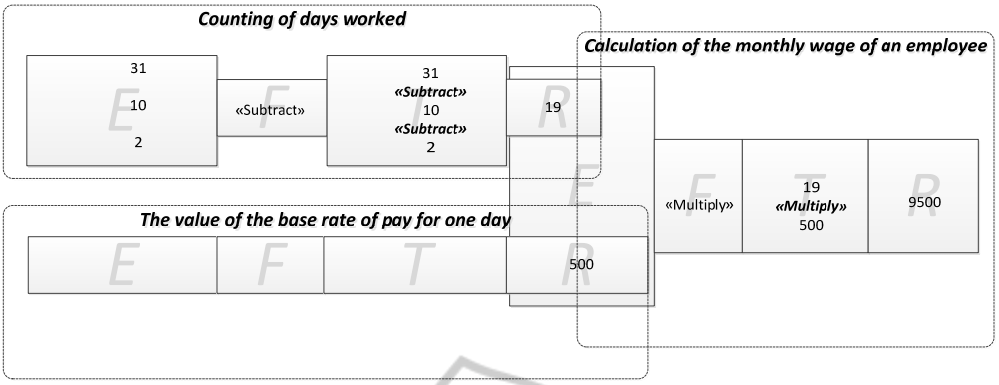

Figure 3: Action "particular employee wages for a particular month".

From structural viewpoint the model of action representation in configurable information system differs from the currently known data models and requires alternative ways to implement. More in detail the differences are presented in the next section.

3 ANALYSIS OF THE APPLICABILITY FOR EXISTING STORAGE TECHNOLOGIES

In order to evaluate the applicability of existing storage technologies for configurable information system we denote key requirements dictated by the model of action representation (Rogozov, Kucherov and Komendantov, 2014):

- Fixed structure of action representation. Regardless of the action content, it must always be presented in four characteristics - elements, functions, tools and results.
- Any of action characteristics actions may have no content or the content may be presented by a plurality of elements with different types;
- As the value type for each of characteristics (except the result) may be action results or a whole action. Herewith in particular action can be used as single types, and various combinations thereof.

The basic storage unit should meet these requirements, because in the other case it will again cause a problem complicating the system (Rogozov, Sviridov and Grischenko, 2014), and may also exhibit negative performance (Grishchenko and Rogozov, 2014).

Table 1: Results of analysis.

Requirement	Relational technology	Object-oriented technology
Fixed structure	Unavailable. The structure of relationship is determined at the time of its setting on domains, the number of which can be arbitrary	Available. Class is always represented by the name, attributes and methods
Multiple values of characteristics	Unavailable. Relation instance contains one value of each domain	Partially available. Class instance is characterized by a specific attribute values. The problem can be solved by complex data types
The multiplicity of values types for a single characteristic	Unavailable. Each domain is a separate data type	Partially available. Class attribute has a specific value type. The problem can be solved by complex data types

From the many created for today data storage technologies clearly the declared basic storage unit only have two - relational and object-oriented. In the first case the basic storage unit is a relation that allows to group and record the individual facts. In the second case, the basic storage unit is a class that contains a set of attributes and operations. Results of analysis mentioned above data storage technologies from the viewpoint of BAfAS implementation are shown in Table 1.

None of the submitted data storage technology is not able to meet the requirements. This is a consequence of the narrow focus and it affects the final form of the base storage unit (Kucherov, Rogozov and Borisova, 2014).

To meet the model requirements configurable information system's storage technology can be obtained in one of two ways:

- development of its own storage technology which has adequate representation in the form of action;
- use of existing storage technologies, allowing to independently define the base storage unit structure.

First method is more efficient in terms of the final result, second method has fewer risks and deadlines. As part of the study is expected to first carry out optimization and testing of the model on existing technologies, and then go on to create their own technology.

For today there is a whole class of models and data storage technologies, combined by the term NoSQL (Sadalage and Pramod, 2012). These solutions include next types: family of columns, key-value, document-oriented etc. Because information systems are focused on factual information storage, document-oriented ways of presenting information is not applicable, in addition, they require additional processing after extraction of information (document) from the storage. Actions have a clear structure and themselves, in turn, form a complex structure - user activity. Therefore, the key-value technology is not applicable too, since storage is not natively focused on associated data objects. Mentioned requirements are met by column family-based types of NoSQL storage.

NoSQL storages oriented to work with families of columns have different ways of presenting data (McCreary and Kelly, 2013). Of greatest interest are so-called schema-free systems (Tweed and George, 2010), wherein the base storage unit has no deterministic structure of representation. The unit structure can be initially set in the adequate to configurable information system form. Such systems use basically principles laid down in the hierarchical and network data models - each data object is a sparse hierarchical tree of elements. Next, consider the conceptual model of the configurable information system's storage based on NoSQL-technology.

4 CONCEPTUAL MODEL OF THE CONFIGURABLE INFORMATION SYSTEM'S STORAGE

Both action types and actions must meet BAfAS requirements. In this connection, convenient to introduce the storage concept as a multi-layered storing system of actions. (Fig. 4). The base of the storage (also the basic storage unit and the first layer) is an BAfAS, consisting of four characteristics - elements, functions, tools and results.).

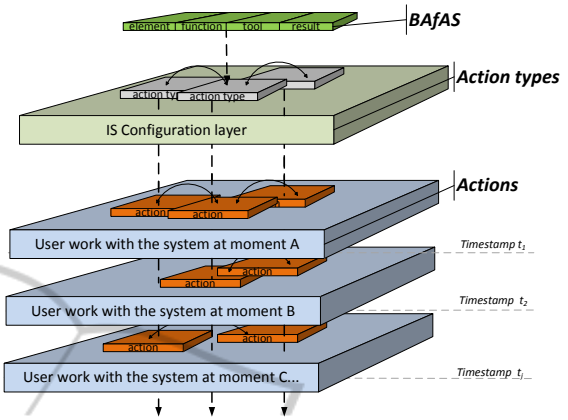

Figure 4: Conceptual model of the configurable information system's storage.

The next layer is the system configuration, expressed by action types. Each action type is the result of applying the basic abstraction to non-automated user activity. Action type contains elements, functions and tools, which implementation will lead to obtain the target result. Every action type can be repeatedly reproduced thereby will be memorized actions reflecting the user work with the system.

The system configuration representing by connected action types (base layer) are created once at the configuring stage. During the IS operation we are building up the layers reflecting results of a different action types in variable contexts (secondary layer). Each secondary layer is a result of action types implementation fixed at a certain point in time. Horizontal upbuilding of the base layer occurs during system configuration by adding new action type. Vertical upbuilding occurs in IS operation (in new times we perform action types from the base layer). Multi-layered character of the storage is a consequence of using the unified abstraction in the process of creating and operating the system. Next consider models of action type, action and constant taking into account the NoSQL technology.

Fig. 5 shows the model of BAfAS representation in NoSQL configurable information system's storage.

Action type is formed through the common pattern (base abstraction for action synthesis), which includes the name of the action, and a list of four characteristics with their constituent components. Result in the description of action type is degenerate, since the action is described, but not yet implemented.

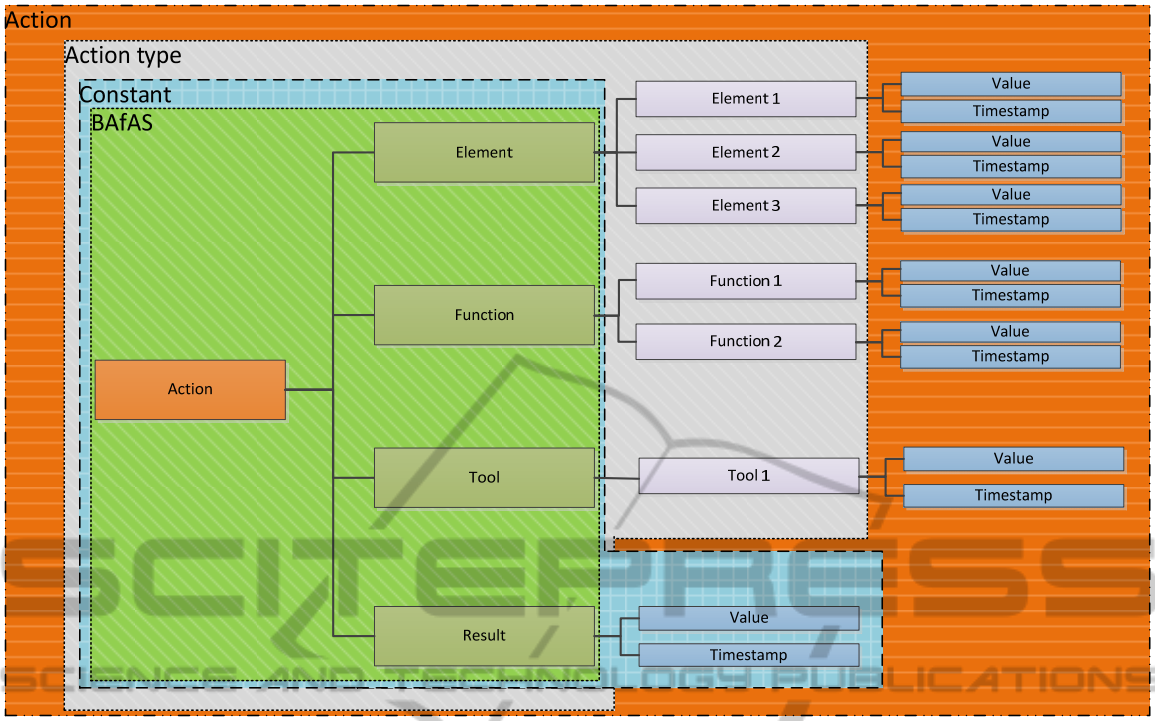

Figure 5: Model of BAfAS representation in NoSQL configurable information system's storage.

Due to schema-free technology the number of components and their types can be determined dynamically.

Action, in contrast to an action type, has values of characteristics. Each value is accompanied by a time-stamped obtaining or assignment. Action also store the result of its implementation, and the timestamp of result value is considered as the moment of action implementation. Constants are a particular example of an action type, because they are recorded at the configuration stage. They have degenerated characteristics of elements, tools and functions and contain only the result with the timestamp.

As we seen in Fig. 5, each base storage unit is a named tree that describes the user action. Inside a node may be hierarchy describing the embedding of actions at each other and allowing to describe complex user activities.

Described above components are basic components of the conceptual model of the configurable information system's storage. Storage itself is done at the expense of hierarchical trees constructing. Next, the example from of the first part shows the general concept of the storage (Fig. 6).

Storage is logically divided into two aspects - configuration and operation.

- The configuration aspect is those components that can be reused for the reproduction of user activity. This includes action types that reflect the user's work. They comprise the specification of characteristics (elements, functions and tools), but they do not contain specific values. Also the configuration aspect include constants, as they contain pre-defined during configuration values and do not reflect the user's work.
- The operational aspect is action types from a configuration aspect, supplemented by value of specified characteristics and implementation results. Thus the operational aspect represents the actions performed by the user within information system. Each characteristic's value is accompanied by a timestamp. The action implementation time is determined by the result's timestamp.

Drawing an analogy with object-oriented and relational technologies it can be said that the action type (by destination) is like a class (entity), and the action is like an object (entity instance). However, unlike entities and classes the action storage model has significantly greater opportunities:

- it is more dynamic;
- it has more semantics than the current data models;

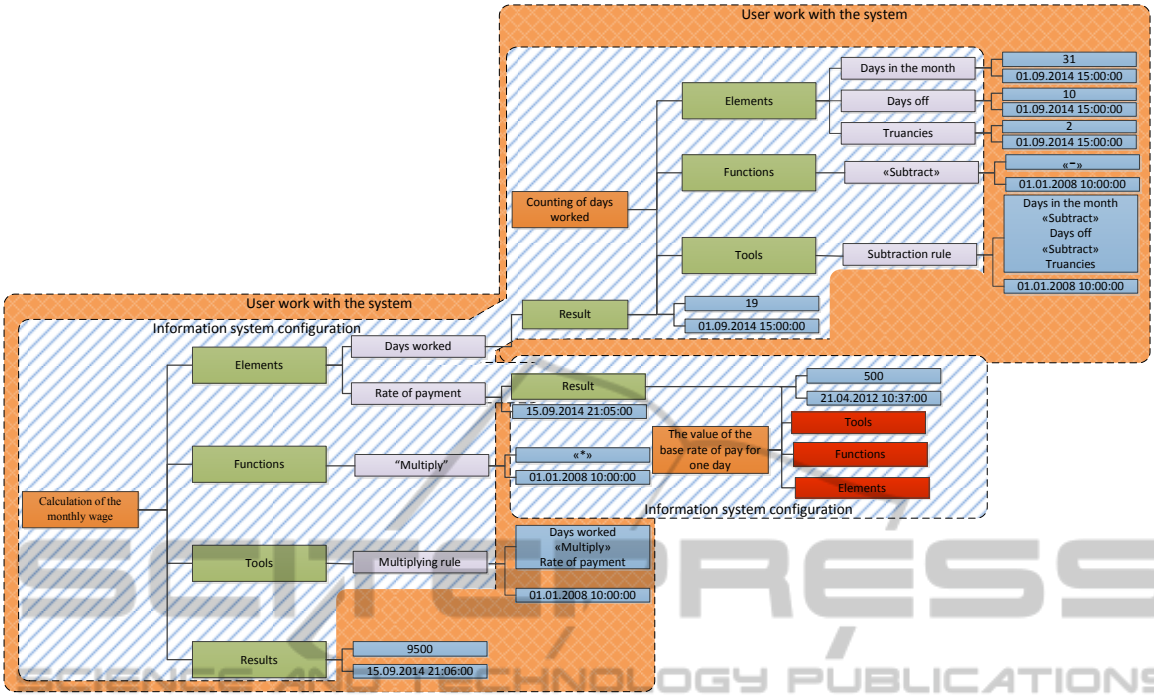

Figure 6: Example of filling NoSQL storage of configurable information system.

- it is more adequate to picture of the real world.

Main ideological difference of the proposed storage is that it is not based on a set of separated actions, but on BAfAS, which allows to receive various types of user actions for further reproduction.

- The proposed storage allows the use of different types of characteristic's values, thereby building a sequence of action types, the implementation of which leads to a result (data fact). By using BAfAS we met requirements of the action representation model in configurable information system, and also the conditions for its implementation without additional funds to bring an adequate form (interfaces). The possibility of "growth" at the expense of schema-free allows unlimited detalization of actions. It also allow to describe complex activities.

5 MANUSCRIPT PREPARATION

The paper proposed the base abstraction of action synthesis (BAfAS) and conceptual model of configurable information system's storage. The model focused not on the storage of certain facts, but on storage of action for their preparation.

Representation of configurable information system, storing of its configuration and its operating results as a set of actions will eliminate currently existing problems, such as: growth of the technical complexity with enhancing flexibility, misunderstanding between end-user and developer, extraction requirements, etc.

BAfAS allows to bring the storage to a new level - data semantics, their occurrence are clearly documented in a readable form for an user. This in turn has a positive effect on both the modeling process, and the process of operation and improvement of the system by end-user.

The model of action representation in configurable information system and storage model proposed in this paper have the following advantages:

- Storing data in an adequate form to information system model;
- Eliminate the need for additional mechanisms of transformation and interfaces between the storage and information system.

The use of NoSQL technologies, in turn, will allow to realize high-performance solutions devoid of drawbacks structure-independent databases (Kucherov, Samoylov, 2014; Kucherov, Sviridov, 2014).

ACKNOWLEDGEMENTS

The research is performed within the government mandate № 0110021005901621. Theme № 213.01-11/2014-17 "Development of methods for data storage creation in configurable information systems and mechanisms for their implementation".

REFERENCES

Rogozov Y., Degtyarev A., 2014a. *The basic foundation of software framework for configuration underwater acoustic information systems with dynamic structure*. Information and Communication Technology for Education (ICTE-2013).WIT Press. Southampton, Boston, Vol. 58., p.181-189

Garcia, J., Goldszmidt, G., 2007. *Building SOA composite business services*
http://www.ibm.com/developerworks/webservices/library/ws-soa-composite/

Rogozov Y., Degtyarev A., 2013. *Evaluating the effectiveness of building a software sonar information systems using configurable software framework.* Engineering Gazette Don. http://www.ivdon.ru/magazine/archive/n4y2013/1877.

Sviridov, A., 2014. *Configuration of information systems in terms of control systems.* News of Southern Federal University. Technical sciences. №6, P.168-173

Kucherov S., Rogozov Y., Sviridov, A., 2014. *The method of configuring dynamic databases.* WIT Transactions on Information and Communication Technologies, 2014, Volume 58, Boston, WIT Press, p.163-173

Rogozov, Y. I., 2013. *Approach to the definition of a meta-system as system.* Proceedings of ISA RAS. - 2013. - Vol. 63., №4 / 2013. - P. 92-110.

Rossi, C., Guevara, A., Enciso. M., 2010. *A tool for user-guided database application development. Automatic Design of XML Models using CBD*. Proceedings of the Fifth International Conference on Software and Data Technologies, ICSOFT 2010, Volume 2, p.195-200

Rogozov Y., Sviridov A., Belikov A., 2014. *Approach to CASE-tool building for configurable information system development*. Information and Communication Technology for Education (ICTE-2013). WIT Press. Southampton, Boston, Vol. 58. p.173-181

Lazarev, V., Rogozov Y., Sviridov A., 2014. *Methodological approach as the successor of object approach in development of information systems.* Journal of Informatization and Communication 2'2014, P. 85-88

Rogozov Y., 2014a. *Methodology of creation of subject-oriented systems.* Journal of Informatization and Communication 2'2014, P. 6-10

Rogozov Y., Sviridov A., 2014a. *Approach to the construction of information systems based on the methodological approach*. // Proceedings of II International conference Innovative technologies and didactics in teaching. Vol 1.- Taganrog: Publishing House SFU, 2014. S. 3-9

Rogozov Y., Sviridov A., 2014b. *Methodological concept of building information systems.* Journal of Informatization and Communication 2'2014, P. 11-14Rogozov Y., Degtyarev A., 2014b. *The method of configuring the functionality of software sonar information systems.* News of Southern Federal University. Technical sciences. №1. – P. 13–18

Fischer, G,. Giaccardi, E.,2006. *Meta Design:A framework for the future of end user development. End User Development: Empowering People to flexibly Employ Advanced Information and Communication Technology*. H. Lieberman, F. Paterno and V. Wulf, Springer. 9: 427-457

Rogozov Y., 2014b. *The general approach to the organization of certain system of concepts based on the principle of generating knowledge.* XII All-Russian Conference on Control VSPU-214. Moscow, 16-19 June 2014: Proceedings. ISBN 978-5-9145-151-5., pp 7822-7833.

Kucherov, S., Lipko, J., Schevchenko, O., 2014.*The integrated life cycle model of configurable information system*.Proceedings of IEEE 8th International Conference on Application of Information and Communication Technologies - AICT2014. IEEE Catalog Number CFP1456H-PRT, ISBN: 978-1-4799-4120-9 2, p. 182-186"

Kucherov, S., 2013. *User-configurable information systems as a means to overcome the semantic gap.* Journal of Informatization and Communication 5'2013, P. 135-137

Kucherov S., Samoylov, A., Grishchenko,A., 2014. *Flexible database for configurable information systems* .Proceedings of IEEE 8th International Conference on Application of Information and Communication Technologies - AICT2014. IEEE Catalog Number CFP1456H-PRT, ISBN: 978-1-4799-4120-9 2, p. 374-377"

Kucherov S., Sviridov, A., Belousova S., 2014. *The formal model of structure-independent databases*. Proceedings of 3rd International Conference on Data Management Technologies and Applications, Vienna, Austria, Scitepress - Science and Technology Publications. ISBN: 978-989-758-035-2. p. 146-152

Codd, E.F.,1981 *Data Models in Database Management*. Proc. Workshop in Data Abstraction, Databases, and Conceptual Modelling (Michael L. Brodie and Stephen N. Zilles, eds.), Pingree Park, Colo. (June 1980): ACM SIGART Newsletter No. 74 (January 1981); ACM SIGMOD Record 11(2), February 1981; ACM SIGPLAN Notices 16(1), January 1981

Shaw, M., Garian, D., 1996. Software architecture: perspectives on an emerging discipline // Prentice Hall, Englewoo Cliffs, NJ, ISBN:0-13-182957-2

Rogozov Y., Kucherov S., Komendantov, K., 2014. *The methodological approach to creation of user data structures in user-configurable information systems* Innovative technologies and Didactics in Teaching:

collected papers. - Berlin: MVB Marketing- und Verlagservice des Buchhandels GmbH, 2014, p. 171-184

Rogozov Y., Sviridov, A., Grishchenko, A., 2014. *The method of data manipulation operations representation as a structure in structure-independent databases oriented on configurable information system development.* WIT Transactions on Information and Communication Technologies, Volume 58, Boston, WIT Press, pp. 189-197

Grishchenko, A., Rogozov Y., 2014. *Criterial rating of the data manipulation methods for configurable information system development*. Innovative technologies and Didactics in Teaching: collected papers. - Berlin: MVB Marketing- und Verlagservice des Buchhandels GmbH, p. 46-52

Kucherov S., Rogozov Y., Borisova, E., 2014. *Structure–independent databases modelling*. Proceedings of IEEE 8th International Conference on Application of Information and Communication Technologies - AICT2014. IEEE Catalog Number CFP1456H-PRT, ISBN: 978-1-4799-4120-9 2, p. 192-196

Sadalage, Pramod J., Fowler, M., 2012. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, ISBN-13: 978-0321826626.

McCreary, D., Kelly, A., 2013. *Making Sense of NoSQL: A guide for managers and the rest of us*. Manning Publications, 312 p. ISBN 978-1-61729-107-4.

Tweed, R.; George, J., 2010. *A Universal NoSQL Engine, Using a Tried and Tested Technology*. http:// www.mgateway.com/docs/universalNoSQL.pdf.