THE INNOVATIVE APPROACH TO INTELLIGENT HELPDESK SYSTEMS DEVELOPMENT

Vladimir Milićević[1]*, Slavoljub Milovanović[2]

[1]Belgrade Metropolitan University, Tadeuša Košćuka 63, 11000 Belgrade, Serbia, vladimir.milicevic@metropolitan.ac.rs
[2]University of Niš, Faculty of Economisc in Niš, Trg kralja Aleksandra Ujedinitelja 11, 18000 Niš, Serbia

REVIEW SCIENTIFIC ARTICLE

ISSN 2637-2150
e-ISSN 2637-2614
UDK 004.78:[004.5:004.891.2
DOI 10.7251/STED1902031M

Rad primljen: 03.11.2019.
Rad prihvaćen: 22.11.2019.
Rad publikovan: 29.11.2019.
http://stedj-univerzitetpim.com

Korespodencija:
Vladimir Milićević, Belgrade Metropolitan University, *Tadeuša Košćuška 63, 11000 Beograd, Srbija.*
vladimir.milicevic@metropolitan.ac.rs

ABSTRACT

The paper examines the current state and the possibilities of using contemporary approaches and tools in HelpDesk systems development with the aim to examine its disadvantages. Disadvantages are accurately analyzed and the paper gives clear guidance how to eliminate them. It has been noticed that current design and development approaches have their own advantages and disadvantages. The application of the unified approach that combines the best individual characteristics of the existing approaches enables development of an automated HelpDesk system that is capable of expanding its own expertise. By combining unified aspect approach with the zero tolerance model driven approach, a new innovative approach to the development of a wide range of expert systems is obtained. This innovative approach is characterized by the highest possible degree of modularity.

The application of the aforementioned approaches has also enabled expansion and improvement of the existing expert systems used for servicing numerous companies operating in various fields of business.

Key words: HelpDesk systems development, object-oriented approach, aspect-oriented approach, intelligent information systems

INTRODUCTION

In today's business environment, the application of modern information technologies is crucial for each participant in the global market. Together with the products and services, the software solut/ions used for supporting businesses are becoming more and more complex. Therefore, modern tools and techniques for software development require constant expansion with the aim of designing and developing software solutions that will fully satisfy current customer requirements and give companies a strategic advantage in the market. The paper identifies service industries (e.g. insurance industry) as the key market place where these systems can be fully applied. HelpDesk system, integrated into information systems of the insurance companies, contributes to a large extent to quality of products and services and increases the level of general satisfaction of employees and service users. For this purpose, tools and techniques for developing intelligent HelpDesk system have been identified through expansion of modern object-oriented technologies by tools that allow the dynamic manipulation of the new knowledge by the information system. Two well-known aspect-oriented approaches have been tested and their dominant characteristics have been combined, thus creating an original unified approach that helped to achieve the intended goals of the paper. In the paper, the implementation of the HelpDesk system in synergy with intelligent programs has been especially considered. The work of such a system is fully automated and expertise is greatly shifted from a man to a computer. By choosing the optimal algorithm for searching a database of questions, HelpDesk system is equipped with a powerful tool – neural network, which fully enables the aforesaid software functionalities.

In that context, the paper is organized in six sections. After introduction section, problem of development of contemporary intelligent information sytems with emphasis on HelpDesk systems is defined. Third section is dedicated to the previous researches of the problem and outline of a new software solution. Fourth section presents innovative development approach based on aspect-oriented software development that enables improvement of contemporary HelpDesk systems. Improvement of intelligent HelpDesk system in the case of insurance company achieved by the innovative development approach is analysed in fifth section. Final section presents conclusions regarding innovative development approach to building contemporary intelligent information systems like the HelpDesk system of insurance company that is taken as a case.

DEFINING THE PROBLEM OF INTELLIGENT HELPDESK SYSTEMS DEVELOPMENT

The last decade brought big changes in how to create and use business software. According to a number of authors (Ballou 2008), business software is the foundation of success in the competitive environment of the global economy. In contemporary industry software, development of software solution, and later its design and implementation, increasingly insist on the use of intelligent computing techniques. The application of these techniques can be found in the solutions to search large databases or Big Data (Curry, Kirda, Schwartz, Stuart, Yoran, 2013), predicting the movements of the major markets (Market Intelligence, 2016), and the transfer of certain management roles to such software systems (Sousa & Effy, 2014).

According to the group of the authors (Dunie, Schulte, Cantara, & Kerremans, 2015), business managers, and experts daily faced with the key challenge. They are expected faster and better decision making to support the modern concept of business - do more for a shorter period of time. Struggling with these challenges is crucial to the implementation of intelligent software solutions to support business operations. Modern information systems use more and more intelligent support for numerous business processes such as automation of hardware; production planning; planning and design of business processes; reducing system complexity; management of financial flows; inventory management; control of equipment and inventory; management of effective communication between employees, etc.

Of particular importance for this work is the identification of options for engagement of intelligent software tools and techniques to support communication between officers of the company and the management of different levels, and between service users and staff. Timely and prompt information in modern business is crucial to quality of this business. The component of information system that enables exchanging of information by scheme question-answer between the experts and other users of the system is called the HelpDesk.

Contemporary HelpDesk systems generally work on such way that the question posted on the page request is forwarded to the appropriate expert to review and respond. From the moment of asking questions to get answers may take a long time, and this can result in a delay in the conduct of business, which is in modern business conditions unacceptable. When designing advanced software solutions developers often have tasks that have a common denominator - designed and implemented software must allow minimization of errors in operations caused by human activities and thereby ensure precise placement of current information. Thus, implemented software should provide a strategic advantage to the company that introduced it in its business. Guided by the above theses, this paper has focus on the analysis of the possible directions of improvement of modern HelpDesk system in service areas such as the insurance industry and creating their own solutions for the automation of their operation, relying on intelligent search techniques. Companies in the service areas lack such systems which would enable the identification of patterns of behavior of employees and service users, and therefore efficient management without the need for intervention by the various levels of management. In this way, it opens up a great field of study of intelligent information systems that can independently recognize new facts, create new knowledge based on the facts that automatically can be incorporated into its functionality without the intervention of a software development team. Therefore, it is necessary to innovate approach to the design and coding of application software for HelpDesk systems.

Of particular importance is the search for new tools and techniques which will enable dynamic knowledge management. Contemporary HelpDesk systems usually present the knowledge to users in the form of FAQ (Frequently Asked Questions) library. Enabling dynamic control of knowledge, HelpDesk system will be able to immediately respond to questions without human intervention – expert intervention. Also, by providing pattern - rule class obtained on the basis of existing expertise - knowledge base will be able to auto-expand with new specialized rules - rule objects. In this way intelligent information system capable of self-learning can be structured.

PREVIOUS RESEARCHES OF THE PROBLEM AND OUTLINE OF A NEW SOFTWARE SOLUTION

By the analysis of modern HelpDesk solutions, as well as the current object-oriented software tools, it is possible to reach the following conclusion: the knowledge base of this expert system, where HelpDesk system would find answers to customer questions, can not be created by default *if - then - else* scenario, because it is executable at the level of implementation of the software solution and can not result in the automatic building in of new knowledge into existing software solutions. Thus, the rules must take the form of class i.e. objects in a very loose connection with the class of existing software solutions that make it possible to think about their dynamic implementation in the existing software solution. As a possible direction in which to look for a solution, there is a young technique of designing and programming known as Aspect Oriented Software Development (AOSD), wherein of special relevance is works of the following group of authors:

1. The authors gathered around Gregor Kiczales (Kiczales et al. 1997; Kiczales et al. 2001) from Xerox - USA;
2. The authors gathered around Kiselev, 2003.
3. The authors gathered around Maja D'Hondt and Viviane Jonckers from Vrije University in Brussels, with special emphasis on the work by María Agustina Cibrán (Suvée, Vanderperren, Wagelaar, Jonckers, 2005.; Suvée, Vanderperren, & Jonckers, 2003; Cibrán, Suvée, D'Hondt, Vanderperren, & Jonckers, 2004; Cibrán, D'Hondt, & Jonckers, 2003; Cibran, 2007).

By research and analysis of works of the above-mentioned groups, it is possible to get a starting point for the development process of the planned software solution. The first two groups of authors have experimented with the aspect-oriented tool known as AspectJ. Program code that is obtained by using this tool is almost identical to the Java code, extremely rapidly executed and linked, but had no ability of dynamically binding to an existing application. Detailed analysis of this tool is possible through the research works of the following authors: V.B. Griswold, E. Hilsdale, J. Hugunin, M., Kersten, J. Palm, G. Kiczales and the others from a Xerox Institute in the USA (Kiczales et al., 1997; Hilsdale, & Hugunin, 2004; Kiczales et al., 2001). In addition to these authors, this tool is studied by Hunt, 2006, and Kiselev, 2003. All these authors in their works represent a new form of software called *Council*, which is an innovative answer of the knowledge base on the request of the main control part of software solution. The authors from the Vrije University (Suvée, et al. 2005; Suvée, et al., 2003; Cibrán, et al., 2004; Cibrán et al., 2003; Cibran, 2007), experimented with a code which is known as a JasCo. Its features include: slower connections and execution, but the ability to dynamically binding to the existing software solution in scenarios: *before, around and after*, which are related to the time of application of the rules corresponding to the answer to question that is sent to HelpDesk system.

It is noted that each of these approaches has its advantages and disadvantages. Since it is based on the logic of open source, by additional development it is possible to combine the best individual characteristics and get the ideal tool for coding object rules. Thus, it is possible to use AspectJ as the basis for creating classes and objects rules in HelpDesk knowledge base that is expanded by the possibilities of dynamic application according to these scenarios. This would represent an original solution in the field of Aspect-oriented software development. This approach can obtain the name of the AspectJ + that was used in this paper. After selecting the optimal tool for the creation of a HelpDesk system knowledge base, it is necessary to define an innovative approach to the development of an information system that would be capable to dynamically manages its own knowledge. Innovation is reflected in the fact that modern object-oriented tools have not been able to independently cope with these problems, and it is necessary to use certain extensions such as Aspect-oriented tools. In this light, the path to the desired software solution is going in the following directions:

- Consideration of the author María Agustina Cibrán (2007), are respected and extended by introducing of a new domain language. At a high level, domain level, rules of HelpDesk knowledge base have been designed and coded by new domain language. By translating domain code in the implementation, we get the code of the highest possible degree of modularity, and this results in a possibility of dynamic integration of rules, tremendous speed of execution of software code and efficient removal of identified errors during testing;
- For searching objects of the knowledge base and discovering new knowledge, monotonic neural networks and optimal training algorithm are programmed. About this issue it is possible to make a detailed analysis on the basis of the work of the following authors: Rojas, 1996. and Cilimkovic, 2011. On the basis of the perceived shortcomings of their approaches and the shortcomings identified during the two years of exploitation of this new intelligent software solution (Milicevic, 2012), it is possible to find a better solution with optimal training algorithm;
- A software component called NDL/Generator that will code newfound knowledge by descriptive domain language is developed;
- A software component called Translator/AspectJ+ which will translate above described code into the objects rules and transfer it to the dynamic incorporation into the knowledge base is developed.

A particular challenge in creating this innovative software solution is to find an optimal algorithm for training of search mechanism based on the monotonous mathematical function by which weighting factors of nodes in the hidden layer of neural network will be determined. Then, by using statistical tests (by mean square error) it should choose the ideal architecture of search mechanism which is reflected in the optimal number of nodes in the hidden layer of neural network. Of particular interest is the use of the selected search algorithm combined with the principle of recognition of natural languages (adjusted for new domain language rules), Zipf's law and the method of reserved words. Also, in the architecture of the neural network, software details known as *semaphores* (Zelenski, 2008) are built. It will have the task to synchronize processes of search mechanism. This is the way to eliminate the shortcomings identified in the two-year exploitation of this software solution on a large the number of search iterations. In addition, in this paper it is proposed the analysis of the model domain according to MDSD (Model Driven Software Development) approach which will be explained in the following section. All identified class rules from which subsequently objects are created, as well as appropriate links with part of software solutions for knowledge management are coded by domain language, which is very similar to the spoken language. In this way, it is possible the active participation of experts in various fields of software development, but they do not have to be familiar with the tools and programming techniques. This language is indicated as the NDL (New Domain Language) and represents a specialization of high-level language for the application in the coding of knowledge and its distribution through HelpDesk systems. As such, it also represents an original way of applying of high-level language in HelpDesk systems. On its basis, NDL/generator software component is developed. The component has the task to code discovered knowledge by NDL language from sample or set of questions forwarded via the HelpDesk system. The coding of knowledge is done by using the principle of reserved words and Zipf's law. Also, NDL/generator has the task to transfer this knowledge by translating into the code that corresponds to the programming language with appropriate aspect extensions. By creating NDL/generator it is set the basis for development of software components from the class of translator which directly translates new knowledge into executable program code. As we noted, this software component is called Translator/AspectJ+. With the support of this component, specific software transformation by which NDL code is translated directly into Java code expanded with AspectJ+ functionalities is created. In the research, transformation performances for translating relationships of object rules into Jasco and AspectJ+ aspects of the relationship are specifically tested. The test showed that the new AspectJ+ approach is dominant in terms of stability and speed of execution of software transformation.

IMPROVEMENT OF CONTEMPORARY HELPDESK SYSTEMS BY INNOVATIVE DEVELOPMENT APPROACH

Starting from the settings explained in the previous section, possible ways to improve intelligent HelpDesk systems considering their development based on contemporary and innovative approach has been identified and analyzed. As already mentioned, the improvement of these systems is achieved by the aspect-oriented software development (AOSD). The justification for the introduction of this approach to software development is found in inherited defects which refer to the degree of modularity of software and that the standard object-oriented approach has not been able to completely resolve. Therefore, it is shown that only software solution with a maximum degree of modularity and with minimal connections between their own modules can realize the dynamic expansion of the knowledge base without re-translation of software solutions. Thus, the knowledge base should be organized in the form of a set of classes (the pattern rules) from which it is possible to create a large number of different objects rules. A particular problem is the realization of objects by which knowledge is connected with the central class of software solution. This is why it is necessary to choose the most suitable aspect-oriented approach that would enable dynamic linking of the new object rule with the application and keeping its pattern in the knowledge base. In this paper, in particular it is shown that the two-existing aspect-oriented approaches (AspectJ and JasCo) have disadvantages that can be eliminated by combining the best of their individual characteristics. In this way, the original, unified AspectJ + approach is obtained and the definition of object-oriented design pyramid is completed. Of particular importance is the evolution of the software design that relying on AspectJ + in combination with the analysis of the domain model. The evolution moves the object-oriented software development paradigm from MDE (Model Driven Engineering) approach to MDSD (Model Driven Software Development) approach, which can be illustrated as in Figure 1. However, in order to the definition of this unified approach would be fully completed, it is necessary to investigate the relevant advantages and disadvantages of AspectJ and JasCo approach. The criteria by which these approaches are tested is: performances and the ability of dynamic connection of object rules with the managerial part of software solution. Performances are directly linked to the speed of execution and stability of software solutions and related to the speed of response of a rule from the knowledge base, as well as the total duration of its engagement. It was observed that there is maintenance of proportionality between these quantities, for two different object rules by using a unified AspectJ+ approach. It has been proven that AspectJ aspects of the relationship are activated on average 18% faster than the JasCo aspects. That results in 24% less time on average of total duration of the engagement of object rules linked by AspectJ aspects in comparison to the identical object rules linked by JasCo aspects. On the other hand, by AspectJ aspects, dynamic linking of new knowledge is impossible to implement, as in the case with JasCo aspects. By redefining AspectJ aspects in a way that gives them the ability to manipulate patterns (classes of object rules) as in the case with JasCo aspects, mentioned disadvantages of AspectJ approach are eliminated. In this way, the original AspectJ+ approach that has negligible worse performance than the original AspectJ approach, but with the possibility of dynamic linking of newly discovered knowledge with part of software solution for knowledge management is obtained. Classes of the new approach are located in Java Web repository and can be freely used when developing various software solutions.

Innovative concept of software design derived from the domain model involves a two-way mode of software development: development of software code based on the model and reengineering model based on changes in the code. This is of particular importance for the development of quality software solutions, because during the process of domain analysis it is possible actively engage experts who better understand the business than programmers and software developers. By defining high-level language (domain language) on domain level in the form of programming language that is close to english language, the presentation of business rules and their relationship with the core of software solution is allowed. A precise definition of the grammar of the language enables linking of its instructions with instructions built into the appropriate object rules. The linking is achieved by special Java classes that support the software transformation related to translation of model code to executable code of the programming language. This is a key part of improving HelpDesk system, because it sets the basis on which an intelligent algorithm searches the text sample (set of questions) and by a pattern (a set of classes in knowledge base) gives the answer to the question asked via HelpDesk system.

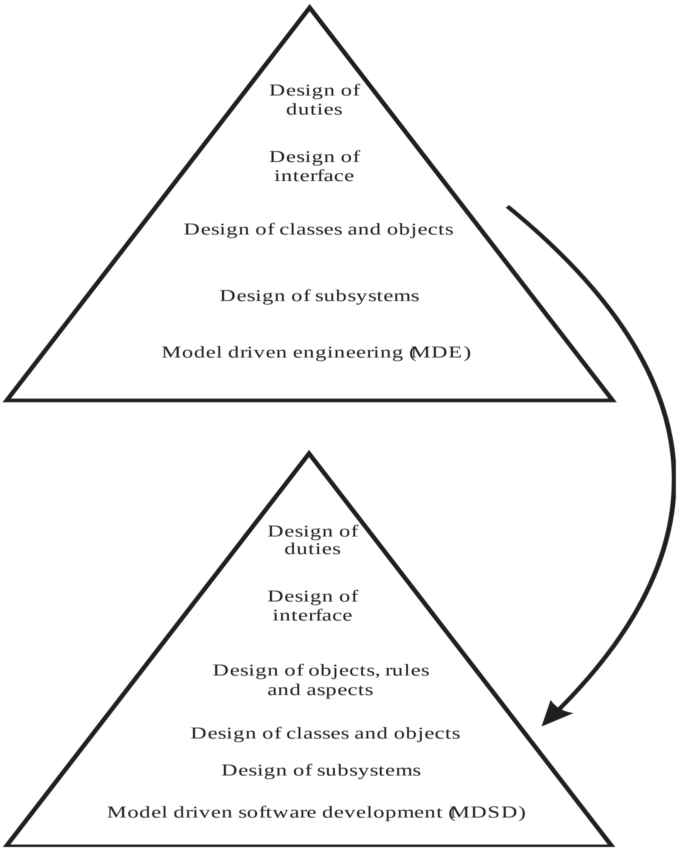

Design of duties
Design of interface
Design of classes and objects
Design of subsystems
Model driven engineering (MDE)
Design of duties
Design of interface
Design of objects, rules and aspects
Design of classes and objects
Design of subsystems
Model driven software development (MDSD)

Figure 1. Evolution of the extended pyramid of object-oriented software development

In the development of innovative software solutions for HelpDesk systems, in addition to object rules and links, the concept of the event or a scenario under which the business rule is activated and comes to the application of its logic by central class of software solutions is introduced. To start rules apropos creation of an object rule from the appropriate pattern, the trigger class is used. The trigger class manipulates the following values: activation of the link, time of application of the rule and the time of joining the necessary information to rule. Especially, in order to a text sample (set of questions), through the code of high level, could be expressed as an executable knowledge (classes and rule objects), it is necessary to define quality and complete set of software transformations for translation of the language domain into the executable Java and AspectJ+ program code. A set of classes enables automatic translation of business rules and links, expressed by domain language, into the corresponding Java object rules and AspectJ+ link aspects. This set of classes builds a unique software component of HelpDesk system, called Translator/AspectJ+. Also, these classes are available to developers via Web Java repository in which are housed after testing. In proving the benefits of this software solution, of particular importance was the performance testing of translator with

AspectJ+ functionalities in relation to the Translator/JasCo which was created on the basis of transformation proposed by Cibran, 2007. It was noted, and shown that AspectJ+ transformations use, on average, about 10% less processor time, which is another advancement in development of the mentioned classes of software solutions. It should be noted that for each of the high-level NDL instruction, specific transformation is defined. The instructions received their own object-oriented or the aspect-oriented representation considering the instructions of object rules or aspect links, respectively. In particular, during the transformation of high-level code, it is insisted on the management of software exceptions. In the definition of the class of rules, and therefore objects of these classes, the logic of business rules is implemented through software blocks: *try* ... *catch*. On the basis of the trigger event, the logic of rules is executed with a permanent monitoring of system behavior by appropriate system classes called *listeners*. In this way, the cancellation of the building rules is avoided in the case of unexpected system events, and this results in the improvement of the degree of reliability of classes and appropriate objects of knowledge base. After the presentation of the transformations by which the business rules are translated into objects of rules, and their links to the correspondent aspects, translation of the specific rule by software component of Translator/AspectJ+ is demonstrated. The rule is taken as a question asked through HelpDesk system and analyzed as a text sample. Finally, a business rule is created in the domain language by using software components called NDL/Generator. The output from this set of classes takes a new set of classes (Translator/AspectJ+) that automatically creates executable program code for realizing the answer to this question.

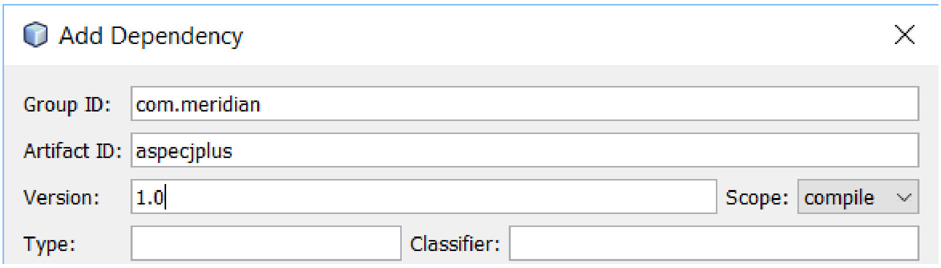
IMPROVEMENT OF INTELLIGENT HELPDESK SYSTEM IN THE CASE OF INSURANCE COMPANY

Development of intelligent HelpDesk system based on described innovative aspect-oriented approach is demonstrated on the example of company in the field of insurance. In doing so, existing known intelligent software solutions with the perceived shortcomings of the search engine are taken into account. With the aim of eliminating the above drawbacks, competitive software concepts called *semaphores* are included. In addition, the architecture of software solution has the following components:

- the server side with the core of application, and databases and knowledge bases;
- PC and mobile client side;
- backup on the cloud server;
- improved search algorithm.

Testing of the functionality of the system is performed by taking concrete rules through HelpDesk system. Two questions are passed through the system with the aim of approving compensation claims from injuries incurred at the workplace and as a result of traffic accidents. The system generated two rules and then sent the answers to the address from which the questions were asked, in a very short period of time. After demonstration of the functionality of automated HelpDesk system, it was necessary to describe a key component of the system which enables search a set of questions and identifying existing and new knowledge. As the ideal solution it is imposed the backwards search algorithm for training a neural network that searches sample. At the outset, it is pointed out the shortcomings of this algorithm, which have been identified through a multi-year exploitation of the specific examples of searching. The main drawback of this algorithm is reflected in the high level of cancellations, after a number of iterations, due to congestion caused by software processes. By integrating the semaphores at key points in the algorithm, we get a complete synchronization of processes, and this results in the execution of the backwards search algorithm, smoothly and without interruption. The special quality to improved the backwards search algorithm gives the fact that after modification, requires 20%

fewer neurons, and thus the software instructions, in the hidden layer of the network to perform the same tasks. This represents a significant shift in performance compared to a conventional backwards search algorithm whose implementation can be found in numerous scientific and professional literature and in various examples. In order to provide the necessary maximum degree of modularity of this information system, a tool for object-relational mapping by which the data retrieved from the database is delivered directly to the objects of core software solution is used. In this way, the traditional relational database management is replaced by the object-relational tool (Hibernate ORM) to achieve the software functionalities. HelpDesk system in the mentioned insurance company has implemented by a prototype method. The first, the functionality of managerial part on the server side of information system was achieved, and subsequently two client components (PC and mobile clients) were implemented. Thus, developed system was tested under two scenarios: 1) the response time to questions asked via the HelpDesk system without intelligent support; 2) response time to questions asked via the HelpDesk system with intelligent support. Through the client software components, the 1000 different questions are sent to HelpDesk system. In the first case, without intelligent support, about 20% of the total number of questions is not resolved until the end of the current business day and longer. The system with intelligent support has solved all issues by processing which lasted less than 10 minutes. Obviously, the software solution with intelligent search proved dominant qualities that lead to: a higher degree of efficiency and effectiveness of business; lower operating costs; a higher degree of satisfaction of system's users based on higher speed of getting information and a short time of solving their requests etc. Finally, in the case of insurance company, the methodology of development of business software solutions by advanced object-oriented tools with the aspect extensions is demonstrated and the upgraded intelligent HelpDesk system is implemented. In addition, a unified approach to define rule objects through the combination of dominant characteristics of existing approaches is developed. Therefore, it is possible to extend and improve the existing expert systems by proposed tools and techniques. Also, an innovative approach to the development of a wide range of expert systems is defined. Following figures are showing the process of including dependencies, developed in this approach, which are essential for this type of software solutions.



Group ID: com.meridian

Artifact ID: aspecjplus

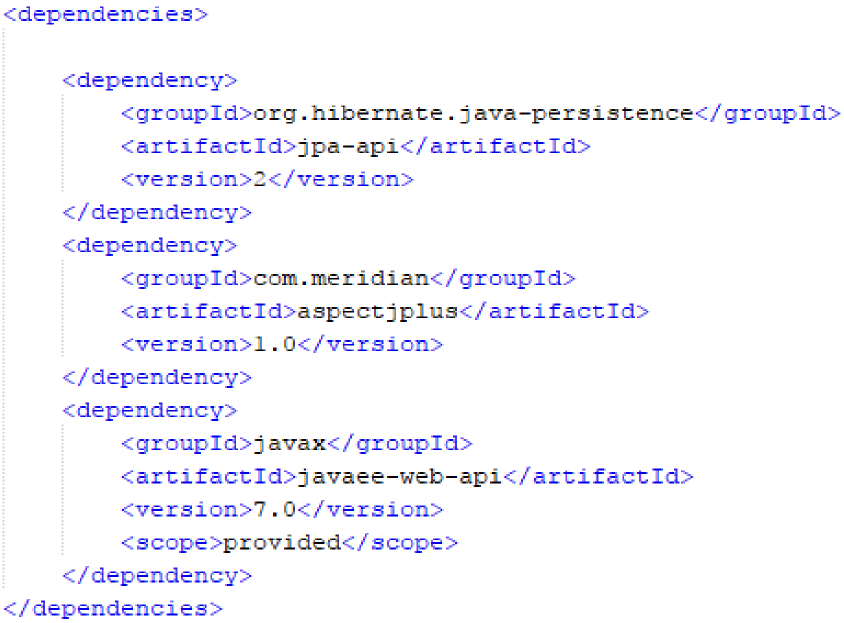
Version: 1.0

Scope: compile

Type:

Classifier:

Figure 2. Adding of the AspectJ+ prototype dependency



```
<dependencies>

    <dependency>
        <groupId>org.hibernate.java-persistence</groupId>
        <artifactId>jpa-api</artifactId>
        <version>2</version>
    </dependency>
    <dependency>
        <groupId>com.meridian</groupId>
        <artifactId>aspectjplus</artifactId>
        <version>1.0</version>
    </dependency>
    <dependency>
        <groupId>javax</groupId>
        <artifactId>javaee-web-api</artifactId>
        <version>7.0</version>
        <scope>provided</scope>
    </dependency>
</dependencies>
```

Figure 3. List of essential dependencies, in code, for this type of software solutions

CONCLUSION

This paper deals with a hybrid intelligent information system capable to automatically recognize new knowledge from the relevant sample (set of questions), to code it and dynamically incorporate into the existing knowledge base (a set of possible answers). The paper is the search for an answer is it possible to provide a technique of designing and programming for the development of information system that from available domain automatically recognizes the new business rules and their relationship, codes them and automatically on the most flexible way, connects them with their own yet functional part for knowledge management.

In this context, of particular importance is the choice of technologies and tools for the development of intelligent HelpDesk system, where two approaches are compared: JasCo and AspectJ. For both approaches, performance of rule response and the duration of the process of rule execution were tested. It is shown that one approach provides better performance (AspectJ) while the second approach (JasCo) enabled dynamic incorporation of new rule objects in existing software solution. By combining the best individual characteristics of these approaches, the original approach called AspectJ+ is created. This approach also has been tested in the same way as the previous two approaches, with the comparison of performance in relation to JasCo approach. Its features associated with dynamic linking of new knowledge and operational software solutions are adopted.

This paper on the example of HelpDesk system of insurance company showed that it is possible to create intelligent information system that automatically recognizes the new business rules from the available domain and their connections, codes them, and automatically on the most flexible way, links them to its own component of knowledge management. By combining the dominant features of a representative aspect-oriented approaches to the development of information systems it is possible to develop improved tool for solving problem of connection of rules with the core of system. Design based on the latest MDSD approach provides expression of executable rules of high level and the corresponding connections.

In addition, the paper has shown that an effective connection and the highest level of modularity can be provided through encapsulation of rules and connections in Java rule objects and AspectJ+ connection aspects. By backward search algorithm that is improved by application of semaphores, and the method of reserved words, it is possible to train a neural network to automatically identify the business rules from the relevant sample.

REFERENCES

Ballou, M.C. (2008). *Improving Software Quality to Drive Business Agility*. IDC – White Paper.

Cibran, M. (2007). Connecting High-Level Business Rules with Object-Oriented Applications: An approach using Aspect-Oriented Programming and Model-Driven Engineering. *Phd Tesis Universiteit Brussel*.

Cibrán, M. A., D'Hondt, M., & Jonckers, V. (2003). Aspect-oriented programming for connecting business rules. In *Proceedings of the 6th International Conference on Business Information Systems* (Vol. 6, No. 7, p. 24). Colorado Springs, USA.

Cibrán, M. A., Suvée, D., D'Hondt, M., Vanderperren, W., & Jonckers, V. (2004). Integrating Rules with Object-Oriented Software Applications using Aspect-Oriented Programming. In *in Proc. of the 5th Argentine Symposium on Software Engineering (ASSE'04),* Córdoba, Argentina.

Cilimkovic, M. (2011). Neural Networks and Back Propagation Algorithm. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin, 15.

Curry, S., Kirda, E., Schwartz E., Stuart W., & Yoran, A. (2013). *Big Data Fuels Intelligence-Driven Security.* RSA Security Brief.

Hilsdale, E., & Hugunin, J. (2004). Advice weaving in AspectJ. In *Proceedings of the 3rd international conference on Aspect-oriented software develop-pment* (pp. 26-35). ACM.

Hunt, J. (2006). Aspect oriented programing with Java. http://www.theregister.co.uk/2006/10/26/aspects_java_aop/. Accessed 20 July 2017.

Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., & Griswold, W. G. (2001). An overview of AspectJ. In *European Conference on Object-Oriented Programming* (pp. 327-354). Springer, Berlin, Heidelberg.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J. M., & Irwin, J. (1997). Aspect-oriented programming. In *European conference on object-oriented programming* (pp. 220-242). Springer, Berlin, Heidelberg.

Kiselev, I., (2003). *Aspect-oriented programming with AspectJ*. Carmel, Indiana: Sams Publishing.

Market Intelligence, (2016). http://www.dobney.com/market_intelligence.htm. Accessed 14 July 2017.

Milićević, V. (2012). *Doprinos razvoju inteligentnog računarskog sistema baziranog na pristupu poslovnih pravila*. PhD Thesys. Faculty of Management, Sremski Karlovci, Serbia.

Rojas, R., (1996). The Backpropagation Algorithm, Chapter 7: *Neural Networks* (pp. 151-184). Berlin: Springer-Verlag.

Dunie, R., Schulte, W. R., Cantara, M. & Kerremans, M. (2015). *Magic Quadrant for intelligent business process management suites.* Gartner Inc.

Sousa, K., Effy, O. (2014). Management Information Systems. Cengage Learning, Inc. United States.

Suvée, D., Vanderperren, W. & Jonckers, V. (2003). JAsCo: an aspect-oriented approach tailored for component-based software development. In *Proceedings of the 2nd international conference on Aspect-oriented software develop-pment* (pp. 21-29). ACM.

Suvée, D., Vanderperren, W., Wagelaar, D., Jonckers, V. (2005). There Are No Aspects. Electronical Notes in

Theoretical Computer Sciences (ENTCS). *Special Issue on Software Composition*, *Vol. 114*, 153-174.

Zelenski, J. (2008). *Thread and Semaphore Examples*. Stanford, CS107, Spring. https://see.stanford.edu/materials/icsppcs107/23-Concurrency-Examples.pdf Accessed 10 July 2017.