Applying Neutrosophic Iadov Technique for assessing an MDD-based approach to support software design

Nemury Silega [1,*], G. F. Castro Aguilar [2,3], I. A. Martillo Alcívar [3], K. M. Faggioni Colombo [3]

[1] Department of System Analysis and Telecommunications, Southern Federal University, Taganrog, Russia
[2] Facultad de Ingeniería, Universidad Católica de Santiago de Guayaquil, Guayaquil, Ecuador
[3] Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil, Guayaquil, Ecuador
Emails: silega@sfedu.ru; gilberto.castroa@ug.edu.ec; inelda.martilloa@ug.edu.ec; katya.faggionim@ug.edu.ec

* Correspondence: silega@sfedu.ru

Abstract

Performing a correct architectural design is essential to satisfy the quality requirements of a software. In this phase, the high-level components that will compose the system, as well as their relationships, are defined. Since the architects must struggle with complex and challenging tasks in this phase, providing them with advanced and helpful tools and methods is suitable. For example, MDD-based approaches are a valuable means to deal with the complexity during the software development process, particularly during the architectural design stage. However, despite the notable benefits of this type of approaches, the architects are often sceptical about adopting new technologies. Hence, before formally adopting new methods or tools, it is suitable to consider the opinion of those using them. In that sense, this paper aims to describe the results of an assessment of an MDA-based approach to support the architectural design. This assessment was carried out applying the Iadov neutrosophic technique. This technique has been extensively applied in a wide variety of domains to analyze the satisfaction level of potential users of different proposals. The results indicate a high satisfaction level of potential users of the assessed approach.

Keywords: Model Driven Architecture (MDA); Iadov neutrosophic technique; architectural design; user satisfaction assessment.

1. Introduction

The complexity of the software development process can be analyzed in terms of essential complexity and arbitrary complexity. The first type of complexity is inherent to the domain (i.e. enterprise management processes, health management processes, etc.). Hence, this complexity is inevitable and cannot be reduced since it does not depend on the software development process (Selic, 2008). Differently to the essential complexity, the arbitrary complexity is related to the internal software development process. This complexity is impacted by the technologies and methods adopted in a software development project as well as by the knowledge of software developers. Therefore, using appropriate technologies and methods is crucial to reduce this complexity. For instance, the complexity of the coding stage will be reduced if a tool that automatically generates a portion of the source code is used. As a result, the productivity of the software developers will increase. On the other hand, using a technique with a high learning curve could make the process less productive. In general, a high essential complexity increases the requirement to lessen the arbitrary complexity. The high-level abstraction components to meet the needs of clients are identified during the architecture design process [1]. Therefore, decisions taken during this stage have a significant impact on the later development stages. Developing systems with a high essential complexity demands adopting technologies and techniques that help to reduce arbitrary complexity. As a result, architects will be more productive, and consequently less time will be required to complete this phase.

The fact that this phase is usually carried out intuitively, depending on the creativity and knowledge of architects affects negatively the success of this phase [1].. These issues lead to frequent redesigns to fix those

Doi: https://doi.org/10.54216/IJNS.190207
Received: February 26, 2022 Accepted: October 08, 2022

errors generated in the design phase but detected in subsequent phases. The redesign increases software development time, affects software quality, and increases project costs. Fixing post-implementation errors cost 100 times more than those detected at the design stage [2-4].
Since that business process models are a primary artifact for the architecture design, the incorrect business process modeling is another factor that affects negatively this phase. Hence, it is desirable to carry out a correct validation of business process models to prevent the propagation of their errors to later phases.
On the other hand, Model Driven Development (MDD) is a paradigm for the software development. MDD aims to reduce the arbitrary complexity of the software development process by increasing the abstraction level, exploiting the automation and encouraging the standardization. Model Driven Architecture (MDA) is the most popular MDD-based method. MDA organizes the software development into three abstraction levels: Computer Independent Models (CIMs), Platform Independent Models (PIMs), and Platform Specific Models (PSMs).
Motivated by the notable benefits of MDA, we developed an MDA-based approach for supporting the architectural design stage. To validate the impact of our approach to improve the architecture design of enterprise management systems, an experiment was conducted. In a previous work the main results of this experiment were analyzed [5]. In spite of this experiment provided quantitative evidence that indicate the advantages of this approach, it is necessary to know the opinion of its potential users. Knowing the opinion of potential users is relevant to assess an approach and a means to obtain feedback before to adopt it in a real environment. Recently the Iadov technique has been applied to assessing different approaches in a wide variety of domains [6-10]. Considering the benefits of this technique, we have used it to assess our approach.
The rest of the paper is structured as follows. Next section presents the methods adopted in this research. Then, the main components of our approach are introduced. In the Result section the results of applying the Iadov technique are analyzed. Finally, conclusions and future work are presented.

2. Materials and Methods

2.1. Iadov Neutrosophic Technique

In this section some of the main concepts of the Iadov technique are presented. This technique was originally created to study a specific social phenomenon by using a survey with three closed questions. The "Iadov logical table" is used to processing the answers and calculating the satisfaction level. An important rule of this technique, it is that the participants are not aware that the three questions are related. In addition to the closed questions, some open questions are included in the survey. These open questions enable the analysis of indetermination (not defined) (I). A process of de-neutrosophication is followed to analyzing these open questions. In this case, I [-1,1].
As it has been described in plenty of works [10-12], the neutrosophic IADOV technique uses single value neutrosophic sets (SVNS) related to linguistic variables enabling to increase the interpretability in the recommendation models and the use of indeterminacy. The definition of a SVNS is as follows:

Let X be a universe of discourse. An SVNS A over X is an object of the form.

$$\mathrm{A} = \{[x, \mathrm{u}_a(x), r_a(x), v_a(x)] : \in \mathrm{X}\}\mathrm{dA} = \{[x, \mathrm{u}_a(x), r_a(x), v_a(x)] : \in \mathrm{X}\}\mathrm{d} \quad (1)$$

Where:

$$u_a(x): X \rightarrow [0,1], r_a(x): X \rightarrow [0,1] \; y \; v_a(x): X \rightarrow [0,1]$$

With

$$0 \leq u_a(X), r_a(X), v_a(X) \leq 3, \forall x \in X$$

In this case, a Single Value Neutrosophic Number (SVNS) will be expressed as $\mathrm{A} = (\mathrm{a}, \mathrm{b}, \mathrm{c})$, where $\mathrm{a}, \mathrm{b}, \mathrm{c} \in [0,1]$ and satisfies $0 \leq \mathrm{a} + \mathrm{b} + \mathrm{c} \leq 3$

To find an SVNS which defines several sets at the same time, aggregation operators are used. For example, the neutrosophic weighted average (WA). WA is defined as follows [10, 13]:

Let the Neutrosophic Weighted Average Operator (WA) be calculated: $\{\mathrm{A1}, \mathrm{A2}, \ldots, \mathrm{An}\} \in \mathrm{SVNS}\,(x)$, $where\ \mathrm{Aj} = (\mathrm{aj}, \mathrm{bj}, \mathrm{cj})(\mathrm{j} = 1, 2, \ldots, \mathrm{n})$,

$$WA\,(A1,2, \ldots, An) = \sum n\ i{=}1[wj\ , Ai] \quad (2)$$

Where:

$$WA\,(w1\ , w2, \ldots, wn) = \sum n\ i = 1[wj\ , Ai]\ is\ the\ vector\ Aj\ (j = 1,2, \ldots, n)\ such\ that\ wn \in [0,1]\ y \sum wj = 1$$

To obtain a single value of this set, a de-neutrosophication process is carried out by applying a scoring function.

Doi: https://doi.org/10.54216/IJNS.190207
Received: February 26, 2022 Accepted: October 08, 2022

Different functions have been used to analyze the results [8-10]. We have used the same function of punctuation applied by Castillo et al [14].

$$S(A) = a - b - c \qquad (3)$$

A linguistic variable will be related to SVNS to calculate the individual satisfaction [10]. In this case, the scales that Table 1 depicts were defined to calculate the scores by using (3).

Table 1: Individual scale satisfaction with SVN values.

Linguistic expression	SVN	Scoring
Clearly pleased	(1, 0, 0)	1
More pleased than unpleased	(1, 0.25, 0.25)	0.5
Not defined	I	0
More unpleased than pleased	(0.25, 0.25, 1)	-0.5
Clearly unpleased	(0,0,1)	-1
Contradictory	(1,0,1)	0

A specific process is carried if the result indicates indeterminacy (not defined) (I).

$$\lambda([a1, a2]) = \frac{a1 + a2}{2} \qquad (4)$$

The Global Satisfaction Index (GSI) is calculated by using the operator aggregation WA and considering the scoring values. It was considered that the weight of all participants is the same, thus $\text{wi} = \frac{1}{n}$. In section 3, the results of applying these concepts are presented.

2.2. An approach based on MDA and ontologies to support the architecture design

In this section, we briefly introduce our proposal to support the architecture design. In previous works we have provided a detailed description of this approach [15, 16]. Since this approach is based on MDD and ontologies, we describe briefly the main concepts related to these two technologies below.

MDA is a methodology promoted by the Object Management Group (OMG). MDA encourages models as the main artifacts for understanding, designing, developing, implementing, and maintaining systems (OMG, 2003). MDA aims to develop systems with high flexibility in implementation, integration, maintaining, and testing. Portability, reusability and interoperability are the three main MDA principles. Three types of models are defined in MDA: Computer Independent Models (CIMs), Independent Platform Models (PIMs) and Platform Specific Models (PSMs).

A CIM is a view of the business, regardless of system specifications. Bridging the gap between business experts and software developers is the main goal of this type of model. A PIM is a system view that does not include platform specifications. This type of model contributes to separating the logical design concerns from the platform-specific concerns. Whereas, a PSM is a view of the system that includes the details of a specific platform.

Model transformations is the other key component of MDA. A model transformation consists of generating a new model from others of the same system. Achieving the automatic model transformation is the aim of the scientific community to increase the productivity in the software development process and enhance the software quality. Several studies have demonstrated the positive impact of MDA-based approaches to improve the productivity of the software development process and the quality of the resulting software [17-19].

On the other hand, an ontology is a formal and explicit description of a discourse domain concepts (classes), the properties of each concept, attributes, and restrictions [20]. The ontologies have been extensively exploited to represent and analyze knowledge in several domains [16, 21-27]. The application of ontologies in the MDA context may enable the models consistency checking and validation. We found several MDA-based works which include models to represent and validate processes through formal models [18, 28-31].

To create our ontologies we used OWL [32]. OWL includes a set of operators to represent different types of relations, such as intersection, union, and negation. Since OWL is based on description logic, it is possible to employ reasoners to automatically check the models consistency. In addition, the tool Protégé supports the management of ontologies in OWL. Whilst the methodology of Noy and McGuinness [20] was adopted to develop our ontology. This methodology has been extensively adopted to guide the development of ontologies [33].

Once introduced briefly the two main underpin technologies of our approach, the main components of the approach are presented below. Fig. 1 depicts a general view of the proposal. In Fig. 1, the rectangles represent the types of models, and the arrows represents the transformations. Three CIMs and two PIMS are included in

Doi: https://doi.org/10.54216/IJNS.190207
Received: February 26, 2022 Accepted: October 08, 2022

the proposal. In addition, four models transformations are included as well. Previous works have described the aim of each model.

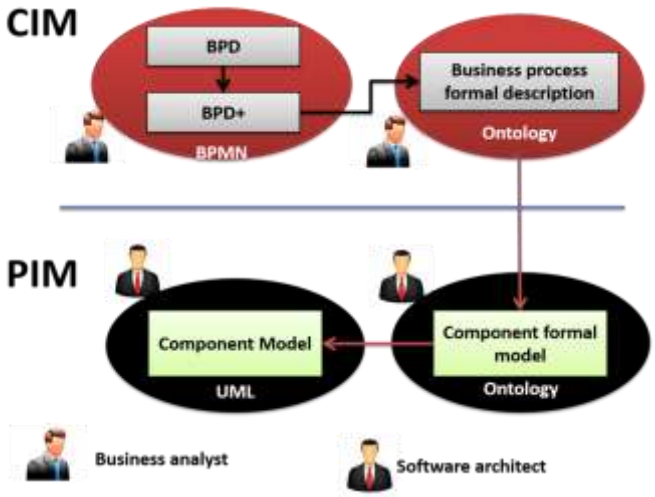

Figure 1: General graphical representation of the proposal: Models and transformations.

2.2.1. Computer Independent Models

BPD: This model allows creating business processes descriptions so that it can be easily understood by both business specialists and software developers. However, some particular domain concepts cannot be represented in this model. Therefore, the following CIM was included in this level.
BPD+: It is an intermediate model to extend the BPD with the particular-domain information that the BPD could not represent.
Ontological model: This ontology includes classes, properties, and restrictions to represent a business process. Some classes are related to BPMN concepts, such as Activity, Processs, *FlowElement*, *Event*, *Gateway*, and Object. The ontology also includes object properties to relate individuals.

2.2.2. Platform Independent Models

At this level, a high-level architecture view is provided. This view includes the components and their relationships. This representation aims to show how the system components are coordinated to meet the business needs. The most significant concepts are *Component*, *Service*, and *Functionality*.
Ontological component model: The ontology includes classes and properties to represent the system architecture. Some relevant classes are Component, Service, and Functionality. A set of object properties were defined to relate the model concepts. For example, it was stated that a *Service* is provided by a *Component* and a *Component* implements some *Functionalities*. Furthermore, some classes to classify the components were included, for example, *BusinessComponent*, *DomainComponent*, and *TechnologyComponent*.
UML component model: Representing the component model by means of an ontology provides the benefits that were mentioned above. However, this type of model is not easy to understand for those who are non-experts with these technologies. Hence, it is suitable to include a model that is easy to understand by software developers. In that sense, the UML component model has been adopted. This is a well-known model and it is interpreted by a wide variety of modeling tools. As Fig. 1 shows, the PIM level is composed of an ontological model and an UML component model.

2.4.2. Tool to support for model transformation

The tool that we developed to support the approach is described below. The main functionalities of this tool are depicted in Fig. 1. This tool was created as a Protégé plugin to automate the three transformations that the approach includes. These transformations are:

1. **BPD -> BPD+**: In this step, the BPD is extended with specific domain information.

Doi: https://doi.org/10.54216/IJNS.190207
Received: February 26, 2022 Accepted: October 08, 2022

2. **BPD+ -> Ontological business process model**: With this transformation, an ontological representation of the processes is generated from the BPD. A set of transformation rules were defined to generate this model.
3. **Ontological business process model -> Ontological component model**: This is a CIM to PIM transformation. In this case, a component model is generated from the business process model. Both models are represented by means of ontologies.
4. **Ontological component model -> UML component model**: A UML component model is generated from the ontological component model in this transformation.

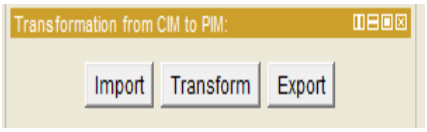

Figure 2: Main functionalities of the plugin.

3. Results

This section describes the results of applying the Iadov technique for assessing our approach. In this study participated 12 software specialists which have at least five years of experience working in different software development projects. We carried out a survey by using three questions. Table 2 shows the Iadov logical table where two closed questions are included as well as an open question.

Table 2: Iadov logical table.

	Do you consider that, to carry out the architectonic design, it is not necessary to use an approach that takes into account the business processes and provides automated support to generate and validate the system architecture models?								
	No			I don't know			Yes		
Are you satisfied with the conception of this approach to support the architectonic design?	Would you use this approach to carry out the architectonic design of a system?								
	Yes	I don't know	No	Yes	I don't know	No	Yes	I don't know	No
Very satisfied	1	2	6	2	2	6	6	6	6
Partially satisfied	2	2	3	2	3	3	6	3	6
I don´t care	3	3	3	3	3	3	3	3	3
More unsatisfied than satisfied	6	3	6	3	4	4	3	4	4
Not at all satisfied	6	6	6	6	4	4	6	4	5
I don't know what to say	2	3	6	3	3	3	6	3	4

Table 3: Summarized results.

Expression	Total	Percent
Very satisfied	9	75 %
Partially satisfied	2	17 %
Not defined	1	8 %
More dissatisfied than satisfied	0	0 %
Clearly dissatisfied	0	0 %
Contradictory	0	0 %

Doi: https://doi.org/10.54216/IJNS.190207
Received: February 26, 2022 Accepted: October 08, 2022

Based on the results presented in table 3, the Group Satisfaction Index (GSI) was calculated. In our case, we obtained GSI =0.83. According to the scale depicted in Fig. 3, this GSI indicates that the participants are satisfied with the approach.

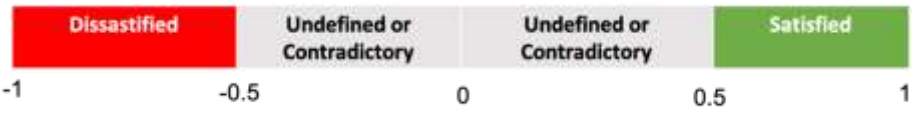

Figure 3: Scale to determine the level of satisfaction according to the Group Satisfaction Index (GSI)

In addition to the closed question, the Iadov technique includes open questions. In this particular case, we asked the following questions:

- What are the negative elements of this approach?
- What do you recommend to improve this approach?

With the answer to these questions, we obtained interesting recommendations to improve our approach. Since the participants are expert in the architectonic design, their recommendations are a valuable insight to enhance the specifications of the proposal.

5. Conclusions

In this paper we have presented the results of assessing an approach to support the architectonic design of a software. The assessment was carried out by applying the Iadov neutrosophic technique. This technique enabled knowing the opinion of potential users regarding the approach. The value of the Group Satisfaction Index indicated that the participants are satisfied with the approach. In addition, the application of this technique contributes to have feedback from potential users. This is an important insight to the success implementation of a technology because the opinion of the users is crucial. In addition, the open questions that the technique includes helped to identify improvements of the approach. Since that some participants have wide experience in the software design, their recommendations are a valuable help to enhance our approach. The results documented in this paper complement the empirical evaluation that was carried out previously. Both evaluations provide evidences that prove the potentialities of this approach.

References

[1] Al-Jamimi, H. and M. Ahmed, *Transition from Analysis to Software Design: A Review and New Perspective.* International Journal of Soft Computing and Software Engineering [JSCSE], 2013. **3**(3): p. 169-176.

[2] Sánchez, L., et al., *Quality indicators for business process models from a gateway complexity perspective.* Information and Software Technology, 2012. **54**(11): p. 1159-1174.

[3] Boehm, B.W., *Software Engineering Economics*. 1981: Prentice-Hall, Englewood Cliffs.

[4] Moreno, I., et al., *A systematic literature review of studies on business process modeling quality.* Information and Software Technology, 2014.

[5] Silega, N.M., Inelda A Castro, Gilberto F. Faggioni , Katya M., *Assessing the impact of an MDA-based approach to support the architecture design*. Revista Cubana de Ciencias Informáticas, 2022. **16**(2).

[6] Hernández, N.B., et al., *Validation of the pedagogical strategy for the formation of the competence entrepreneurship in high education through the use of neutrosophic logic and Iadov technique*. 2018: Infinite Study.

[7] Calle, W.A.C., A.S.G. Betancourt, and N.J. Enríquez, *Validation of the proof reversal on the inexistence of untimely dismissal by using neutrosophic IADOV technique.* Neutrosophic Sets and Systems, 2019. **33**(1): p. 33-36.

[8] Velázquez, A.I.U., et al., *Neutrosophic Iadov tecnique for assessing the proposal of standardization of the beef cutting for roasting in Patate canton, Ecuador.* Neutrosophic Sets and Systems, 2020. **34**: p. 86-92.

[9] Zatán, L.G.G., et al., *Neutrosophic Iadov for measuring of user satisfaction in a virtual learning environment at UNIANDES Puyo.* Neutrosophic Sets and Systems, 2020. **34**: p. 117-125.

[10] Falcón, V.V., et al., *Managing Contradictions in Software Engineering Investigations using the Neutrosophic IADOV Method.* Neutrosophic Sets and Systems, 2021. **44**: p. 100-107.

[11] Morales, L.G., R.W.P. Ventura, and A.H. González, *Iadov Neutrosofico para medir la satisfacción de los docentes con la aplicación del Solver de Excel en la programación lineal.* Revista Asociación Latinoamericana de Ciencias Neutrosóficas. ISSN 2574-1101, 2019. **5**(1): p. 19-28.

[12] Abdel-Basset, M., et al., *A novel model for evaluation Hospital medical care systems based on plithogenic sets.* Artificial intelligence in medicine, 2019. **100**: p. 101710.

Doi: https://doi.org/10.54216/IJNS.190207
Received: February 26, 2022 Accepted: October 08, 2022

[13] Andino Herrera, A., et al., *Use of the neutrosophic IADOV technique to diagnose the real state of citizen participation and social control, exercised by young people in Ecuador.* Neutrosophic Sets & Systems, 2019. **26**.
[14] Zuñiga, V.J., et al., *Validation of a Model for Knowledge Management in the Cocoa Producing Peasant Organizations of Vinces Using Neutrosophic Iadov Technique.* Neutrosophic Sets and Systems, 2019. **30**(1): p. 20.
[15] Silega, N., M. Noguera, and D. Macias, *Ontology-based Transformation from CIM to PIM.* IEEE Latin America Transactions, 2016. **14**(9): p. 4156-4165.
[16] Silega, N. and M. Noguera, *Applying an MDA-based approach for enhancing the validation of business process models.* Procedia Computer Science, 2021. **184**: p. 761-766.
[17] Kharmoum, N., et al., *A method of model transformation in MDA approach from E3value model to BPMN2 diagrams in CIM level.* IAENG International Journal of Computer Science, 2019. **46**(4): p. 1-17.
[18] Melouk, M., Y. Rhazali, and H. Youssef, *An Approach for Transforming CIM to PIM up To PSM in MDA.* Procedia Computer Science, 2020. **170**: p. 869-874.
[19] Martínez, Y., C. Cachero, and S. Meliá, *MDD vs. traditional software development: A practitioner's subjective perspective.* Information and Software Technology, 2013. **55**(2): p. 189-200.
[20] Noy , N.F. and D.L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*. 2001, Stanford Medical Informatics: Stanford.
[21] Bouzidi, R., et al., *OntoGamif: A modular ontology for integrated gamification.* Appl. Ontology, 2019. **14**(3): p. 215-249.
[22] Nicola, A.D., M. Melchiori, and M.L. Villani, *Creative design of emergency management scenarios driven by semantics: An application to smart cities.* Inf. Syst., 2019. **81**: p. 21-48.
[23] Xinga, X., et al., *Ontology for safety risk identification in metro construction.* Computers in Industry, 2019. **109**: p. 14–30.
[24] Yang, L., K. Cormicana, and M. Yub, *Ontology-based systems engineering: A state-of-the-art review.* Computers in Industry, 2019. **111**: p. 148–171.
[25] Freitas, S., E. Canedo, and D. Jesus, *Calculating similarity of curriculum lattes.* IEEE Latin America Transactions, 2018. **16**(6): p. 1758-1764.
[26] Keet, C.M. and R. Grütter, *Toward a systematic conflict resolution framework for ontologies.* Journal of Biomedical Semantics, 2021. **12**(1).
[27] Bencharqui, H., S. Haidrar, and A. Anwar, *Ontology-based Requirements Specification Process.* E3S Web Conf., 2022. **351**.
[28] Kharmoum, N., et al., *A Method of Model Transformation in MDA Approach from E3value Model to BPMN2 Diagrams in CIM Level.* IAENG International Journal of Computer Science, 2019. **46**(4).
[29] Li, Z., X. Zhou, and Z. Ye, *A Formalization Model Transformation Approach on Workflow Automatic Execution from CIM Level to PIM Level.* International Journal of Software Engineering and Knowledge Engineering, 2019. **29**(09): p. 1179-1217.
[30] Laaz, N., et al. *Integrating Domain Ontologies in an MDA-Based Development Process of e-Health Management Systems at the CIM Level*. in *International Conference on Advanced Intelligent Systems for Sustainable Development*. 2019. Springer.
[31] Laaz, N., N. Kharmoum, and S. Mbarki, *Combining Domain Ontologies and BPMN Models at the CIM Level to generate IFML Models.* Procedia Computer Science, 2020. **170**: p. 851-856.
[32] Xing, J. and T. Ah-Hwee, *CRCTOL: A semantic-based domain ontology learning system.* Journal of the American Society for Information Science & Technology, 2010. **61**(1): p. 150-168.
[33] Sattar, A., et al., *Comparative analysis of methodologies for domain ontology development: A systematic review.* Int. J. Adv. Comput. Sci. Appl., 2020. **11**(5): p. 99-108.

Doi: https://doi.org/10.54216/IJNS.190207
Received: February 26, 2022 Accepted: October 08, 2022