Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 644 – 652

Procedia
Computer Science

18th International Conference on Knowledge-Based and Intelligent
Information & Engineering Systems - KES2014

Implementing Virtual Engineering Objects (VEO) with the Set of Experience Knowledge Structure (SOEKS)

Syed Imran Shafiq[a]*, Cesar Sanin[a], Edward Szczerbicki[b] & Carlos Toro[c]

[a]*The University of Newcastle, University Drive, Callaghan, 2308, NSW, Australia.*
[b]*Gdansk University of Technology, Gdansk, Poland.*
[c]*Vicomtech-IK4, San Sebastian, Spain.*

Abstract

This paper illustrates the idea of Virtual Engineering Object (VEO) powered by Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA). A VEO is the knowledge representation of an engineering object, having embodiment of all its associated knowledge and experience within it. Moreover, VEO is a specialization of Cyber-Physical System (CPS) in terms of that its extension in knowledge gathering and reuse, whereas CPS is only aimed towards data and information management. The SOEKS/DDNA is a flexible and standard knowledge representation structure to acquire and store experiential knowledge. The article also presents a case study to demonstrate implementation of VEO in the manufacturing scenario. The decision making in industrial design and manufacturing will benefit from this approach, as it includes capturing, storage and reuse of experience and knowledge of an engineering artefact.

Peer-review under responsibility of KES International.

Keywords: Desional DNA; Set of Knowledge Experience Structure (SOEKS); Virtual Engineering Object (VEO).

1. Virtual Engineering Object (VEO) and Cyber-Physical Systems (CPS) reconciliation

There exists a great interest by the European commission under the Horizons 2020 programme, in Cyber-Physical Systems (CPS). In specific under the Factories of the Future PPP (Public Private Partnership)

---

* Corresponding author. Tel.: +61 405408834.
E-mail address: SyedImran.Shafiq@uon.edu.au.

Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.146

initiative, the EU commission will fund projects related to such technologies by an amount of millions of euros starting in 2014 and aiming towards a full implementation in 2020.

The aforementioned fact, exemplifies the point that CPS is becoming a must have technology as it is in fact needed by the industry.

CPS are integrations of computation with physical processes [1, 2]. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. In the physical world, the passage of time is inexorable and concurrency is intrinsic. Neither of these properties is present in today's computing and networking abstractions [2].

CPS-based processes optimisation aiming towards adaptive and smart manufacturing systems bringing together novel concepts for CPS, progress in advanced control and new modelling and simulation technologies will be arguably one of hot topics in research and that is why, in this section, the relation between the VEO concept and CPS concept in terms of a terminological reconciliation is provided .

Scalable CPS architectures for adaptive and smart manufacturing systems which dynamically enable the continuous design, configuration, monitoring and maintenance of operational capability, quality, and efficiency are in fact a requirement for the industry [3]. According the European commission, the Self-learning closing the feedback loop between production and design should be included in future factories for advantaging and maximizing energy expenditure and minimizing the waste as a direct relation to the enhancement in control and immediate information processing that a CPS sensor based architecture will provide.

In our definition [4], A VEO is a knowledge representation of an engineering artefact comprising experience models, domain and functionally along a physical attachment to the virtual object in its conceptualization.

The relation between the two concepts is evident in the sense that a VEO is a specialized kind of CPS system aiming towards the gathering of experiential knowledge and re-use whereas a bare bone CPS is aimed towards data and information gathering and management. A VEO is then a generalization of a CPS which by its conceptualization will fall into a homogenous line of thinking aligned to our previous presented work [5].

In this article, we present an approach that integrates VEO with Decisional DNA (DDNA) to capture and reuse manufacturing experiences. In addition, we demonstrate this approach as a test case study in order to test the usability and suitability of VEO in supporting decision making and knowledge reuse.

DDNA as a domain-independent, flexible and standard knowledge repository can not only capture and store experiential knowledge in an explicit way but can also be applied to various domains. DDNA supports decision making and standard knowledge sharing among these systems [6-8.]

This paper is organized as follows: section two describes the Set of Experience Knowledge structure and Decisional DNA; section three presents the structure of a VEO; section four demonstrates implementation of the Decisional DNA applied to VEO and the experiment results of our approach. Finally, section five discusses concluding remarks and the scope of future work.

2. The Set of Experience Knowledge Structure (SOEKS) and Decisional DNA

The Set of Experience Knowledge Structure (SOEKS or shortly SOE) is a domain-independent, flexible and standard knowledge representation structure. It has been developed to acquire and store formal decision events in an explicit way. It is a model based upon available and existing knowledge, which must adapt to the decision event it is built from (i.e. it is a dynamic structure that depends on the information provided by a formal decision event); besides, it can be represented in XML or OWL as an ontology in order to make it transportable and shareable[9, 10].

SOEKS is composed of variables, functions, constraints and rules as shown in figure 1. Variables normally implicate representing knowledge using an attribute-value language (i.e. by a vector of variables and values), and they are the centre root of the structure and the starting point for the SOEKS. Functions represent relationships between a set of input variables and a dependent variable; moreover, functions can be applied for reasoning optimal states. Constraints are another way of associations among the variables. They are restrictions of the feasible solutions, limitations of possibilities in a decision event, and factors that restrict the performance of a system. Finally, rules are relationships between a consequence and a condition linked by the statements IF-THEN-ELSE. They are conditional relationships that control the universe of variables [11].

Additionally, SOEKS is designed similarly to DNA at some important features. First, the combination of the four components of the SOEKS gives uniqueness, just as the combination of four nucleotides of DNA does. Secondly, the elements of SOEKS are connected with each other in order to imitate a gene, and each SOEKS can be classified, and acts like a gene in DNA [11]. As the gene produces phenotypes, the SOEKS brings values of decisions according to the combined elements. Then a decisional chromosome storing decisional "strategies" for a category is formed by a group of SOEKS of the same category. Finally, a diverse group of SOEKS chromosomes comprise what is called the Decisional DNA [6].

In short, as a domain-independent, flexible and standard knowledge representation structure, SOEKS and Decisional DNA provide an ideal approach which can not only be very easily applied to various domains, but also enable standard knowledge communication and sharing among these systems [8].

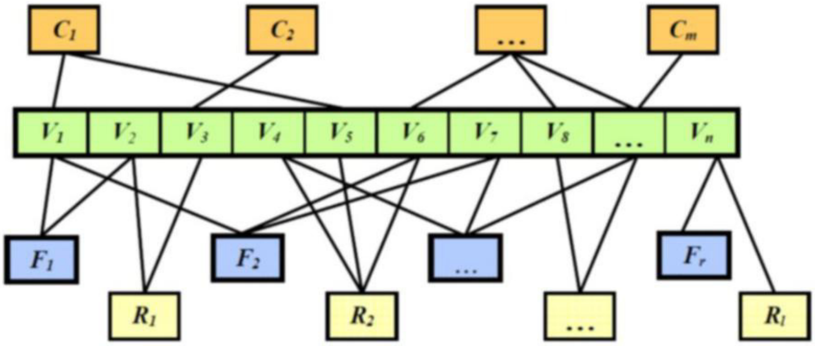

Fig.1. The structure of set of experience [12,9]

3. Virtual Engineering Object (VEO)

A VEO is a living representation of an artefact which can behave like an expert of that area and can help the practitioners in effective decision making based on the past experience [8]. The concept of Virtual Engineering Object (VEO) is powered by SOEKS and DDNA; it is designed to have all the knowledge of the engineering artefact along with the associated experience embedded in it.

VEO provides a standard knowledge representation format and eventually form various networks of interconnected VEOs based on their past manufacturing experience. These networks of VEOs form a part of a bigger CPS umbrella as discussed in section 1 and it is pretty similar to the concept of Cyber Physical Production Systems (CPPS), with the difference that such systems are aimed towards data and information management, whereas VEO networks are focused on Knowledge and Experience gathering and re-use.

3.1 Structure of a VEO

In this section we propose and discuss the structure architecture of a VEO. A VEO is able to encapsulate knowledge and experience of every important feature related with a given engineering object. This can be achieved by gathering information from six different aspects of an engineering object namely Characteristics, Functionality, Requirements, Connections, Present State and Experience as illustrated in Fig. 2.

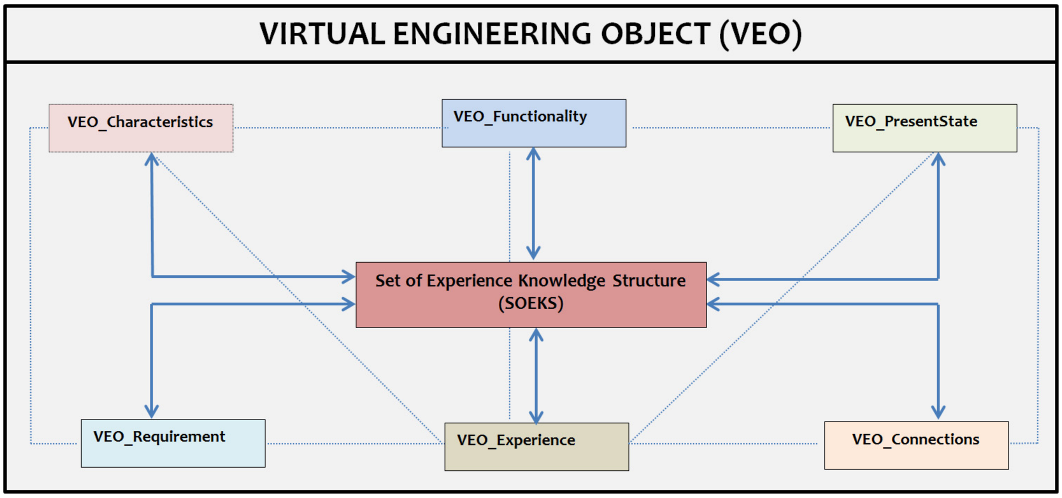

Fig. 2. Proposed VEO Structure [8, 13]

The main features of a VEO (shown in Fig. 2) are as follows:

- *Characteristics* describe the set of physical features and expected benefits offered by the artefact represented by the VEO. Not only the information like its geometry dimensions, appearance, weight etc. are captured in this module but also the possible advantages like ‘versatility’ and the ‘ease of operation’ can also be achieved from this. Knowledge stored in Characteristics assist in better decisions making like, which VEO is best suited for a given physical condition and also when more than one VEO of similar kind are available it helps to decide which is the best in the given situation.
- *Functionality* describes the basic working of the VEO and principle on which it accomplish its operation. Operational Knowledge related with an object like time consumed, its working boundary limits and the outcome of the process that is performed are stored in Functionality. This module of the VEO assists in storing, selecting and reusing the practical details.
- *Requirements* describe the set of necessities of the VEO for its precise working. Information like what and how much power source, how much space and the extent of user expertise necessary for operating a VEO can be stored.
- *Connections* describe how the VEO is related with other VEOs. Many engineering object works in conjunction with other object, these connecting VEO’s may be a “part” or may be a “need” of each other. This module of VEO structure is essential for the scaling up and establishing the interconnection of VEO’s in manufacturing scenario.
- The *present state* of the VEO highlights parameters of the VEO at the current moment. It is like whether the VEO is ready for a particular operation? If yes, then it also gives an idea about the background of the VEO like its ‘reliability’ and ‘precision’ up till now. If it is busy it can predict the expected time when the VEO will be free for the next operation.
- The *Experience* of the VEO deals with the knowledge and information which is dynamic in nature and keeps on changing with each new decision, operation or event. In other words every formal decision

related with the VEO is stored in the experience. This element of the VEO keeps on updating with every activity that is done on the VEO.

4. Implementation of a VEO

For the purpose of implementation of VEO, we integrated it with the Decisional DNA. As discussed in section 2 SOEKS comprises of Variables, Functions, Constraints and Rules. Moreover in section 3 we also discussed that a VEO structure include elements like Characteristics, Functionality, Requirements, Connections, Present State and Experience. SOEKS files for each element of the VEO in the system are created individually. The goal behind this was to provide a more scalable setting, similar to the one that would be found in describing a diverse range of engineering objects. Weights are assigned to the attributes of the variables of an artefact, and then the six sets of SOEKS are generated. These individual SOEKS are combined under an umbrella (VEO), representing experience and knowledge.

4.1 Test Case Study

The purpose of the present test case study is to apply our proposed architecture of VEO to gather, store and reuse information related to the Drilling Machines in a manufacturing setup. Table 1 illustrates the detailed VEO structure for drilling machines. Effort is made to capture and store all the relevant information of the VEO adhering to the format of the SOEKS.

Table 1. Structure of a VEO

	VARIABLES	FUNCTIONS	CONSTRAINTS	RULES
CHARACTERISTICS	veoCODE veoNAME veoLENGTH VeoWIDTH veoHEIGHT veoWEIGHT veoCOLOR veoMaxWPWeight veoCAPACITY veoMATERIAL veoOPERATOR veoMANUFACTURER veoPURCHASE veoServiceDate	veoAREA=veoLength*veoWidth veoVOL=veoLength*veoWidth*veoHeight	veoLength>0 veoWidth>0 veoHeight>0 veoWeight>0	IF veoOperator=Skilled THEN 0<veoEaseofOperation<=3 IF veoOperator=Semi-skilled THEN 3<veoEaseOfOperation<=6 IF veoOperator=Un-skilled THEN 7<veoEaseOfOperation<=10
			veoWPLength<veoMaxWPLength veoWPWidth<veoMaxWPWidth veoWPHeight<veoMAXWPHeight	IF veoNoOfOp<5 THEN 0<Ease of Adaptability<=5 ELSE <Ease of Adaptability <=10
				IF Date-veoServiceDate> 6 months THEN “Service Due” ELSE “Next service due on … days”
FUNCTIONALITY	veoCODE veoNAME VeoFUNCTION veoPRINCIPLE veoSpindleSpeed veoCuttingSpeed veoFEED veoDepthOfCut veoMaxHoleDia veoMaxHoleDepth veoCoolantUsed	veoMachining Time = (60*veoSpindleSpeed)/ (π*veoDepthOfHole)	50≤ veoSpindleSpeed≤ 3000 10≤ veoCuttingSpeed≤50 .2 ≤ veoFeed≤32 veoDiaOfHole<veoMaxHoleDia veoDepthOfHole<veoMaxHoleDepth	IF 1.6 < veoSurfaceFinish <= 3.6, THEN Smooth IF 3.6 < veoSurfaceFinish <= 4.6, THEN Moderate IF 4.6 < veoSurfaceFinish <= 6.3, THEN Rough
				IF veoDiaofHole-0.05 ≤ veoPrecision ≤ DiaofHole +0.05 ,THEN “Good Precision” ELSE “Not Good Precision”
				IF veoMachiningTime>60, THEN “Use Coolant”

REQUIREMENTS	veoCODE veoNAME veoPowerSource veoSpaceRequired veoOperatorRequired		veoSpaceRequired ≥ veoVolume 220v ≤ veoPowerSource ≤ 250v	
CONNECTIONS	veoCODE veoNAME veoNEED/PART veoVEO[1]			
PRESENT STATE	veoCODE veoNAME veoTotalOpTime veoSTATUS veoReliability veoProductivity veoPrecision veoBreakdownDetails	veoTotalOpTime= $\sum_{veoCODE=1}^{n} (veoMachiningTime)$ veoReliabilty =f(precision,breakdown)		IF veoSTATUS =Busy, THEN wait for veoMachiningTime ELSE "Ready"
EXPERIENCE	veoCODE veoNAME veoOperationNo veoWorkpieceMaterial veoDiameterOfHole veoDepthOfHole veoWorkpieceDim veoMachiningTime veoDrilledPrecision veoFEED veoCutttingSpeed veoDepthOFCut veoOperator veoSurafceFinish veoToolUsed veoWorkHoldingDevice veoWPLength veoWPWidth veoWPHeight			

In the Characteristics section of drilling machine VEO physical parameters like area, volume, maximum capacity, manufacturer details, service details are stored. Furthermore rules are laid to extract knowledge about the VEO like 'ease of operation' and 'adaptability'. In Functionality, variables related with the functioning of a drilling machine like cutting speed, feed, depth of cut, drilling diameter, drilling depth etc. are defined along with their operational limits. In addition to this knowledge, the outcome of drilling operation like quality of surface finish and precision can also be represented in the form of rules. How much Space is required? What and how much power source is required? What kind of expertise of the operator is necessary? All these information for each and every operation can be stored in the Requirements section of the VEO.

While working on drilling machine the drilling tool e.g. Twist Drill and the work holding device e.g. vice are considered as separate VEO's. And Information of these and their relation with main drilling machine VEO are stored in the connections. In the present state not only whether the VEO is free or idle is determined but also knowledge about VEO like its overall reliability and precision till date can be extracted from this segment.

And lastly in the Experience all the dynamic information related with each operation performed and the formal decisions taken on the drilling machine are stored.

This VEO structure is implemented using JAVA programing language, reason being the Decisional DNA constructed in JAVA[7] has been successfully applied in various other domains. Every Variable (see table 1) is stored as the SOEKS variable, to provide an illustration of such a structure, what follows is the VEO Name stored as a SOEKS variable:

```
<variable>
   <var_name>VEO_Name</var_name>
   <var_type>CATEGORICAL</var_type>
   <var_cvalue>DM1 </var_cvalue>
   <var_evalue>DM1 </var_evalue>
   <unit></unit>
   <internal>false</internal>
   <weight>0.0</weight>
   <l_range >0.0</l_range>
   <u_range >0.0</u_range>
   <categories>
         <category></category>
   </categories>
   <priority>0.0</priority>
</variable>
```

4.2 Decisional DNA based VEO Experimental Results

Figure 3 shows six JAVA classes for a VEO each having SOEKS Variables, SOEKS Functions, SOEKS Constraints and SOEKS Rules. This schema is the JAVA version of the architecture depicted in figure2. SOEKs for each class are stored individually. In a separate class these SOEKS are combined to form knowledge and experience repository of an entire VEO. From this knowledge base manufacturing information related with the VEO can be extracted for future decision making.

The formal decisions that are taken with regard to the engineering object are stored adhering to the structure of SOEKS and VEO. Thus, we are able to capture and store information of every operation that is performed and then update the knowledge base of the VEO. The gathered information is effectively and efficiently converted into Decisional DNA structure. Furthermore we are able to query the VEO and based on the experience it can predict and suggest options available according to our need.

5. Conclusions and future work

We presented an approach that allows a VEO to capture and reuse its own experiences by applying Decisional DNA. The SOEKS and DDNA based VEO seems to be a suitable and comprehensive tool for knowledge discovery. We described the architecture of our approach and implemented and tested our concepts in the form of a case study. The illustrative empirical case study presented in this article also showed that this structure can be used as an effective and precise prediction tool. The Decisional DNA, as a novel knowledge representation structure, not only can be applied to VEO but indeed can improve the manufacturing performance when it is using experience. Finally, the concept of VEO is a specialized form of CPS that enables users to make their knowledge shareable, transportable, and easily understandable.

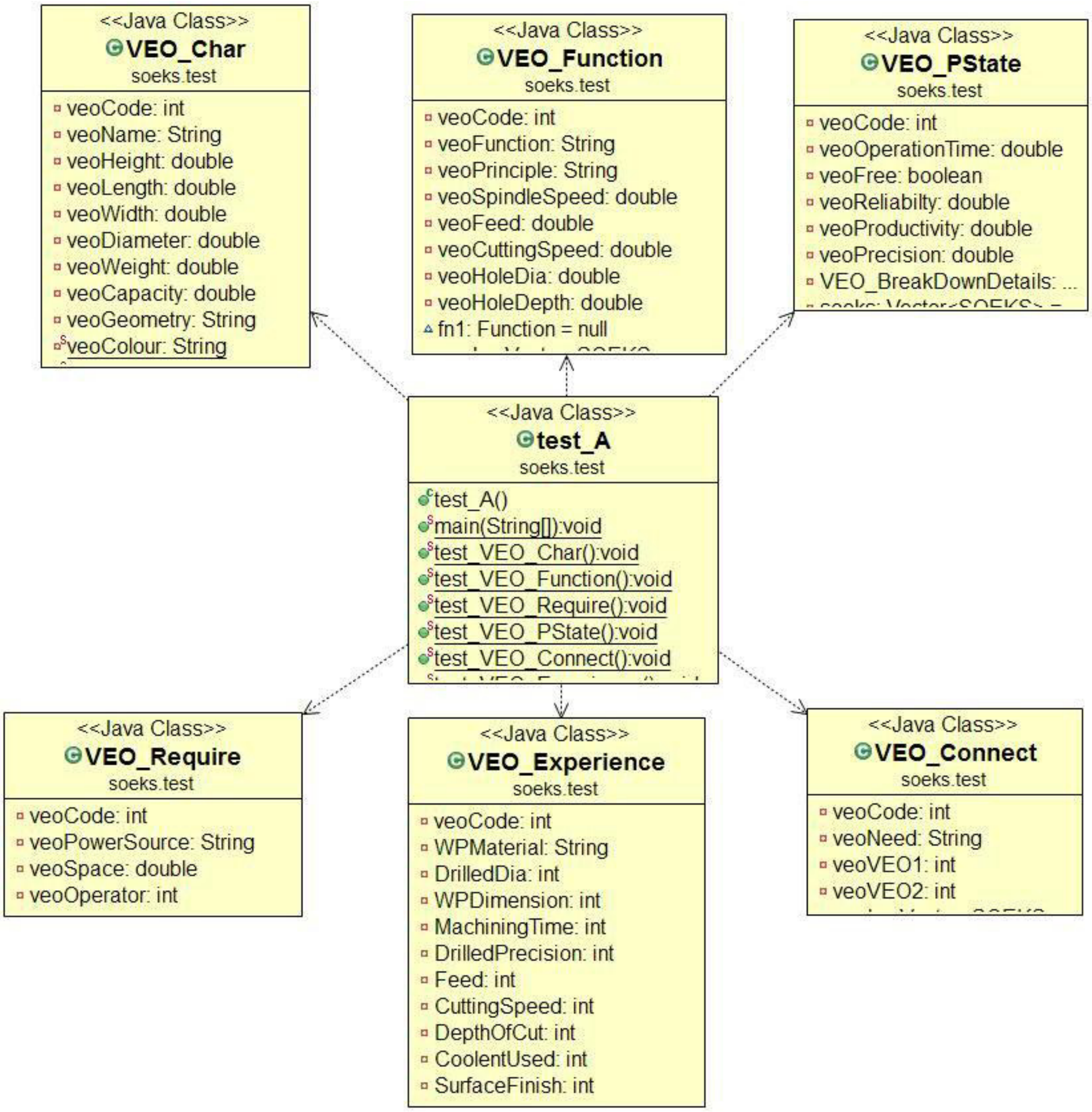

Fig.3. JAVA class diagram

To continue with this idea, further research and refinements are required, and our efforts are currently directed towards:

- Formulation of general VEO structure.
- Refinement of variables that can represent VEO in a more general way.
- Further development of the VEO rule base database.
- Further development of user management based on gathered experience.

References

1. E. A. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?," presented at the In Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin, TX, 2006.
2. E. Lee, "Cyber Physical Systems: Design Challenges," University of California, Berkeley2008.
3. A. Garcia-Crespo, B. Ruiz-Mezcua, J. L. Lopez-Cuadrado, and J. M. Gomez-Berbis, "Conceptual model for semantic representation of industrial manufacturing processes," *Computers in Industry,* vol. 61, pp. 595-612, 9// 2010.
4. S. I. Shafiq, C. Sanin, E. Szczerbicki, and C. Toro, "Using Decisional DNA To Enhance Industrial And Manufacturing Design: Conceptual Approach," presented at the Information Systems Architecture and Technology, Szklarska Poreba, Poland, 2013.
5. C. Sanin, C. Toro, Z. Haoxi, E. Sanchez, E. Szczerbicki, E. Carrasco*, et al.*, "Decisional DNA: A multi-technology shareable knowledge structure for decisional experience," *Neurocomputing,* vol. 88, pp. 42-53, 7/1/ 2012.
6. C. Sanín and E. Szczerbicki, "Experience-based Knowledge Representation: SOEKS," *Cybernetics and Systems,* vol. 40, pp. 99-122, 2009/02/12 2009.
7. C. Sanin and E. Szczerbicki, "Towards the Construction Of Decisional DNA: A Set Of Experience Knowledge Structure JAVA Class within an Ontology System," *Cybernetics and Systems,* vol. 38, pp. 859-878, 2007/10/31 2007.
8. S. I. Shafiq, C. Sanin, and E. Szczerbicki, "Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA): Past, Present and Future," *Cybernetics and Systems,* vol. 45, pp. 200-215, 2014.
9. C. Sanin and E. Szczerbicki, "Extenteding Set Of Experience Knowledge Structure into a Transportable Language extensible Markup Language," *Cybernetics and Systems,* vol. 37, pp. 97-117, 2006/03/01 2006.
10. C. Sanín, L. Mancilla-Amaya, Z. Haoxi, and E. Szczerbicki, "Decisional DNA: The Concept and its Implementation Platforms," *Cybernetics and Systems,* vol. 43, pp. 67-80, 2012/02/01 2012.
11. C. Sanín, L. Mancilla-Amaya, E. Szczerbicki, and P. CayfordHowell, "Application of a Multi-domain Knowledge Structure: The Decisional DNA," in *Intelligent Systems for Knowledge Management*. vol. 252, N. Nguyen and E. Szczerbicki, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 65-86.
12. C. Sanin and E. Szczerbicki, "Set of Experience: A Knowledge Structure for Formal Decision Events," *Foundations of Control and Management Sciences,* vol. 3, pp. 95-113, 2005.
13. S. I. Shafiq, C. Sanin, E. Szczerbicki, and C. Toro, "Decisional DNA based framework for representing Virtual Engineering Objects," in *Intelligent Information and Database Systems, 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, April 7-9, 2014, Proceedings, Part I* vol. 8397, B. A. N.-T. Nguyen, B. Trawinski, K. Somboonviwat Ed., ed: Springer, Lecture Notes in Artificial Intelligence, 2014.