Virtual Engineering Object (VEO): Towards Experience-based design and manufacturing for Industry 4.0

Syed Imran Shafiq[1], Cesar Sanin[1], Carlos Toro[2], and Edward Szczerbicki[3]

[1]*The University of Newcastle, University Drive, Callaghan, 2308, NSW, Australia*

[2]*Vicomtech-IK4, San Sebastian, Spain*

[3]*Gdansk University of Technology, Gdansk, Poland*

Address correspondence to Syed Imran Shafiq, The University of Newcastle, Newcastle, NSW, Australia. E−mail: SyedImran.Shafiq@uon.edu.au

Virtual Engineering Object (VEO): Towards Experience-based design and manufacturing for Cyber-Physical Systems and Industry 4.0

In this article we propose the concept, its framework and implementation methodology for Virtual Engineering Objects (VEO). A VEO is the knowledge representation of an engineering object having embodiment of its associated knowledge and experience. A VEO is capable of adding, storing, improving and sharing knowledge through experience. Moreover, it is demonstrated that VEO is a specialization of Cyber-Physical System (CPS). In this paper, it is shown through test models how the concept of VEO can be implemented with the Set of Experience Knowledge Structure (SOEKS) and Decisional DNA. The test model confirmed that the concept of VEO is able to capture and reuse the experience of engineering artefacts, which can be beneficial for efficient decision making in industrial design and manufacturing.

Keywords: Virtual engineering objects (VEO), Set of Experience Knowledge Structure (SOEKS), Decisional DNA (DDNA)

Background

Knowledge and experience hve been important assets for manufacturing organizations through the ages. Today's enterprises need to react and adapt to changes rapidly, and they are conscious that proper Knowledge Management (KM) processes will help them survive in a dynamic environment. Industrial designers and decision makers base their current decisions on lessons learned from previous similar situations (Sanin and Szczerbicki 2005b). However, much of an organization's experience is not properly capitalized at all because of inappropriate knowledge administration, leading to reprocessing decisions, high-response times, and lack of flexibility to adapt in dynamic environments. Knowledge-based industrial design and manufacturing techniques have been used in the past with considerable success. However, they have limitations such as being time-consuming, costly, domain-specific, unreliable in their intelligence, and unable to take previous experience into account (Danilowicz and Nguyen 1988, Qiu, Chui, and Helander 2008, Duong, Nguyen, and Jo. 2010). The role of knowledge in industrial design and engineering management has therefore become increasingly important for manufacturing companies.

Furthermore there is a lot of interest globally in 'Industry 4.0', which is termed as the fourth industrial revolution —following the steam engine, the conveyor belt, and the first phase of IT and automation technology. Industry 4.0 is a powerful concept, which promotes the computerization of traditional manufacturing plants and their ecosystems towards a connected and continuously available resources handling scheme through the use of Cyber Physical Systems (CPS) (n.a. 2014, Weber 2014). The goal is the intelligent factory, which is characterized by adaptability, resource efficiency, and ergonomics as well as the integration of customers and business partners in business and value processes (Weber 2014, Böhler 2012).

A number of authors exemplify the point that CPS is emerging as a "must have" technology crucial for industry (Baheti and Gill 2011, Lee 2008). CPS are integrations of computation with physical processes (Lee 2006, Lee 2008). 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 (Lee 2008). CPS aims to integrate knowledge and engineering principles across the computational and engineering disciplines (networking, control, software, human interaction, learning theory, as well as electrical, mechanical, chemical, biomedical, material science, and other engineering disciplines) to develop new CPS science and supporting technology.

In manufacturing, the potential for cyber-physical systems to improve productivity in the production process is vast. Consider processes that govern themselves, where smart products can take corrective action to avoid damages, and where individual parts are automatically replenished. 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, true and current requirements for the industry (Garcia-Crespo et al. 2010) . According to the European Commission under the Horizons 2020 programme, the self-learning closing feedback loop between production and design should be included in future factories for optimizing energy expenditure and minimizing waste as a direct relation to the enhancement in control and immediate information processing that a CPS will provide.

The Internet of Things will make a new wave of technological changes that will decentralize production control and trigger a paradigm shift in manufacturing. It is highly likely that the world of production will become more and more networked until everything is interlinked with everything else.

Considering the importance of the above mentioned aspects, this research proposes a novel approach to provide engineering artefacts with an experience-based representation. We introduce the concept of 'Virtual Engineering Object' (VEO), which permits dual computerized/real-world representation of an engineering artefact (Shafiq et al. 2014c, Shafiq, Sanin, and Szczerbicki 2014). VEO is a specialization of Cyber-Physical System (CPS) in terms of its extension in knowledge gathering and reuse, whereas CPS is only aimed towards data and information management. A VEO is a generalization of a CPS and its proposed conceptualization falls into a homogenous line of thinking aligned with our research presented previously (Shafiq et al. 2013, 2014a, b).

The concept of Virtual engineering Object uses a standard knowledge representation technique called Set of Experience Knowledge Structure (SOEKS), which comprises Decisional DNA (DDNA) (Sanin and Szczerbicki 2008a, 2005a). Decisional DNA is proposed as a unique and single structure for capturing, storing, improving and reusing decisional experience. Its name is a metaphor related to human DNA, and the way it transmits genetic information among individuals through time.

Figure1 summarizes the general idea of this research in an industrial domain where smart decisions can be made based on intelligent virtual objects and systems representing real-life machines, material, parts etc.

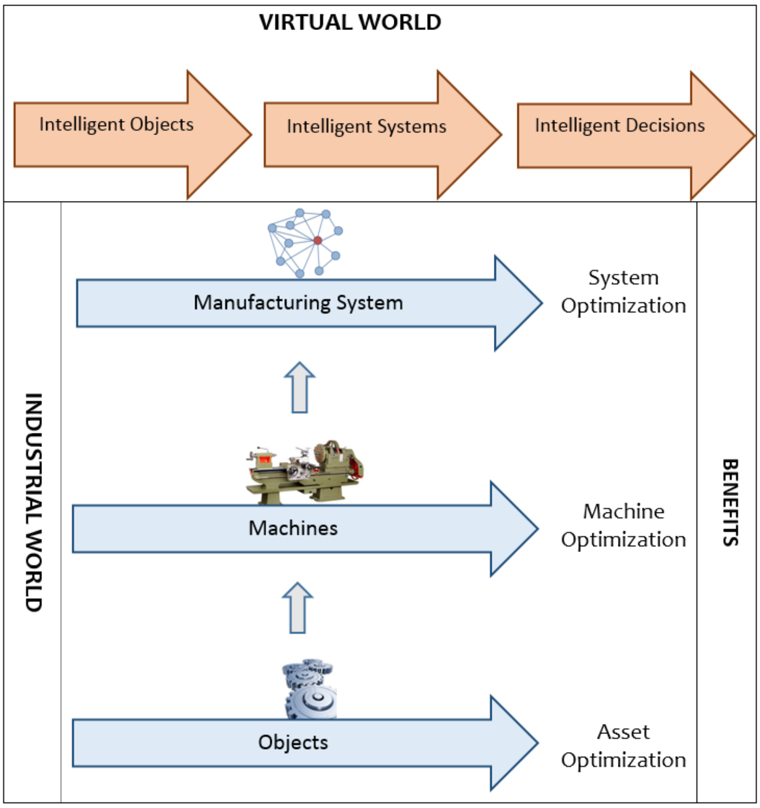

FIGURE 1. Cyber-physical perspective of our research

The structure of this paper is as follows. Section 2 presents the concept of SOEKS and DDNA which is the backbone of VEO. Section 3 illustrates conceptual background and architecture of Virtual Engineering. Section 4 describes the case study of VEO implementation with a number of experiments performed and their results. Finally, section 5 presents the conclusions and future work.

SET OF EXPERIENCE KNOWLEDGE STRUCTURE (SOEKS)

A Set of Experience (SOE) knowledge structure has been developed to store formal decision events in an explicit way (Sanin and Szczerbicki 2005a, 2006, 2007, 2008b, 2009, Sanin et al. 2012, Zhang, Sanín, and Szczerbicki 2013). SOE is a formal model of experience-based knowledge related to every-day decision-making events. Four basic components surround decision-making events: variables, functions, constraints, and rules. They are stored in a combined dynamic structure that comprises a Set of Experience. Identification of the *Variables* that intervene in the process of decision-making is the first step in the construction of the Set of Experience. These variables are the root of the structure, because they are the origin of the other components. *Functions*, the second component, describe associations between a dependent variable and a set of input variables; moreover, functions can be applied for reasoning optimal states, because they come out from the goals of the decision event. Therefore, a Set of Experience uses *Functions*, and establishes links among the variables constructing multi-objective goals. A *Constraint*, the third component of a Set of Experience, is a restriction of the feasibility of solutions in a decision problem, and a factor that limits the performance of a system with respect to its goals. Finally, *Rules* are suitable for associating actions with the conditions under which the actions should be performed. *Rules*, the fourth component of Set of Experience, are another form of expressing relationships among variables. They are conditional relationships that operate in the universe of variables. Rules are relationships between a condition and a consequence connected by the statements IF-THEN-ELSE. In conclusion, the Set of Experience consists of Variables, Functions, Constraints and Rules, which are uniquely combined to represent a formal decision event. SOEKS can be used in platforms to support decision-making, and new decisions can be made based on existing sets of experience. SOEKS is able to collect and manage explicit knowledge of different forms of formal decision events.

The concept of DDNA is the metaphor of human DNA. A group of SOE of the same category comprises of a kind of artificial decision making chromosome, as DNA does with genes. These chromosomes or groups of SOE make a category, and they are foundations for making decisions. Each module of chromosomes forms an entire inference tool, and a number of chromosons creates a Decisional DNA

VIRTUAL ENGINEERING OBJECT (VEO)

A VEO is knowledge representation of an engineering artefact. It has three features: (*i*) the embedding of the decisional model expressed by the set of experience, (*ii*) a geometric representation, and (*iii*) the necessary means to relate such virtualization with the physical object being represented.

A VEO acts as the object's living representation, capable of capturing, adding, storing, improving, sharing and reusing knowledge through experience, in a way similar to a human expert (Shafiq, Sanin, and Szczerbicki 2014).

Architecture of VEO

A VEO can encapsulate knowledge and experience of each important feature related with an engineering object. This can be achieved by gathering information from six different aspects of an object: Characteristics, Functionality, Requirements, Connections, Present State and Experience as illustrated in Fig. 2.

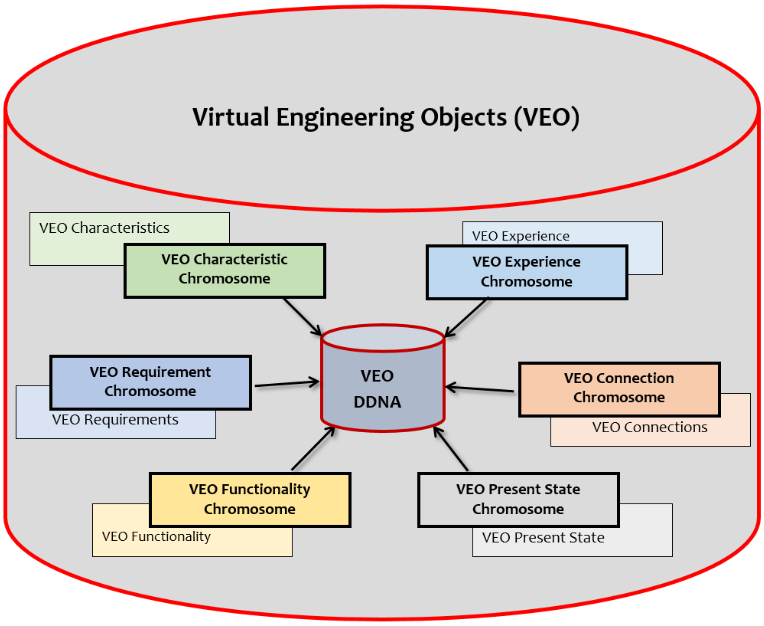

FIGURE 2. VEO Structure

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

- *Characteristics* - by describing the set of physical features and expected benefits offered by the artefact represented by the VEO. Not only will the information on its geometry dimensions, appearance, weight etc. be captured in this module, but also the possible concurrency attributes, for example, ‘versatility’ or ‘ease of operation’. Knowledge stored in *Characteristics* will assist in enhanced decision making, answering questions of the type: “Which VEO is best suited for a given physical condition?”
- The *Functionality* - by describing the basic workings of the VEO and principles on which it accomplishes its operation. Operational knowledge

related to an object such as time consumed, its working boundary limits, and the outcome of the process that is performed, will be stored in *Functionality*. This module of the VEO will assists in storing, selecting and reusing operation-specific details.

- *Requirements* - by describing the set of "necessities" of the VEO required for its precise working. Information on the kind and amount of power the VEO needs, on the required space, and on the extent of user expertise necessary for operating a VEO will be stored here.
- *Connections* - by describing how the VEO is related to other VEOs. Many engineering objects work in conjunction with other objects; these connecting VEO's may be a "part" or may be a "need" of each other. This module of VEO structure will be essential for the scaling up and establishing the interconnection of VEO's in a manufacturing scenario.
- The *Present State* of the VEO - by highlighting parameters of the VEO at the current moment. It will answer the question whether the VEO is ready for a particular operation. If the required VEO is busy, it will predict the expected time when it is free for the next operation.
- The *Experience* of the VEO - by including 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 to the VEO will be stored in the *Experience*. This element of the VEO will keep on updating in real time together with every activity in which VEO takes place (Shafiq, Sanin, and Szczerbicki 2014).

As discussed previously, VEO is a knowledge representation for an engineering artifact. We must take into account that when we say 'an engineering artifact', we can be talking about something simple like a valve, or we can be talking about something complex like a painting cell. For such reason, the VEO specification must have a complexity level concurring its functionality. In our very first approximation, we identified four different levels of VEO, as can be seen in Figure 3.

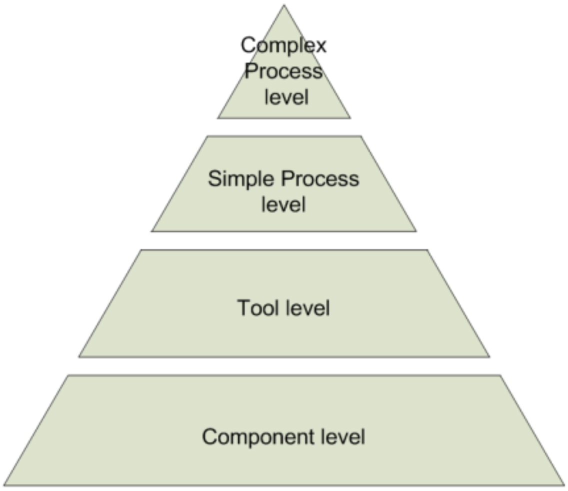

FIGURE 3. VEO complexity pyramid.

At Component level, VEO represent just a component (usually a part of kind of machinery). By itself, this component has not any functionality that can be considered "useful" in a production process. Of course, it has its functionality in the machinery where it is part of. Examples of VEO at this level can be valves, printed circuit boards, etc.

Above the Component level there is the Tool level. VEOs placed here represent those artefacts that have a basic functionality, being considered as useful unities in an industrial process. Nevertheless, they do not constitute an industrial process by itself. An example of VEO at this level can be a robot that picks an object and moves it to another position.

Next level is Simple Process level. In this level, we consider that VEO represent artefacts which accomplish a full simple process. We consider a simple process those processes that made a simple change in the ‘product’ that is involved in it. An example could be a painter cell (where the simple process is painting; the product enters in one color and exits in another one).

Finally, at the top of the complexity pyramid is the Complex Process level. The complex process level VEO is a combination of various simple process level VEOs. An example could be car door manufacturing (where many simple processes take place, like welding, painting etc.).

Implementation of VEO

For the purpose of implementation of VEO, we integrated it with the Decisional DNA consisting of SOEKS containing Variables, Functions, Constraints and Rules. In Section 3, we also discussed that a VEO structure includes elements like Characteristics, Functionality, Requirements, Connections, Present State and Experience. SOEKS are created for each of the above elements of the individually. The goal behind this approach is to provide a more scalable setting, similar to the one that could 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 of VEO, representing experience and knowledge (Figure 4).

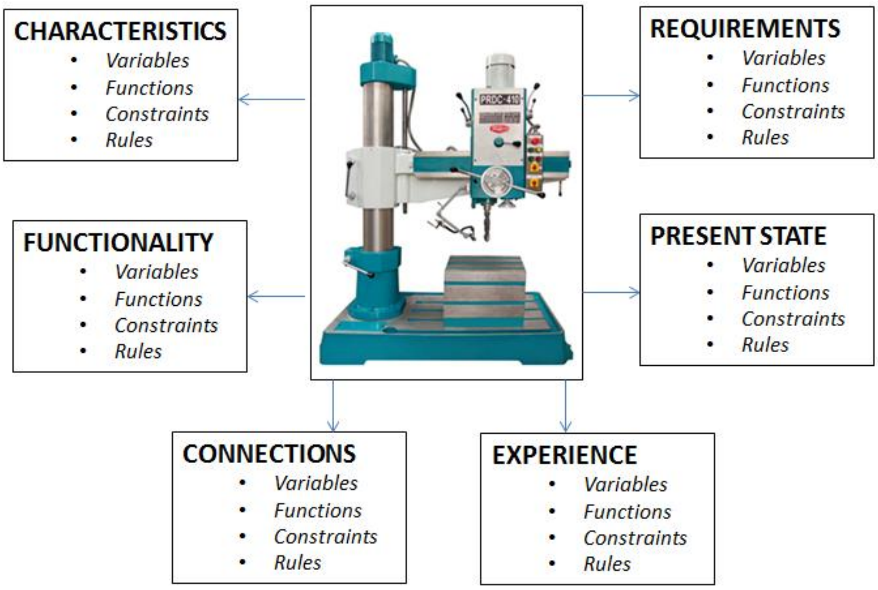

FIGURE 4. Structure of a VEO

Case Study

As a case study, we considered a manufacturing set up having three different drilling machines, drilling tools, and work holding devices (Fig. 5). Information and specifications about these above mentioned engineering objects were gathered from standard sources and data is stored according to the SOEKS format. Moreover, every formal decision taken is also stored as a SOE, which leads to the formation of interconnected VEO's.

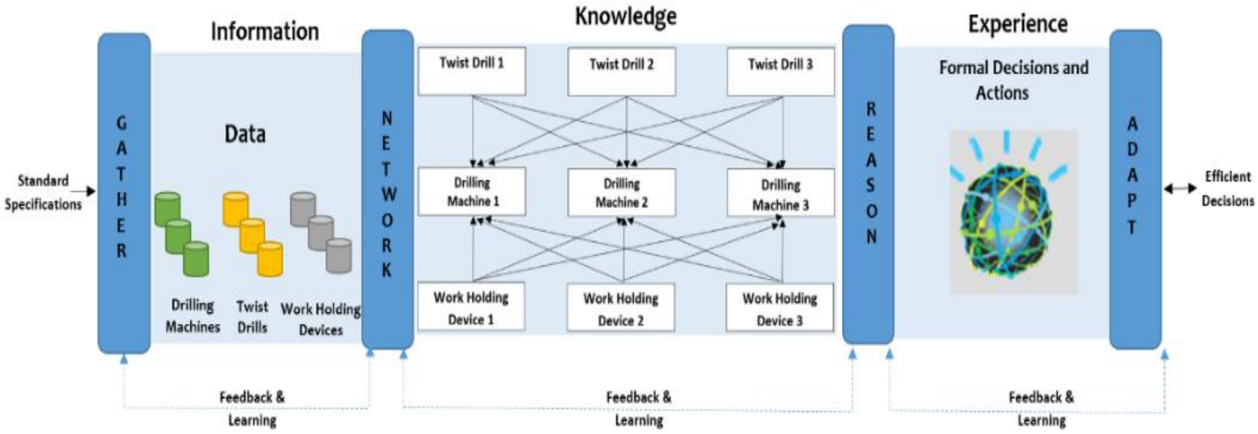

FIGURE 5. Framework for the case study

The objective of this case study is not only to develop VEO's for engineering artefacts but also to demonstrate that different VEOs connect and form a network. Furthermore, to prove that the experience captured from this VEO network can be reused for better future decision making and efficient utilization of resources.

First, the necessary drilling associated information was identified and effort was made to capture and store the relevant information of the VEO adhering to the format shown in Fig 4. CSV (Comma Separated Values) files for Characteristics, Functionality, Requirements, Present State, Connections and Experience were build. Table 1 shows the structure of one a CSV file for Characteristics. Similar files were developed for other VEO elements.

Table 1. CSV file format for VEO Experience

Variables																	
OpNo	veo1	veo2	veo3	WPMaterial	WPDepth	Operation	Drill/ToolDia	Feed	CuttingSpeed	DD-F-S	SpindleSpeed	fceRough	CoolentUsed	Breakdown	MachiningTime	Date
1	DM1	T1	H1	MS	47	Drilling	5	0.7	760	5-0.7-760	2297	SMOOTH	YES	NO	1.753840413	30/06/2012
2	DM1	T1	H1	MS	48	Drilling	5	0.5	570	5-0.5-570	819	ROUGH	YES	NO	5.023547881	14/08/2012
2186	DM3	T3	H3	CI	49	Boring	30	0.35	90	30-.35-90	2639	ROUGH	YES	NO	1.591511936	23/01/2010
2187	DM3	T3	H3	CI	53	Boring	30	0.3	75	30-.30-75	1081	ROUGH	YES	NO	4.202458041	28/11/2010

Constraints
veo1 = {DM1, DM2, DM3}
veo2 = {T1, T2, T3}
veo3 = {H1, H2, H3}
WPMat = {MS, HSS, CI}
SurfaceFinish = {SMOOTH, ROUGH}
CoolentUsed = {YES, NO}
Breakdown = {YES, No}

Functions
MachiningTime = (60 * SpindleSpeed) / (3.24 * DepthOfHole)

With CSV files in hand, a parser was written in Java programming language. Figure 6 shows the simplified JAVA class diagram for the parser (*veoParserCSV*) which reads variables, functions, and constraints from Table 1 and creates SOEKS.

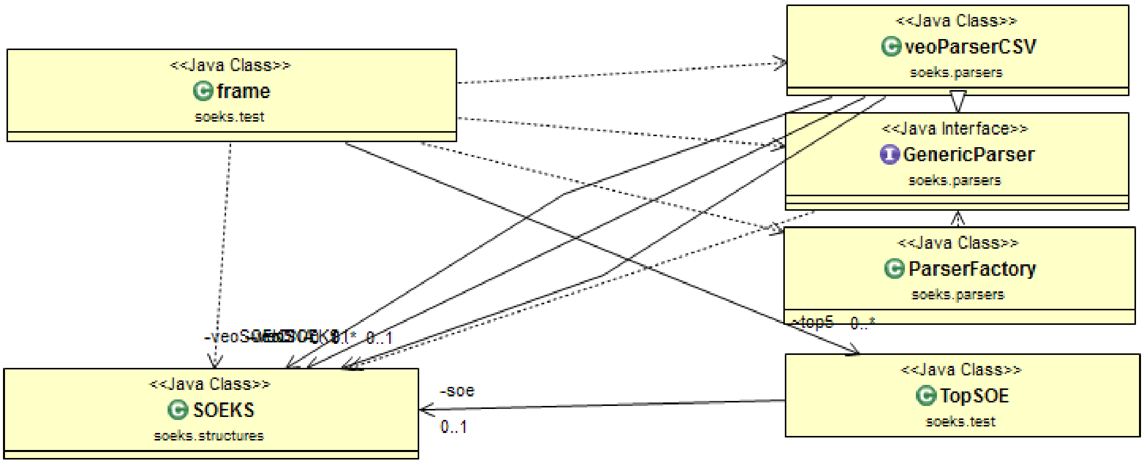

FIGURE 6. JAVA Class Diagram for VEO Parser

The above parsing procedure is repeated for all VEO elements for which there are corresponding CSV files similar to the one shown in Table 1 collecting SOEs in the final form of Decisional DNA that can be queried (Sanin, Szczerbicki, and Toro 2007). A query is used to exemplify this case study (Figure 7).

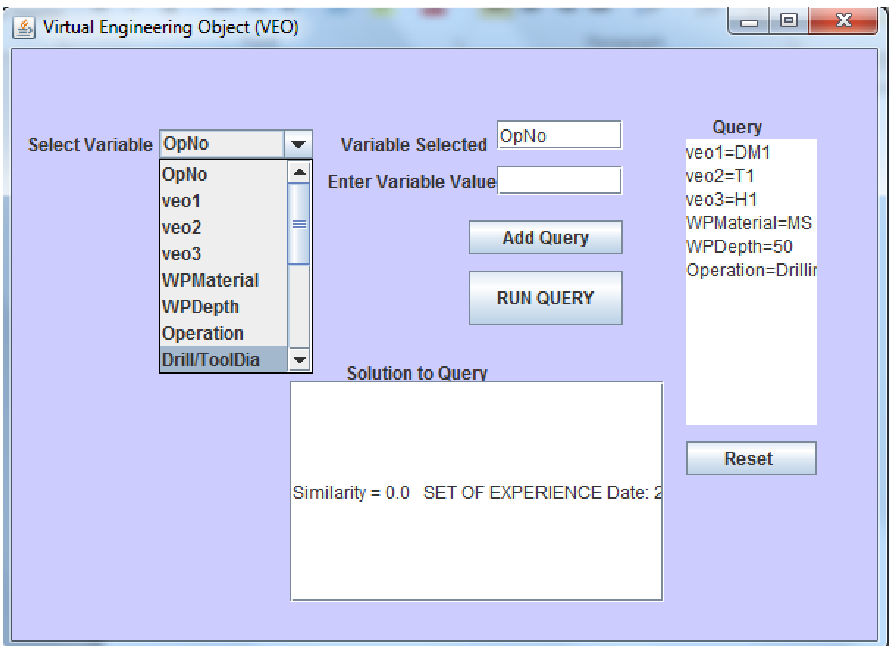

FIGURE 7. Graphical User Interface (GUI) for building a query

Figure 7 shows GUI for creating a query. User selects variables and their values to be queried. These variables are converted into a SOE and then this new SOE is compared for similarity with each SOEs already stored in the VEO Decisional DNA (VEO DDNA). For each comparison a similarity factor is calculated and top 5 most similar SOEKS from VEO DDNA are returned. Figure 8 sums up in a flowchart the above process.

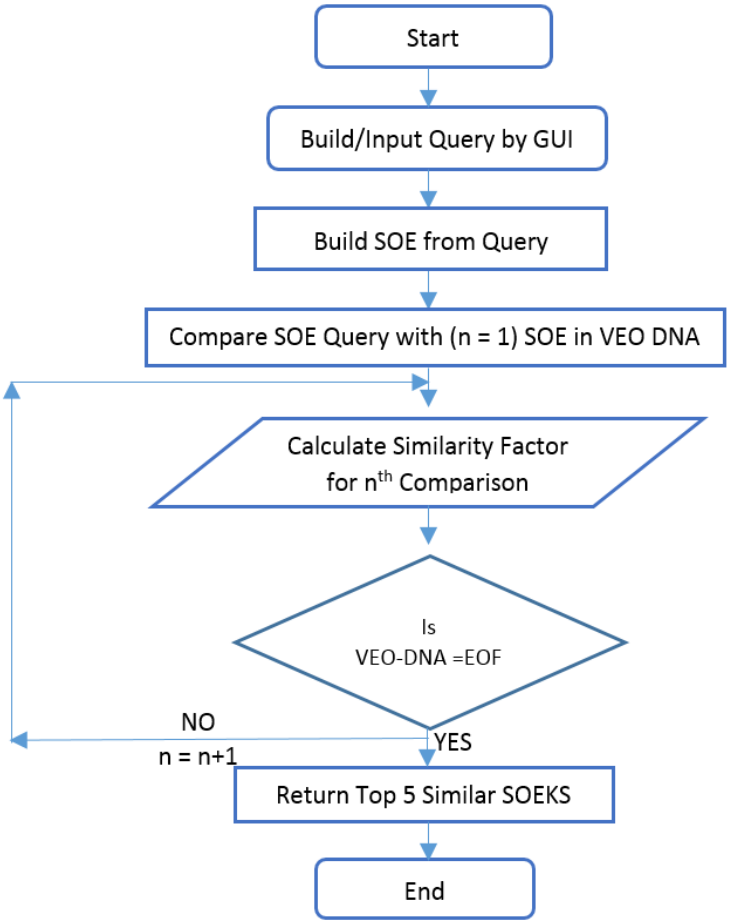

FIGURE 8. Flowchart for running a query

Results and discussion

This VEO architecture was implemented in Java programming language on a Windows 7 operating system. The VEO DDNA consists of SOEs representing Characteristics, Functionality, Requirements, Present State, Connections and Experience, each one having 53 variables, 3 functions and 28 constraints. For testing purposes, we query the repository of 2256 SOEKS.

The parsing process of the VEO DDNA was executed, producing a parsing time of 703.0 ms as it can be observed from Fig. 9 (with Experience CSV file taking 509.0 ms). This is considered a very good time taking into account that those SOE are quite complex due to the substantial number of variables, functions and constraints, adding up to a total of 84 key features per formal decisional event.

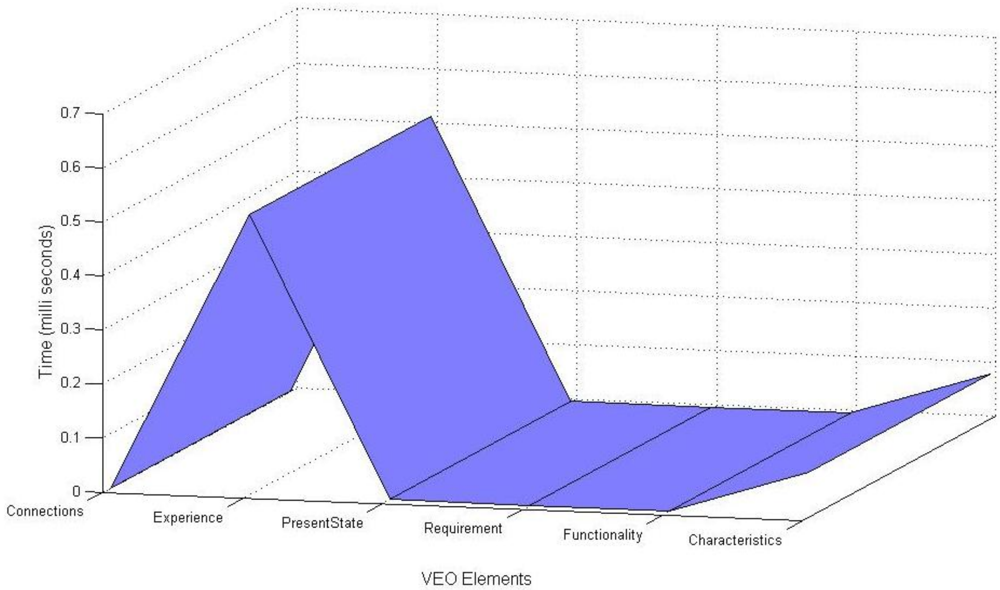

FIGURE 9. Parsing time versus VEO elements

The detailed parsing process of the VEO decisional chromosome produced an average parsing time per SOE of 228.0 ms (Variables = 537.0, Functions = 67.0 ms, Constraints = 80.0 ms). It can be noticed from Fig. 10 that most of the time is dedicated to the variables due to their large number in VEO knowledge representation.

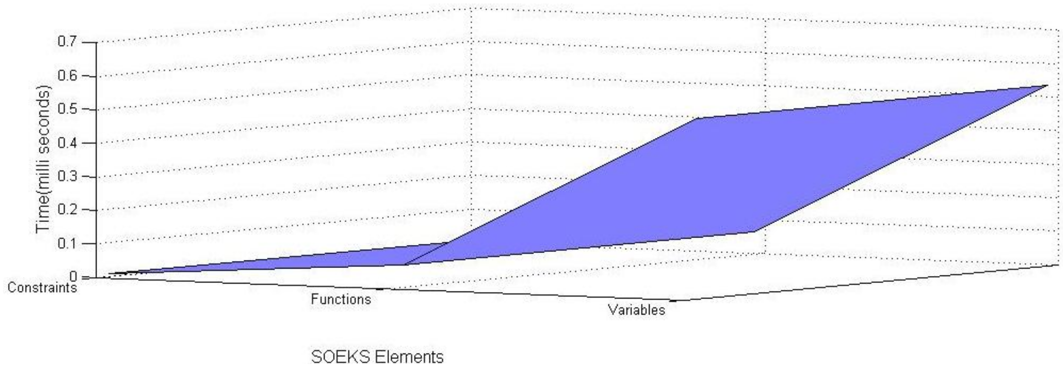

FIGURE 10. Parsing time versus SOEKS elements

The searching process for a similar SOE within the VEO DDNA was executed 100 times, producing an average time around 6.49 ms to find the closest existing match. Figure 11 illustrates the plot for time taken to parse VEO DDNA 100 times to search for similar SOE. And average parsing time for each SOEKS is 0.0131ms, Figure 12 plots time taken for each 2254 SOEKS to find top 5 similar SOEKS. This is considered an excellent time for such large group of SOE and the required number of similarity comparisons performed

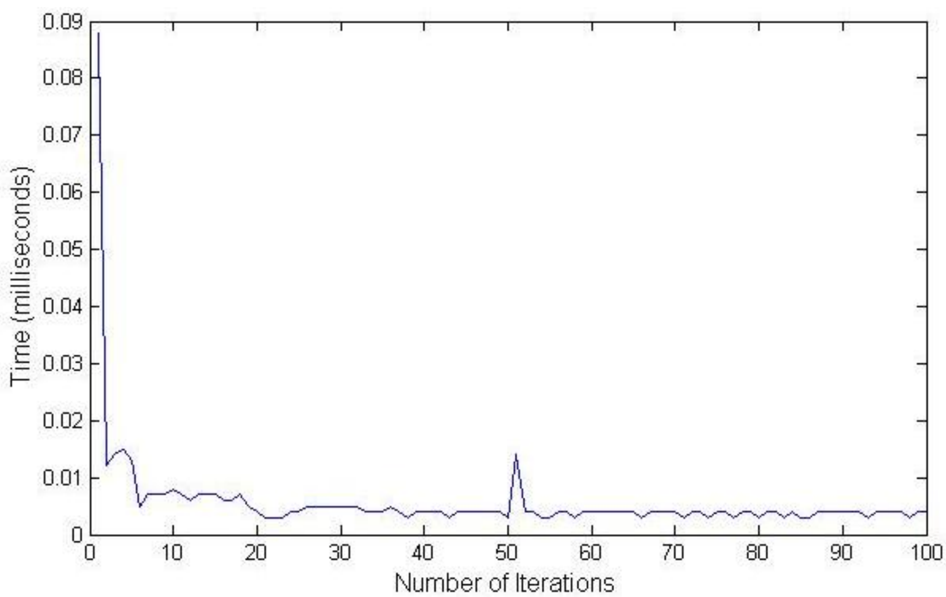

FIGURE 11. Parsing times for 100 iterations for VEO

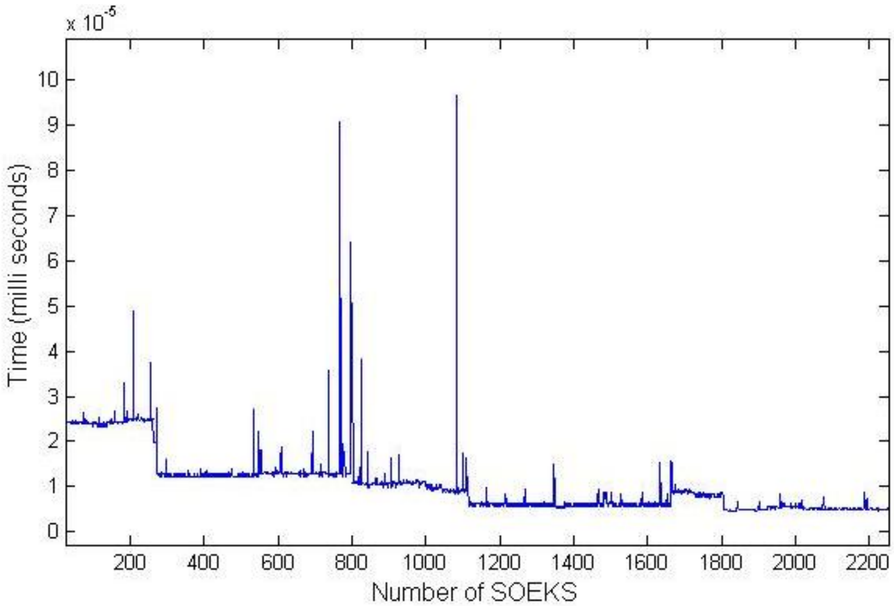

FIGURE 12. Parsing time per SOEKS

CONCLUSIONS AND FUTURE WORK

The main contribution of the work presented throughout this article is the formal definition of mechanisms leading to implementation of knowledge engineering in the manufacturing field for future cyber-physical systems as required by Industry 4.0. It explains through a practical case the process of collecting experience from engineering artefacts, and then using this information for the construction VEO Decisional DNA. Decisional DNA and the Set of Experience were applied as a knowledge representation structure for capturing and storing experience. Afterwards, they were used as tools for decision making processes that can enhance different manufacturing systems with predicting capabilities and facilitates knowledge engineering processes for decision making. The relation between CPS and VEO 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 standard CPS is aimed at data and information gathering and management.

Future work on this proposal includes developing a network of VEOs having a wide variety of engineering objects ranging from simple standalone artefacts to a complex multitasking machines. In addition, it is desirable to evaluate the proposed techniques in real-life operational contexts in order to refine, improve, and determine the actual applicability of the proposal presented in this article.

REFERENCES

Baheti, Radhakisan, and Helen Gill. 2011. Cyber Physical Systems. In *The Impact of Control Technology*, edited by T. Samad and A.M. Annaswamy: www.ieeecss.org.

Böhler, T. M. 2012. Industrie 4.0-Smarte Produkte und Fabriken revolutionieren die Industrie. *Produktion Magazin*.

Danilowicz, C., and N. T. Nguyen. 1988. "Consensus-Based Partitions in the Space of Ordered Partitions." *Pattern Recognition Letters* no. 21 (3):269-273.

Duong, T. H., N. T. Nguyen, and G. S. Jo. 2010. "Constructing and Mining—A Semantic-Based Academic Social Network." *Journal of Intelligent & Fuzzy Systems* no. 21 (3):197-207.

Garcia-Crespo, A., B. Ruiz-Mezcua, J. L. Lopez-Cuadrado, and J. M. Gomez-Berbis. 2010. "Conceptual model for semantic representation of industrial manufacturing processes." *Computers in Industry* no. 61 (7):595-612. doi: http://dx.doi.org/10.1016/j.compind.2010.01.004.

Lee, Edward. 2008. Cyber Physical Systems: Design Challenges. University of California, Berkeley.

Lee, Edward A. 2006. Cyber-Physical Systems - Are Computing Foundations Adequate? In *In Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*. Austin, TX.

n.a. *Zukunftsprojekt Industrie 4.0*. http://www.bmbf.de/de/9072.php 2014.

Qiu, Yuan Fu, Yoon Ping Chui, and Martin G. Helander. 2008. "Cognitive understanding of knowledge processing and modeling in design." *Journal Of Knowledge Managementimited* no. 12 (2):156-168. doi: 10.1108/13673270810859587.

Sanin, Cesar, and Edward Szczerbicki. 2005a. "A complete example of set of experience knowledge structure in XML." In *Knowledge Mangement:Slected issues*, edited by A. Szuwarzynski, 99-112. Gdank: Gdansk University Press.

Sanin, Cesar, and Edward Szczerbicki. 2005b. "Set of Experience: A Knowledge Structure for Formal Decision Events." *Foundations of Control and Management Sciences* no. 3:95-113.

Sanin, Cesar, and Edward Szczerbicki. 2006. "Extending Set Of Experience Knowledge Structure into a Transportable Language extensible Markup Language." *Cybernetics* *and Systems* no. 37 (2-3):97-117. doi: 10.1080/01969720500425046.

Sanin, Cesar, and Edward Szczerbicki. 2007. "Towards the construction of Decisional DNA: A Set of Experience Knowledge Structure JAVA Class within an Ontology System." *Cybernetics and Systems* no. 38 (8):859-878. doi: 10.1080/01969720701601189.

Sanin, Cesar, and Edward Szczerbicki. 2008a. *Decisional DNA and the Smart Knowledge Mangement System: A process of transforming information into knowledge* Edited by A. Gunasekaran, *Techniques and Tool for the Design and Implemention of Enterprise Information Systems*. New York: IGI.

Sanin, Cesar, and Edward Szczerbicki. 2008b. "Towards Decisional DNA: Developing Holistic Set of Experience Knowledge Structure." *Foundation of Control and Management Science* no. 9:109-122.

Sanin, Cesar, and Edward Szczerbicki. 2009. "Application of a Multi-domain Knowledge Structure: The Decisional DNA, in Intelligent Systems for Knowledge Management." In, 65-86. Springer-Verlag.

Sanin, Cesar, Edward Szczerbicki, and Carlos Toro. 2007. "An OWL Ontology of Set of Experience Knowledge Structure." *Journal of Universal Computer Science* no. 13 (2):209--223.

Sanin, Cesar, Carlos Toro, Zhang Haoxi, Eider Sanchez, Edward Szczerbicki, Eduardo Carrasco, Wang Peng, and Leonardo Mancilla-Amaya. 2012. "Decisional DNA: A multi-technology shareable knowledge structure for decisional experience." *Neurocomputing* no. 88 (0):42-53. doi: http://dx.doi.org/10.1016/j.neucom.2011.08.029.

Shafiq, Syed Imran, Cesar Sanin, and Edward Szczerbicki. 2014. "Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA): Past, Present and Future." *Cybernetics and Systems* no. 45 (02):200-215. doi: 10.1080/01969722.2014.874830.

Shafiq, Syed Imran, Cesar Sanin, Edward Szczerbicki, and Carlos Toro. 2013. Using Decisional DNA To Enhance Industrial And Manufacturing Design: Conceptual Approach. In *Information Systems Architecture and Technology*, edited by Leszek Borzemski Jerzy Świątek, Adam Grzech, Zofia Wilimowska. Szklarska Poreba, Poland: Wrocław University of Technology, Wrocław.

Shafiq, Syed Imran, Cesar Sanin, Edward Szczerbicki, and Carlos Toro. 2014a. "Implementing Virtual Engineering Objects (VEO) with the Set of Experience Knowledge Structure (SOEKS)." *Procedia Computer Science* no. 35 (0):644-652. doi: http://dx.doi.org/10.1016/j.procs.2014.08.146.

Shafiq, Syed Imran, Cesar Sanin, Edward Szczerbicki, and Carlos Toro. 2014b. "Virtual Engineering Objects (VEO): Designing, Developing and Testing Models." In *System Analysis Approach to the Design,Control and Decision Support*, edited by L Borzecki A Grzech, J. Swiatek, Z. Wilimowska, 183-192 Wroclaw: Wroclaw University of Technology Press.

Shafiq, SyedImran, Cesar Sanin, Edward Szczerbicki, and Carlos Toro. 2014c. "Decisional DNA Based Framework for Representing Virtual Engineering Objects." In *Intelligent Information and Database Systems*, edited by NgocThanh Nguyen, Boonwat Attachoo, Bogdan Trawiński and Kulwadee Somboonviwat, 422-431. Springer International Publishing.

Weber, M. *Industry 4.0*. http://www.pt-it.pt-dlr.de/de/3069.php 2014.

Zhang, Haoxi, Cesar Sanín, and Edward Szczerbicki. 2013. "Implementing Fuzzy Logic To Generate User Profile In Decisional DNA Television: The Concept and Initial Case Study." *Cybernetics and Systems* no. 44 (2-3):275-283. doi: 10.1080/01969722.2013.762280.