Decisional DNA Based Framework for Representing Virtual Engineering Objects

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

[1] The University of Newcastle, University Drive, Callaghan, 2308, NSW, Australia
{syedimran.shafiq,Cesar.Maldonadosanin}@uon.edu.au
[2] Gdansk University of Technology, Gdansk, Poland
Edward.Szczerbicki@newcastle.edu.au
[3] Vicomtech-IK4, San Sebastian, Spain
ctoro@vicomtech.org

Abstract. In this paper, we propose a frame-work to represent the Virtual Engineering Objects (VEO) utilizing Set of Knowledge Experience Structure (SOEKS) and Decisional DNA. A VEO will enable the discovery of new knowledge in a manufacturing unit and the generation of new rules that drive reasoning. The proposed VEO framework will not only be knowledge based representation but it will also have its associated experience embedded within it. This concept will evolve and discover implicit knowledge in industrial plant, which can be beneficial for the engineers and practitioners. A VEO will be a living representation of an object; capable of adding, storing, improving and sharing knowledge through experience, similar to an expert of that area.

Keywords: Decisional DNA (DDNA), Set of Experience Knowledge Structure (SOEKS), Virtual Engineering Objects (VEO).

1 Introduction

The manufacturing companies are facing intense pressure from the market, demanding customers and technological advancement. So, need of the hour is to take effective decisions, to develop products within time and cost constraints without compromising on the quality. In order to achieve these objectives, companies are taking support of various technologies, Knowledge based engineering is one such technology. Organizations generate new knowledge by solving problems, however due to lack of appropriate knowledge management techniques; knowledge has to be reprocessed in order to solve new problems with similar conditions. Knowledge representation reuses this information to make future decisions in a more efficient way, without wasting time and resources [1].

The use of knowledge based manufacturing and industrial design is arguably unexplored. Researches have shown that a large percentage of time during industrial design is spent on routine tasks. It is also observed that around 20% of the designer's time is spent searching for and absorbing information, and '40% of all design information requirements are met by personal stores [2]. It is clear that industrial design

N.T. Nguyen et al. (Eds.): ACIIDS 2014, Part I, LNAI 8397, pp. 422–431, 2014.

and manufacturing information is not represented in a shared and easily accessible knowledge base. Knowledge based industrial design techniques have been used in the past with fair bit of success. Nevertheless they have their share of limitations like they may be time consuming, costly, domain specific and at times not very intelligent.

This work proposes a novel concept of Virtual Engineering Objects (VEO). A VEO will have all the knowledge of the engineering artifact along with the associated experience embedded in it. This will help the practitioners in effective decision making based on the past experience. A novel technique of knowledge representation called Set of Experience knowledge structure (SOEKS) and Decisional DNA [3] is used for developing VEO.

The structure of this paper is as follows-section 2 describes the concept of SOEKS and DDNA. In section 3, we introduce the idea of VEO. In section 4, implementation and formulation of the VEO is discussed. In section 5, we conclude this paper and section 6presents ideas for further research.

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

As discussed in section 1, a large amount of previous knowledge is needed to design and manufacture a new component; the information may not be exactly the same but may be from the family of the related object. However, it has been observed that not much effort is made in the past to retain the knowledge. Knowledge and experience are lost indicating that there is a clear deficiency on its collection and reuse [4]. Some of the reasons for lack of knowledge base are:

- the non-existence of a common knowledge-experience structure which is able to collect multi-domain formal decision events, and
- the non-existence of a technology able to capture, store, improve, retrieve and reuse such collected experience [5].

Sanin and Szczerbicki proposed a new smart knowledge based decision support tool called Set of Knowledge Experience Structure (SOEKS) and Decisional DNA, having three important elements:

- a knowledge structure able to store and maintain experiential knowledge.
- a solution for collecting experience that can be applied to multiple applications from different domains.
- a way to automate decision making by using such experience, that is, retrieve collected experience by answering a query presented [3, 6, 7].

The SOEKS is a compound of variables (V), functions (F), constraints (C) and rules (R), which is uniquely combined to represent a formal decision event. Functions define relations between a dependent variable and a set of input variables; therefore, SOEKS uses functions as a way to establish links among variables and to construct multi-objective goals (i.e., multiple functions). Similarly, constraints are functions that act as a way to limit possibilities, restrict the set of possible solutions, and control the performance of the system with respect to its goals. Finally, rules are used to represent inferences and correlate actions with the conditions under which they should be executed. Rules are relationships that operate in the universe of variables and express the connection between a condition and a consequence in the form ‘if then else’ [3, 8].

Chromosomes are groups of (Set of Experience) SOE that can accumulate decisional strategies for a specific area of an organization. Multiple SOE can be collected, classified, and organized according to their efficiency, grouping them into decisional chromosomes. Finally, sets of chromosomes comprise what is called the Decisional DNA of the organization as shown in Figure 1.

DDNA is a metaphor of human DNA, because of its ability to capture and carry information (experience and knowledge).It has four elements provided by the SOEKS (variables, functions, constants and rules) which can be related in the same manner to the elements in DNA (adenine, thymine, guanine and cytosine).

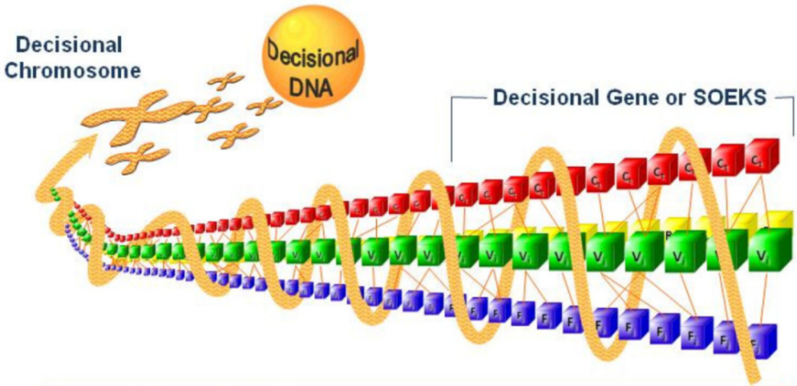

Fig. 1. SOEKS and Decisional DNA [5, 6]

3 Virtual Engineering Objects (VEO)

A Virtual Engineering Object (VEO) is a living representation of an artefact having knowledge and experience embedded within it and can behave like an expert of that area [9].

3.1 Is VEO knowledge Representation?

In this section we examine that whether a VEO can be termed as knowledge representation. According to Davis [10], a knowledge representation notion can be understood in terms of five fundamental roles that it play:

- A Knowledge Representation is a surrogate.

Viewing knowledge representations as surrogates leads naturally to two important questions. The first question about any surrogate is its intended identity: what is it a surrogate for? There must be some form of correspondence specified between the surrogate and its intended referent in the world; the correspondence is the semantics for the representation. In VEO case, this surrogate will be mainly oriented to decision making process for industrial plants.

The second question is fidelity: how close is the surrogate to the real thing? Which attributes of the original does it captures and make explicit, and what does it omit? Perfect fidelity is generally impossible, both in practice and in principle. It is impossible in principle because the only completely accurate representation of an object is the object itself.

VEO model intends to be the most complete possible model for a specified domain. Probably, it won't be necessary to have a perfect geometric model of each element, but many physical characteristics are fundamental, as they are the set of requirements and characteristics specified for each artifact. What these characteristics and requirements are that will be discussed in section 3.2. It is also important to highlight the experience that can be obtained from the use of that artifact.

- A Knowledge Representation is a set of ontological commitments.

Selecting a representation means making a set of ontological commitments. The commitments are in effect a strong pair of glasses that determine what we can see, bringing some parts of the world into sharp focus, at the expense of blurring others.

In VEO case, our domain is the industrial plant, being the focused part the industrial processes, and the blur part, all the administrative work done in a real plant.

- A Knowledge Representation is a fragmentary theory of intelligent reasoning.

In VEO case, we will use technologies like Reflexive Ontologies, SOEKS and Semantic Reasoners. This set of tools will allow us to infer new knowledge from both explicit knowledge and experience [11].

- A Knowledge Representation is a medium for efficient computation.

In VEO case, as we said before, we will use semantic tools like Reflexive Ontologies. To model these ontologies, we will use a methodology based on engineering standards. Using this methodology, we can assure that information is organized in a standard way, being the easiest way to facilitate the inference. In addition, Reflexive Ontologies are more efficient when queried than standard ontologies.

- A Knowledge Representation is a medium of human expression.

Using a methodology based on engineering standards assures that a VEO is a way of communication among people related to industry.

From the above deliberation it is established that VEO fulfills the requirements to be qualified as knowledge representation.

3.2 VEO Structure

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 according to its functionality. In our very first approximation, we identified four different levels of VEO, as can be seen in Figure 2.

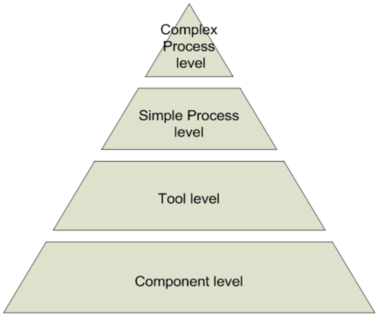

Fig. 2. VEO complexity pyramid

At Component level, VEO represent just a component (usually from any 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 is the Tool level. VEOs placed here represent those artifacts 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 pick an object and move it to another position.

Next level is Simple Process level. In this level, we consider that VEO represent artifacts 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.).

After classifying VEO on the basis of complexity level, in the next step, we propose a structure for a VEO. A VEO comprises of knowledge of 5 different aspects/parts of an object to be represented, as depicted in Figure 3. Characteristics, Functionality, Requirements, Connections and Present State are designed in such a way that they consists various variables, which can represent maximum facets of the object. Besides having knowledge these different modules of a VEO have their past experience embedded in them. The main features of these modules are as follows:

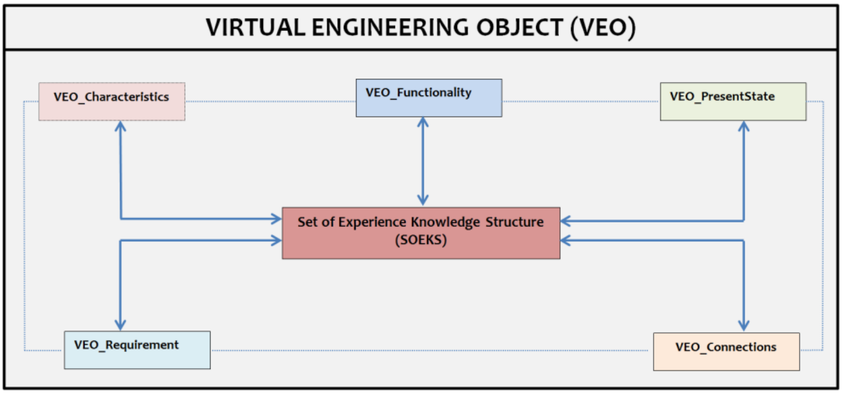

Fig. 3. Proposed VEO Structure

- *Characteristics* describe the set of expected benefits offered by the artefact represented by the VEO. Such characteristics will depend on what kind of artefact are we considering. For example, if our VEO is a SCARA robot, some characteristics will be the action range, functionalities, max speed, etc., but if our VEO is a software application, characteristics focused will be input/output formats etc.
- *Functionality* describes the basic working of the object and principle on which it works and accomplishes its operation.
- *Requirements* describe the set of necessities of the VEO for its correct work. In a similar case like characteristics, the set of requirements depend on what kind of VEO we are considering. For the SCARA Robot we can mention power supply, required space, etc. for the software application the requirements will be associated to operative system, memory etc.
- *Connections* describe how the VEO is related with other VEOs. These connections can be of different types. Some of them can be a *need* relationship, e.g. a robot that needs a computer application to control it. Other kind of relation can be, of course, *part of* relationship, e.g. a gear is part of an engine.
- The present state of the VEO indicates parameters of the VEO in the current moment. For example, information like for how much time has been this machine powered on? Or the machine is busy or idle at present? [9]

SOEKS is connected with the rest of the parts i.e. characteristics, functionality, requirements, connections and the present state and is able to capture and store new information. SOEKS contains the knowledge and experience acquired about a particular VEO over a span of time.

4 Formulation and Implementation

In the previous sections, we discussed the complexity levels and the structure of the VEO. This section presents the proposed structure of SOEKS conceived for the VEO.

The prototype was implemented using Java (Oracle 2011). The structures of all the modules of VEO are shown in Figure 4. JAVA variables are designed and perceived in such a way that they are able to represent most of the features of the engineering objects related to manufacturing.

SOEKS files for each module (Characteristics, functionality, requirement, connection and present state) 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 range of engineering objects.

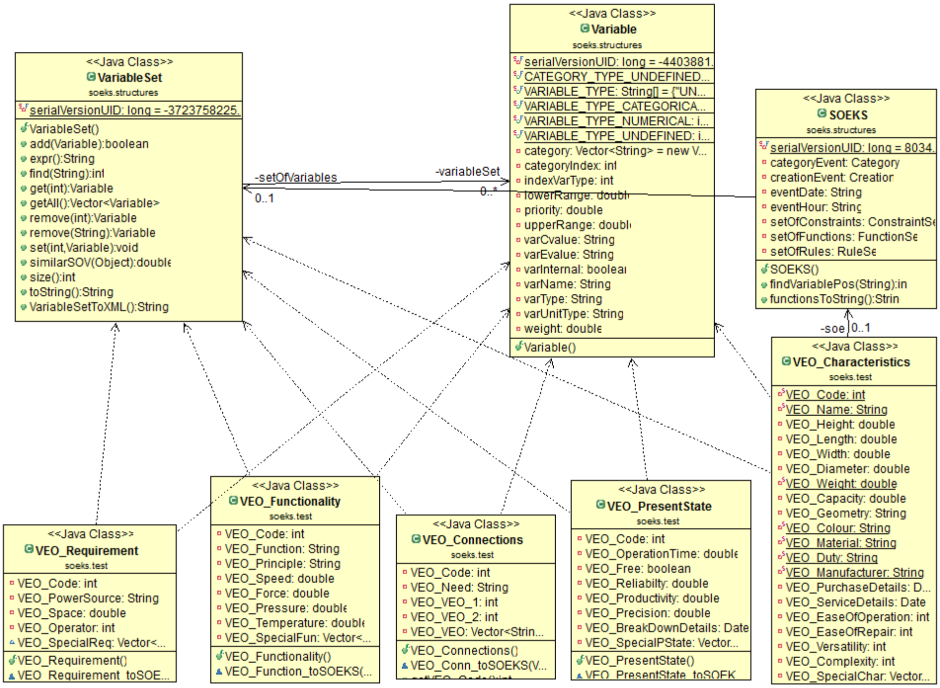

Fig. 4. Class diagram for VEO

Weights are assigned to the attributes of the variables of the above mentioned modules of an artefact, and then the five sets of SOEKS are generated. These individual SOEKS are combined under an umbrella (VEO), representing experience and prediction.

The output given below shows the XML representation of the attributes of one of the variable (Height of the VEO). In the similar fashion all the variables of the five modules of a VEO are assigned.

```
<set_of_variables>
<!-- Variables included in the model -->
<variable>
<var_name>VEO_Height</var_name>
<var_type>Numerical</var_type>
<var_cvalue>20.0</var_cvalue>
<var_evalue>20.0</var_evalue>
<unit>Metres</unit>
<internal>true</internal>
<weight>1.0</weight>
<l_range>1.0</l_range>
<u_range>1.0</u_range>
<categories>
<category>CATEGORY UNDEFINED</category></categories>
<priority>1.0</priority>
</variable>
```

Once VEO of all the engineering artifacts of a manufacturing plant is developed, then a network of the interconnected VEOs, shown in Figure 5 is established. These relations/connections will be based on the connection made in the VEO-Connections in section 3.2.

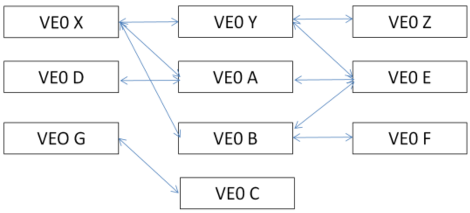

Fig. 5. Network of Interconnected VEO's

The next step in our plan, after getting the knowledge representation of all the engineering artifacts of a manufacturing unit is to extract the right knowledge out of it. Figure 6 shows the flowchart of achieving it. When a query is loaded in the DDNA, it will search its SOEKS and according to it will propose the VEO's that can perform the desired task or process. VEOs will be having their knowledge and past experience implanted in them, which in turn will help in deciding specific parameters for the queried task. So DDNA will give its complete and explicit recommendation to perform that specific task. If the recommendation is accepted and executed by the user then SOEKS will update itself by including the information of this particular task in its repository. If it is not, then DDNA will execute the query again and will present the next best solution according to its experience.

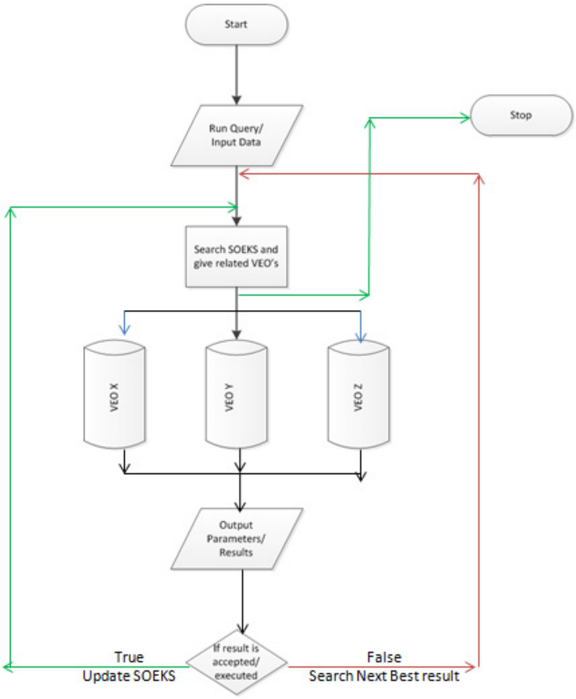

Fig. 6. Flowchart of an extracting information

5 Conclusion

In this article, we presented an approach to represent engineering artifact based on knowledge and experience. We described the architecture of our approach and implementation that uses SOEKS/DDNA to represent VEO. We demonstrated this approach through some initial tests. As the illustrative result shows, we can model and represent engineering artifact virtually, which can capture, store and reuse the associated knowledge and experience of the object. The Decisional DNA, as a novel knowledge representation structure, not only can be easily applied to the concept of VEO but indeed improves the decision making process in manufacturing units using experience.

6 Future Work

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

- 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. Verhagen, W.J.C., Garcia, P.B., Van Dijk, R.E.C., Curran, R.: A critical review of Knowledge-Based Engineering: An identification of research challenges. Advanced Engineering Informatics 26, 5–15 (2012)
2. Baxter, D., Gao, J., Case, K., Harding, J., Young, R., Cochrane, S., Dani, S.: An engineering design knowledge reuse methodology using process modelling. International Journal of Research in Engineering Design 18, 37–48 (2007)
3. Sanin, C., Szczerbicki, E.: Set of Experience: A Knowledge Structure for Formal Decision Events. Foundations of Control and Management Sciences 3, 95–113 (2005)
4. Sung, R.C.W., Ritchie, J.M., Lim, T., Kosmadoudi, Z.: Automated generation of engineering rationale, knowledge and intent representations during the product life cycle. Virtual Reality 16, 69–85 (2012)
5. Sanin, C., Szczerbicki, E.: Towards the Construction of Decisional DNA: A Set of Experience Knowledge Structure Java Class within an Ontology System. Cybernetics and Systems 38, 859–878 (2007)
6. Sanin, C., Toro, C., Haoxi, Z., Sanchez, E., Szczerbicki, E., Carrasco, E., Peng, W., Mancilla-Amaya, L.: Decisional DNA: A multi-technology shareable knowledge structure for decisional experience. Neurocomputing 88, 42–53 (2012)
7. Sanin, C., Szczerbicki, E.: Extending Set of Experience Knowledge Structure into a Transportable Language extensible Markup Language. Cybernetics and Systems 37, 97–117 (2006)
8. Sanín, C.: Smart Knowledge Management System, Thesis of Doctor of Philosophy Degree From The University of Newcastle Department Of Mechanical Engineering, Newcastle, Australia (2007)
9. Shafiq, S.I., Sanin, C., Szczerbicki, E., Toro, C.: Using Decisional DNA to Enhance Industrial and Manufacturing Design: Conceptual Approach. In: 34th International Conference on Information Systems Arhitecture and Technology, Szklarska Poreba, Poland, pp. 23–32. Wroclaw University of Technology (2013)
10. Davis, R., Shrobe, H., Szolovits, P.: What is a Knowledge Representation? AI Magazine 14, 17–33 (1993)
11. Toro, C., Sanín, C., Szczerbicki, E., Posada, J.: Reflexive Ontologies: Enhancing Ontologies With Self-Contained Queries. Cybernetics and Systems 39, 171–189 (2008)