Postprint of: Sanin C., Waris M., Toro C., Szczerbicki E., Manufacturing collective intelligence by the means of Decisional DNA and virtual engineering objects, process and factory, JOURNAL OF INTELLIGENT & FUZZY SYSTEMS, Vol. 32, Iss. 2 (2017), pp.1585-1599, DOI: 10.3233/JIFS-169152

Manufacturing collective intelligence by the means of Decisional DNA and Virtual Engineering Objects, Process and Factory

Cesar Sanin[a] , Imran Shafiq[a], Mohammad Maqbool Waris[a], Carlos Toro[b], and Edward Szczerbicki[c]
[a]*School of Engineering, The University of Newcastle, University Drive, Callaghan, NSW 2308*
[b]*Vicomtech, Paseo Mikeletegi 57, 20009 Donostia/San Sebastián, Spain*
[c]*Faculty of Management and Economics, The Technical University of Gdansk, Gdansk, Poland*

Abstract. Engineering collective intelligence is paramount in current industrial times. This research proposes and presents case studies for collective knowledge structures required in the industry field. Knowledge structures such as Set of Experience and Decisional DNA are extended into more advanced knowledge structures for manufacturing processes. These structures are called Virtual Engineering Object, Virtual Engineering Process and Virtual Engineering Factory. All knowledge structures are implemented and tested in two industrial manufacturing cases of collective knowledge, plus one more case of manufacturing innovation where the case study results proved them as practical standards for engineering collective intelligence.

Keywords: Decisional DNA, Set of Experience Knowledge Structure, Virtual Engineering Object, Virtual Engineering Process, Virtual Engineering Factory

1. Introduction

Collective Intelligence (CI) has been defined in several areas of the human action, from humanistic to technocratic points of view. In a general form, we take the definition given by [10] as “Group(s) of individuals doing things collectively that seem intelligent; Groups addressing new or trying situations; Groups applying knowledge to adapt to a changing environment”. Moreover, CI has been proposed in several fields such as arts, business, health, finance, IT, etc. [19, 21]. However, CI, when implemented, has been within a single organization or group of them that work together in a closed form; therefore, limiting CI by operating in an specific domain and/or following a vendor specification. Moreover, most of these implementations are leveraged by internet technologies and the support of the semantic web, but, CI lacks of accepted standards and not even an organization that promotes techniques or frameworks for collective intelligence.

The most advanced standardization examples that support CI are provided by the World Wide Web Consortium (W3C). Different technologies and metalanguages such as XML or HTML have overcome the vendor barrier to turned into accepted standards implemented in numerous applications, softwares, and industry fields. However, little can be said about particular collective intelligence technologies which are vendor/domain/company independent, and consequently, there are limitation on creating a consistent and real collective intelligence. Hence, we identified that CI requires the construction of established standards to achieve a greater intelligence that can be created in a collective way as it is expected.

In the engineering field, several approaches to a collective intelligence have been proposed an implemented; mostly in the areas of product design, product marketing and industrialization. They can be catalogued within the field of Cyberphysical systems (CPS), and in a more elaborated way, Europe has proposed Factory 4.0. Under these circumstances, still remains the idea that standards must be created and if Europe, with its Factory 4.0 proposal, is going to lead, it will require structures/models to collect engineering knowledge in a collective form. This paper respond to such need.

This paper presents a group of knowledge structures able to collect and share collective intelligence/knowledge related to engineering. Such technologies involve Set of Experience Knowledge Structure and Decisional DNA as the basis for more advanced structures: Virtual Engineering Object (VEO), Virtual Engineering Process (VEP) and Virtual Engineering Factory (VEF).

The paper is structure as follows. In Section 2, we will present the knowledge structures that will support engineering CI, Sections 3, 4 and 5 introduce case studies and applications of the mentioned structures, and in Section 6, we will conclude and establish some future work.

2. Knowledge Structures

Industrial manufacturing is a highly complex, creative, and knowledge intensive process that involves collaborative information exchange from various sources with continuous production conditions changes. Thus, for representing such dynamic environment, a flexible knowledge structure capable of handling fluctuating parameters at each level is required. The knowledge representation structure facilitating experience based intelligence as the technological base for this work are Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA). SOEKS-DDNA [22, 23] is a unique and single structure for collecting, storing, improving, and reusing experience of intelligent decision-making. SOEKS is composed of variables, functions, constraints and rules associated in a DNA shape permitting the development of the Decisional DNA of an organization which embodies its collective intelligence. Variables normally implicate representing knowledge using an attribute-value language (i.e. by a vector of variables and values). Variables are the centre root of the structure and starting point for the SOEKS. Functions represent relationships between a set of input variables and a dependent variable; moreover, functions can be used for optimal state reasoning. 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 condition and consequence linked by the logical statements IF-THEN-ELSE. They are conditional relationships that control the universe of variables.

Decisional DNA is a metaphor related to natural DNA and the way it transfers genetic information and knowledge among individuals through time. The DDNA consists of stored experienced decision events (i.e. experiential knowledge) that can be grouped according to areas of decision or categories. In other words, each SOE (short form for SOEKS) built after a formal decision event can be categorized and acts similarly to a gene in DNA. A gene guides hereditary responses in living organisms, as a SOE directs responses of certain areas of the organization. Furthermore, assembled genes create chromosomes and human DNA, as groups of categorized SOE create decisional chromosomes and Decisional DNA.

Furthermore, dynamic structures of SOEKS and DDNA provide flexibility to the structures of VEO, VEP and VEF. Thus, the broad aim of this research is to develop engineering fingerprint or engineering DNA of a company which is built through engineering collective intelligence.

This work aims at demonstrating that structures such as SOEKS and DDNA are required to achieve higher levels of collective intelligence and therefore respond to the above presented need. Structures that collect and replicate knowledge and experience of a manufacturing factory and represent it virtually. As shown in Figure 1, the physical manufacturing scenario can be divided into three levels: resources, processes and factory. In the manufacturing domain, a factory performs various manufacturing processes, and a process in turn uses different resources. For the comprehensive knowledge representation (KR) of a manufacturing system, we divided it into three levels; the first is the resource/object level or VEO, the second is the process level or VEP and the third is the factory/system level or VEF. VEO, VEP and VEF, all store collective experience captured in the forms of SOEKS-DDNA, captured during the normal operation of the system. Thus, all the three virtual KR are built on SOEKS-DDNA and have mechanisms to store and reuse collective experience related to objects, processes and factory. Different KR models of these levels have been developed both separately and in conjunction with each other. KR of engineering objects, processes and system collected in an intelligent manner will help optimise assets, machines and the whole system, respectively. Critical, effective and creative decisions can be made based on these intelligent virtual manufacturing levels.

A VEO is a representation at the individual object/resource/artefact level, and represents all information at the resource level such as in a machine, the machining parameters, tolerances and surface conditions; or a tool, tool parameters, and functionality. The VEP deals with information at the process or shop-floor level, such as operation sequences, process parameters, time and cost. The VEF stores the experience and formal decisions related to various aspects at the system level, such as material handling, storage, quality control and transportation. Besides representing knowledge at the factory level, the VEF also contains VEOs and VEPs. The combination of VEOs, VEPs and the VEF constitutes the virtual industrial manufacturing platform for collecting engineering collective intelligence.

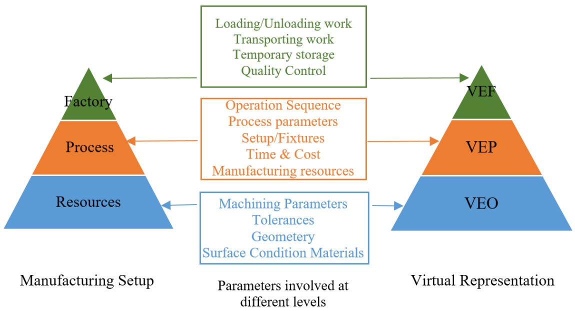

Fig 1 Correlation of physical and virtual manufacturing world

Complete concepts of VEO, VEP and VEF are already developed, implemented and tested [19-21]. For completeness, the next section presents a brief description of them.

2.1. Virtual Engineering Object (VEO)

A VEO is a KR of an engineering artefact and it has three main features: (i) the embedding of the decisional model expressed by the SOE (ii) a geometric representation, and (iii) the necessary means to relate this virtualization to the physical object being represented [9, 17, 19, 20].

A Virtual Engineering Object is a living representation of an object capable of capturing, adding, storing, improving, sharing, and reusing knowledge through experience in a way similar to a human expert. A VEO can encapsulate knowledge and experience of every important feature related to an engineering object. This can be achieved by gathering information from six different aspects (manufacturing chromosomes) of an object namely VEO-Characteristics, VEO-Functionality, VEO-Requirements, VEO-Connections, VEO-Present State and VEO-Experience.

Virtual Engineering Object is developed on the cradle-to-grave approach, which means that the contextual information and decision making regarding an engineering object right from its inception until its useful life is stored or linked in it. The SOEKS-DDNA technique allows VEO not to adhere to any rigid arrangement of parameters which provides dynamicity and flexibility to the structure; such a feature enables VEO to represent complex and discrete engineering objects.

2.2. Virtual Engineering Process (VEP)

A VEP is a KR of a manufacturing process or process planning of an artefact that gathers and stores all shop-floor-level knowledge regarding the operations required, their proper sequence and the resources (VEOs) needed to manufacture it. The VEP selects the necessary manufacturing operations and determines their sequences, as well as selecting the manufacturing resources needed to transform a design model into a physical component. In addition to this, information of all the VEOs of the resources associated with the process is also linked into VEP. Therefore, to encapsulate knowledge of the above mentioned areas, the VEP is designed with the following three main modules:

1. VEP-Operations: it stores all of the information related to the operations that are required to manufacture an engineering component. This includes knowledge in the form of SOEKSs related to operational processes and scheduling as well as functional dependencies between operations.
2. VEP-Resources: it stores information based on past resources' experience used to manufacture a component mentioned in the VEP-Operations module. Moreover, the VEO information categorised under VEO-Characteristics, VEO-Requirements, VEO-Functionality, VEO-Present State, VEO-Connections and VEO-Experience is also linked in this module.
3. VEP-Experience: it stores links to SOEKSs of VEOs along with VEPs containing past formal decisions relating to manufacture engineering components. Thus, the information in this module represents links to SOEKSs based on past experience of that particular machine perform-

ing a given operation along with operational and routing parameters.

As well as in the VEO, the SOEKS-DDNA technique allows VEP not to adhere to any rigid arrangement of parameters providing dynamicity and flexibility to the structure.

2.3. Virtual Engineering Factory (VEF)

VEF is KR of a complete manufacturing process and it is an extension of the VEO-VEP concept to a factory level. A manufacturing factory is a collection of integrated equipment and human resources whose function is to perform one or more processing and/or assembly operations starting with a raw material, part, or set of parts. The main components of a manufacturing system can be broadly classified as:

– production machines and tools,
– material handling and work-positioning devices,
– computer systems and
– human resources required to keep the system running.

The architecture of VEF is conceived based on the components and their functionality at a factory level. Hence, a VEF comprises five elements, each linked to the associated VEPs and VEOs representing all of the collective knowledge and experience related to a manufacturing factory. The arrangement of these six VEF elements, along with their VEOs and VEPs, is shown in Figure 2. The VEF elements are as follows:

4. VEF-Loading/Unloading: it stores information related to loading and unloading work units at each station along with the positioning of work units at each station.
5. VEF-Transportation: This module stores knowledge associated to transporting work units between stations in a multistation system. Work units either flow through the same sequence of workstations or are moved through a variety of different station sequences.
6. VEF-Storage: This module stores all knowledge related to the permanent and temporary storage of tools, objects, raw materials and work during the manufacturing process.
7. VEF-Quality Control: This module contains the quality control strategy adopted, its implementation method and outcome.
8. VEF-Experience: it stores the entire history of formal decision events made at the factory level, along with links to the VEPs and VEOs related to those decisions. In other words, all past collective experience is captured in this module.

Each factory level experience (i.e. VEF-SOEK) is associated with a component (VEP-SOEKS) to be manufactured and that component in turn needs resources/objects (VEO-SOEKS) for its manufacturing. This idea is shown in Figure 2; VEF-DDNA is created by collecting, connecting, and linking VEF-SOEKS, VEP-SOEKS and VEO-SOEKS. Therefore, a VEF can be defined as collective experience-based manufacturing DNA or manufacturing footprints bearing traces of all decisions made at the product, process and factory levels.

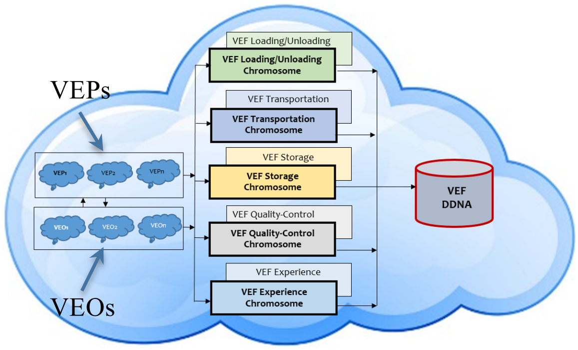

Fig 2 VEF architecture linking VEOs and VEPs

2.4. Salient Features of proposed virtual engineering object, process and factory

As mentioned in the previous sections VEO, VEP and VEF are based on the knowledge representation technique of SOEKS and Decisional DNA. This technique is capable of creating engineering manufacturing DNA (collective computational manufacturing intelligence) as it has manufacturing nucleotides (variables, function, constraints, rules), manufacturing genes (collection of SOEKS), manufacturing chromosomes (collections of manufacturing genes namely VEO-Characteristics, VEO-Requirement, VEO-Functionality, VEO-Present State, VEO-Connections, VEO-Experience, VEP-Resources, VEP-Operations, VEP-Experience, VEF-Loading/Unloading, VEF-Transportation, VEF-Storage, VEF-Quality Control, VEF-Experience). Experimental case-studies [19-21] have proven that a DDNA-based VEO-VEP-VEF knowledge system has the following features:

– a versatile and dynamic knowledge structure, which provides the flexibility necessary to change according to the situation;

- the ability to store day-to-day explicit collective experiences in a single structure, which will continuously evolve;
- transportable, adaptable and shareable knowledge;
- prediction and decision-making abilities based on collected past experience, and
- the ability to achieve decisional trust by having the right quality and quantity of knowledge at the right time.

As shown in Figure 2, the VEO-VEP-VEF system is also envisaged on a cloud computing platform to facilitate the operation of engineering collective intelligence relating to multifaceted interrelationships.

3. Case-Study: Creating Engineering DNA

The following case study presents a model for collecting collective engineering DNA by the means of the proposed knowledge structures presented in Section 2: SOEKS-DDNA, VEO, VEP, and VEF. This system creates manufacturing DNA with retaining, predicting, and decision making capabilities based on the collected past collective experience.

In this case study, the VEF concept is demonstrated and implemented in a manufacturing system to produce an engineering component. This case study extends the previous VEP and VEO case studies [19,20], which were based on manufacturing a simple combustion chamber in a conventional machining setup. The basic operations required to manufacture this combustion chamber are taper turning, turning, and drilling; such information is stored in a VEP, which is shown as a work-in-process assignment ('WIPA') in Figure 3. The manufacturing setup in this case study has two different lathe and drilling machines each. Factory-level information about work-piece loading/unloading, quality control, transportation, storage, and previous experience are stored in the VEF and within it as SOEKS.

First, VEOs of the machines required to produce the engineering component are developed. Then, the VEPs to produce an engineering component are built based on the case-specific collective experiences of that manufacturing unit. Finally, the VEF having all of the factory-level knowledge along with links to the VEPs and VEOs is constructed. The VEOs along with collective experience of the engineering processes (VEPs) form the experience repository of a manufacturing unit. Files storing formal decisions related to VEF-Loading/Unloading, VEF-Transportation, VEF-Storage, VEF-Quality Control, and VEF-Experience were built for the component to be made, that is, a combustion chamber. CVS, TXT, XML or OWL formats can be used for managing and storing data; we proposed SOEKS-XML format as data transfer among applications [5]. Since VEO-DNA and VEP-DNA were already developed and presented in previous case studies, the next step was to develop VEF-DNA and link it with VEO-DNA and VEP-DNA to create a complete Manufacturing DNA.

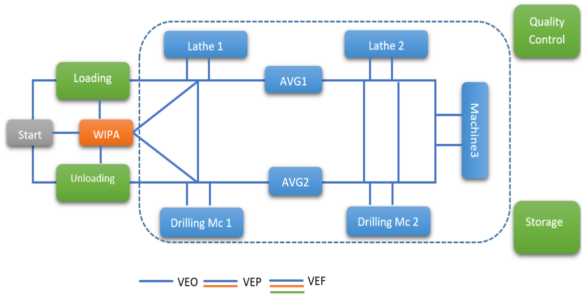

Fig 3 Case study framework involving the elements of VEF

Our platform was written in Java, and uses SOEKS-XML to transfer information; however, it has the capacity to read from any other file formats mentioned above. The platform parsers the collective experiences into the knowledge structures in Java. Each file representing a category, and a collection of SOEKS of the same category forms a chromosome of either a VEO, VEP or VEF. Even though our platform was developed in Java, all the Virtual Engineering structures, SOEKS and DDNA can be developed in any other programming language (a Python library has also been developed). Then, the collection of all chromosomes forms a Decisional DNA of a VEF, i.e. VEF-DNA. Once the VEF-DNA is constructed, DDNA has feature capabilities of being queried [16, 21].

DDNA includes similarity metrics for its elements; therefore, given a set of parameters and a repository of DDNA, the platform is able to calculate distances among the parameters and collected experiences and find the most suitable experiences for a defined query[3]. However, other similarity metrics can be applied such as [15]. In those terms, the platform allows generating a query by a GUI, which is then programmatically converted into a query SOE (*querySOE)*. Depending whether it is related to the object, process or factory level, the platform calculates the similarity of the *querySOE* with each

SOEKS stored in the VEF-DNA. Finally, the calculated similarities are sorted and the five most similar SOEKSs are presented to the user.

Collective manufacturing experience is produced on day-to-day basis and stored in a repository of experiences; the one that is going to be queried. When there is a need to produce a new component, the VEF repository is scanned for similar components (a combustion chamber in this case). The VEF reads the experience of that component in its repository and returns information relating to the previous most similar collective manufacturing experience stored. Next, the query relating to specific factory-level details required for the component is specified. For this query, the VEF returns VEP-SOEKS for process/process planning and VEOs for each operation, along with the SOE that best suits the queried resources details. The most similar VEO-SOEKSs are gathered and combined with the most similar VEP-SOEKSs. This information combined with the most similar VEF-SOEKSs forms the solution to the query.

A simple user friendly GUI (see Figure 7) is designed to build queries; user specifies information regarding the product, its variables and variable values. Information is extracted from the VEF-DNA for most similar VEF-SOEKS and further details of VEP-SOEKS and VEO-SOEKS corresponding of that experience can be viewed through GUI. In the results section of the GUI, the user can see the similarity indexes along with codes of the most similar VEF, VEP and VEO SOEKS according to the to the query. The user can also view the complete VEF-SOE, as well as the VEP-SOEs and the VEO-SOEs associated to such VEF.

Principles of this case-study can be followed to effectively scale-up the knowledge representation of complex industrial set-ups. Thus, flexible and dynamic structures of SOEK-DDNA, VEO, VEP and VEF are capable of representing and gather experience in a collective approach from any manufacturing environment.

3.1. Results and discussion

The implementation of this study was carried on a DELL laptop with the Windows 7 Enterprise operating system, Intel (R) Core (TM) i5-3210M CPU @ 2.50 GHz processor and 8 GB of RAM. The significance of the VEO-VEP-VEF models used in the case study are analysed by doing the following:

- assessing the time taken to create SOEKSs from the VEO, VEP and VEF files
- obtaining the most similar SOE to a query and calculating query execution time
- analysing changes in similarity patterns due to varying query input parameters.

3.1.1. Time taken to create SOEKSs from the VE object, process and factory files

The present VEF study comprises SOEKSs from VEF-Loading/Unloading, VEF-Transportation, VEF-Storage, VEF-Quality Control and VEF-Experience having a minimum of 47 variables and 10 constraints. In addition, VEP-DNA comprises SOEKSs from VEP-Resources, VEP-Operations and VEP-Experience, having 20 variables and 12 constraints. Moreover, the VEO-DNA comprises SOEKSs from VEO-Characteristics, VEO-Functionality, VEO-Requirements, VEO-Present State, VEO-Connections and VEO-Experience, having 53 variables, 3 functions and 28 constraints. For testing purposes, we queried VEO-Drilling Machine from a repository of 2256 SOEKSs, VEO-Lathe Machine from 1920 SOEKSs, VEP from 320 SOEKSs and VEF from 26 SOEKSs.

The parsing process of the VEF, VEP and VEO decisional chromosomes were executed, producing a parsing time of 664.0 ms for VEO_Drilling, 504.0 ms for VEO_Lathe, 161.0 ms for the VEP and 10 ms for the VEF (see Figure 4). This is considered an excellent time taking into account the fact that these SOEs are very complex due to the number of variables, functions and constraints involved, adding up to a total of 141 key features per formal decision event. The model is fairly effective as far as the time taken to parse VEO, VEP and VEF is concerned.

3.1.2. Time taken to respond to a query

Different queries were designed as a way to test the platform, some with varying the number of variables or some with the same number of variables as it can be seen in [21]. For example, a query VEF similarity is calculated for ‘Combustion Chamber’ where MST = 528 min, WIP = 146 mins, Machining Time = 109 mins and Idle Time = 273.

Figure 5 illustrates the time results of this query. VEF-DNA returns the five most similar SOEKSs having similarities 0.43934, 0.45154, 0.45384, 0.45537 and 0.45654, respectively. The time taken to execute this query is 6.766 ns which is fairly short.

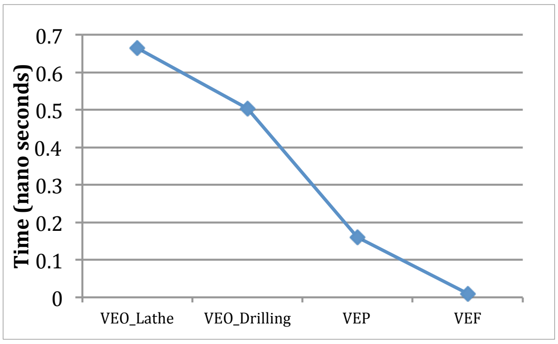

Fig 4 Time taken to parse VEO, VEP and VEF

To determine the performance and robustness of our model, a set of queries having a decreasing number of variables while all other parameters were the same was executed. As illustrated in Figure 5, as the number of query variables decreases the similarity value increases, which validates the efficiency of the model.

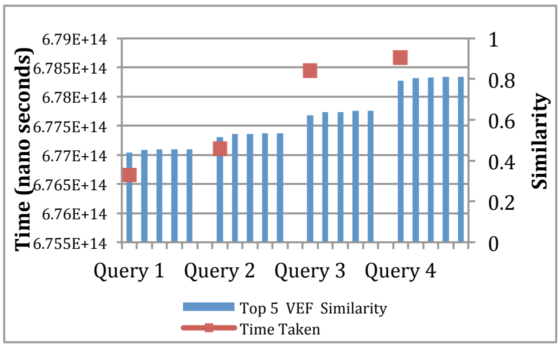

Fig 5 Calculating similarity for queries and corresponding response time for query execution

3.1.3. Analysing the change in similarity pattern with varying query input parameters

The behaviour of the model was also analysed by executing queries having input variables varied. As presented above, a similar pattern of the five most similar SOEKSs for each query was calculated as depicted in Figure 6. The similarity calculation was found to be quite accurate and the execution time of this set of queries was fairly short as well.

3.2. Case Study Conclusion

The main contribution of this work is to demonstrate and implement a collective knowledge based virtual engineering environment. The Manufacturing DNA which is the representation of manufacturing process collective computational intelligence is created by capturing experience of engineering objects, engineering processes, and factory by the means of Virtual Engineering Object, Process and Factory. Set of Experience Knowledge Structure and Decisional DNA were applied as the knowledge representation structure for gathering the experience. Further, VEF-VEP-VEO were used as a tool for decision making processes that can enhance different manufacturing systems with predicting capabilities and facilitate knowledge engineering processes. The platform copes with self-organizing production and control strategies; being this a significant example of linking product lifecycle management, industrial automation, and semantic technologies.

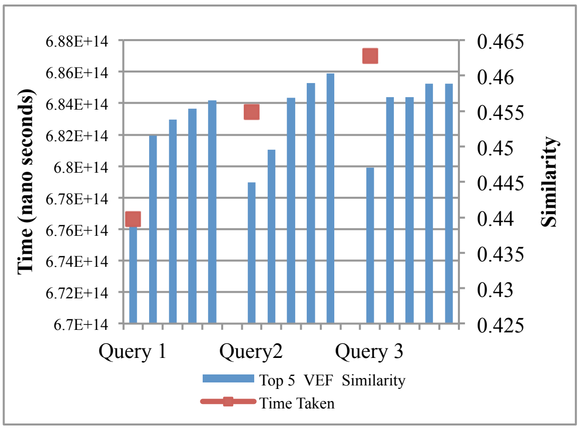

Fig 6 Similarity calculation for varying variable values

4. Case Study: CI and Factory 4.0

This case study contributes to the intelligent factory concept proposing a model that entails rapid transfer of new collective knowledge into industrial processes and products. In our work, we focus on the knowledge based conceptual model, architecture and key elements needed for the support of Industrie 4.0. The proposed framework (see Figure 7) follows four stages: (i) Data Collection and Communication platform (ii) Data preparation or Basic Data Analysis (iii) Semantic Analysis and (iv) Real-time visualization.

The architecture for intelligent factory can serve to create horizontal value networks at a strategic level, provide end-to-end integration across the entire value chain of the business process level and enable vertically integrated and networked design of manufacturing systems.

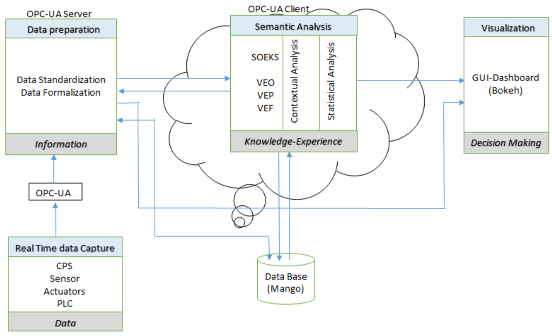

Fig. 7. Architecture for the intelligent factory

4.1. Data Collection and Communication platform

In all industrial applications, data/information plays a very important role. Standardization and languages for standardization of communications in a machine-to-machine context like OPC (Object Linking and Embedding - OLE for Process Control) and more recently OPC-UA (Unified Architecture), using a unified architecture not dependent on OS, play a very important role. The benefits of using the aforementioned approaches are quite evident in the sense that an abstraction layer from the manufacturer's programming interface and proprietary languages in the PLC's, sensors and actuators are simplified acting as an inter-language for communication. The data collection in the OPC-UA approach is representational state transfer (REST) oriented, client-server implemented and provides a mechanism to subscribe to data changes in an asynchronous manner. Data collection then can be serialized and the gathered data stored in different databases that will be implemented as clients consuming the data. Analysis in terms of data changes and event changes are also benefited, as the synchronous/asynchronous need of a given application is a feature that will become easy to handle and maintain.

4.2. Data preparation

Once the data is collectively collected, it is necessary to prepare it for its exploitation. First of all, there is a necessity of some filtering, as not all the raw data is useful. The outliers and any other fragment of data that is considered noise are eliminated here. Then, the data is standardised. Here, we use AutomationML [1], which is an open standard based on XML for the storage and exchange of plant engineering information. AutomationML describes real plant components as objects encapsulating different aspects. An object can consist out of other sub-objects, and can itself be part of a bigger composition.

Finally, the data is aligned and synchronized. Since sensors do not normally have a real-time clock, as computers have, it is the responsibility of the device that is capturing to set a time reference. Moreover, each sensor has its own sample time that depends on the dynamic of the system that is monitoring. So, all the captured data is organised and rearranged in this module to send it to the cloud in a synchronous pace.

4.3. Semantic Analysis

The semantic enhanced intelligent factory model agglutinates the entire reasoning process. The semantization process starts with an IN/OUT module that synchronizes the information to be enriched with the communication layer messages/serialized-responses maintained between the server and the client.

As presented above, the knowledge representation technique of Set of experience knowledge structure (SOEKS)-Decisional DNA (DDNA) is used for developing VEO and VEP models and it is the sematic reasoner adopted.

4.4. Real-Time Visualization

Visual techniques are increasingly being used for exploratory analysis and to quickly identify patterns in industrial processes. As Visual Analytics are especially suited for complex real world problems with large amounts of data, they fit perfectly in this field. The proposed framework contains a Visual Analytics module that offers a graphical output to the semantically enhanced collective experience stored in the architecture.

In our approach, we implemented a flexible dashboard system instead of a single universal visualization. The diversity of problems that can appear in a manufacturing environment is too high to create a unique type of visualization. It is better to build an interactive tool that can create customized visualizations. The user can visualize in real time different variables, graphs and charts, and compose its own visualization configuration.

The visualization module is based on Bokeh, which is a Python interactive visualization library that targets modern web browsers. Its goal is to provide elegant, concise construction of novel graphics in the style of D3.js (another library for data visualization), but also deliver this capability with high-performance interactivity over very large or streaming datasets [2].

4.5. Methodology

Experience is collected from four sensors, measuring different parameters: temperature, pressure, spindle-speed and metal removal rate. These are key operation parameters as they affect surface finishing, machining time and other output indicators; thus, they must be monitored and analysed. Total number and type of these sensors may vary according to different machining conditions. Some of the salient features of the case study implementation are:

- Using CPS-like devices and OPC-UA to support collective experience captured coming from sensors and actuators recording specific activities of the machines.
- Standardising data representation by using AutomationML.
- Using SOEKS converting machine experience stored in database (Offline) as Set of Knowledge Experience Structure (SOEKS).
- Using SOEKS to create VEO and VEP according to their format.
- Plotting streaming data in the client using visualization API based on BOKEH.

4.6. Results

As illustrated in Figure 8, collective experience is continuously being pushed from machines. Storing streaming data is effective for the evaluation of machine performance and for its maintenance. Any significant change to the status of the monitored machine can be detected. The change can be defined as a dramatic variation (high and low) in machine health value, a maintenance action or a change in the working regime. During the life cycle of a machine, these streaming data will be accumulated and used to construct the time-machine history of the particular asset. This active time-machine record will be used for peer-to-peer comparison between assets. Once the asset is failed or replaced, its relative time-machine record will change status from active to historical and will be used as similarity identification and synthesis reference.

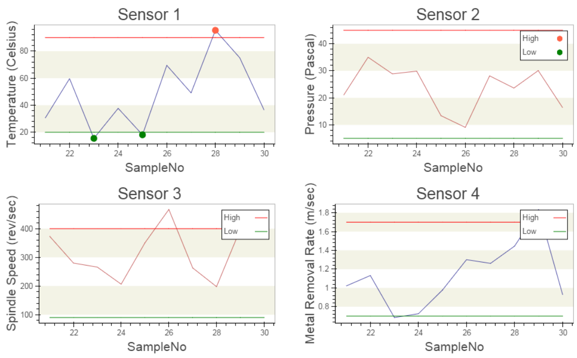

Fig. 8. Visualization of streaming data

Experience coming from four sensors is captured and arranged in the SOEKS format to represent formal decisions taken while operating the machine. To compare the current machine behaviour, similarity with each past SOEKS of the machine is calculated [3].

Figure 9 shows similarity index calculated for each SOEKS in the repository with the query SOEKS. The SOEKS marked with a red dot indicates the most similar SOE. Once the patterns are matched, future behaviour of the monitored system can be predicted more accurately.

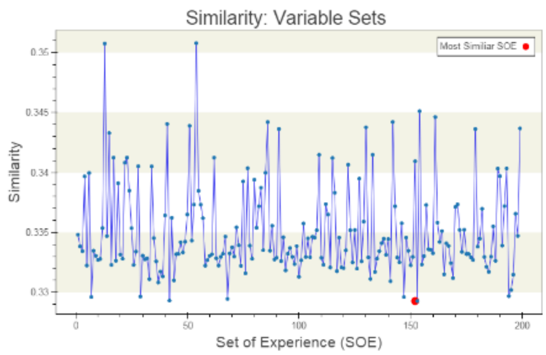

Fig. 9. Similarity identification for each SOEKS

For each SOE, functions calculate machine health index and tool life. Figure 10 illustrates corresponding machine health index and tool life for each SOEKS. Predicting remaining useful life of assets helps to maintain just-in-time maintenance strategy in the manufacturing plant. In addition, life prediction along with historical time machine records can be used to improve the asset utilization efficiency based on its current health status. Historical utilization patterns of similar asset at various health stages provide required information to simulate possible future utilization scenarios and their outcome for the target asset. Among those scenarios, the most efficient and yet productive utilization pattern can be implemented for the target asset.

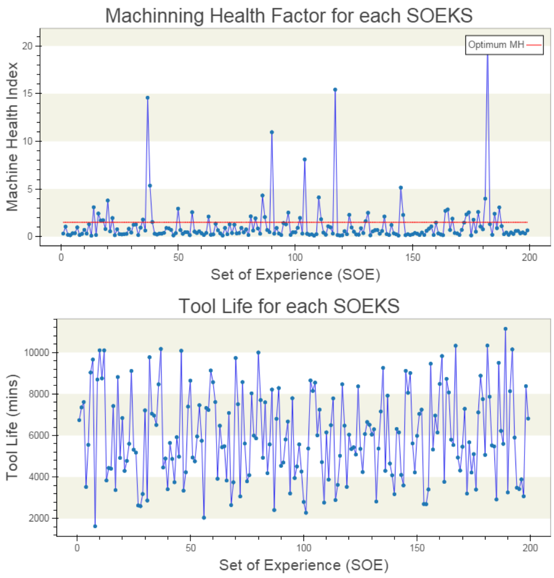

Fig. 10. SOEKS Functions evaluation for each formal decision

4.7. Case Study Conclusion

In this case study, a framework for building a collective intelligent factory is presented. The collective intelligent factory concept holds a huge potential as it enables dynamic manufacturing business last-minute changes to production and delivers the ability to respond flexibly to disruptions and failures. End-to-end transparency is provided over the manufacturing process, facilitating optimised decision-making. The presented case study has prospects to facilitate building of bigger environments of Industry 4.0.

5. Smart Engineering Innovation

5.1. Genetic structure of a product

First the artefact, or the product, is structured in terms of a hierarchy of nested parts [11]. The product is divided into a number of systems performing some specific function of that product, which can be represented as subsystem level 1. Similarly, subsystem level 1 can be subdivided into further subsystems, represented as subsystem level 2 which are subassemblies associated with some sub-function that collectively perform the function at subsystem level 1. This nesting continues until the subsystem level reaches the component level. The number of subsystems is different for any particular system performing a particular function of the product and can go up to level 10 or more [14] to reach the component level. Moreover, the level of each subsystem in the same product does not need to be the same, as it depends upon the complexity of the system. Consider the automobile car as a product, it can be divided into subsystems level 1 like car body, engine, fuel system, suspension system, braking system, electrical system and so on. At subsystem level 2, it can be a piston and so on until it reaches the component level which can be a simple pressure ring under the engine system or a self-locking nut in car body system.

There are inter-relationships among the systems, subsystems and components [18]. These relationships can be at the same level in the same system like piston and cylinder in the engine system, or it can be between subsystems at the same or different level under different systems.

5.2. Decisional DNA and Product Innovation

Organizations involved in manufacturing products need to find out new ideas and innovate continuously to survive and prosper [6]. Innovation is defined as the process of making changes to something established by introducing something new that add value to users and contributes to the knowledge store of the organization [13]. A systematic and proper approach in product innovation can increase the life of the product.

Based on innovative objectives, organizations can find out which features or functions of the product need to be upgraded, which ones may be excluded and which new features or functions may be added to the product. These features and functions are attributed to some systems of the product. Innovative changes in the product can be performed by modifying one or more systems of the product. These modifications or changes can be at system, subsystem or component level. Accordingly, the required changes can be incorporated into the product to complete the innovation process.

An architecture for product innovation DDNA is shown in Figure 11. It is the knowledge representation of a product which is capable of capturing, storing, adding, improving, sharing as well as reusing knowledge in decision making in a way similar to an innovator or entrepreneur. The product innovation

DDNA contains knowledge and experience of each important feature of a product. This information is stored in eight different modules of a product: Characteristics, Functionality, Requirements, Connections, Process, Systems, Usability, and Cost. The first five modules come under the decisional DNA of virtual engineering objects / virtual engineering process (VEO / VEP). The information and experiential knowledge from VEO/VEP can be easily shared and used for innovation process.

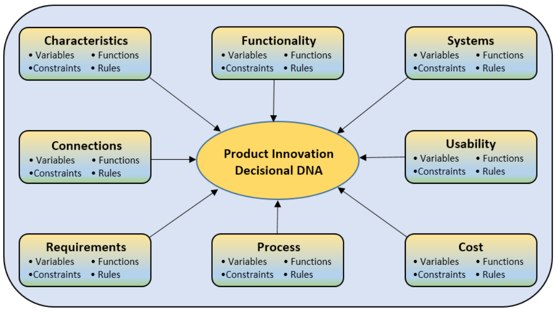

Fig. 11. Architecture of a Product innovation DDNA

Characteristics represents knowledge about the physical system, subsystems and components of the product as well as some operation attributes.

Functionality represents knowledge about the basic object action (object collectively represents system, subsystem or a component of the product) and its operational principles.

Connections represents the knowledge about the relations between the VEOs in conjunction with the manufacturing scenario.

Requirements represents knowledge about the VEO limitations for its precise working.

Process represents the knowledge about the manufacturing process/process planning of the artefact, VEP.

All the knowledge represented by the above mentioned five modules can be extracted from the VEO/VEP DDNA. Further, three more modules are introduced which are described as follows;

Systems represents not only the knowledge about the relationships between various systems, subsystems and components like their hierarchy and dependability so as to represent a complete product but also stores the past history of every system, subsystem and component that were used for performing the same function. It also stores the possible alternative systems, subsystems or components that have the potential of replacing the current one. This module is continuously updated with the alternative systems used in advanced products as well as the new technological systems, inventions and advanced materials. This is the most important module for innovation process.

Usability represents the knowledge about the use of a particular system, subsystem or component of the product in other products. This will help in assessing its performance in other products. Information like which products have stopped using this system or component and in which products it has been introduced recently and its effect on the performance, popularity, sales or price of the product.

Cost represents knowledge about total cost of all systems, subsystems and components. It will help in comparing and selecting the optimum manufacturing process on cost basis.

The query based on innovative objectives is fed into the SOEKS/DDNA system. This query is converted to a SOEKS containing a unique combination of variables, functions, constraints and rules. The system will look for the most similar SOEKS for comparison and based on the similar experiences will provide proposed solutions. For example, the innovative objectives suggests possible changes in five functions or sub-functions. The system will relate these functions and sub-functions with some systems and subsystems of the product. Comparing the experiences from the past having some common innovative objectives, the system will provide the set number of possible solutions. Based on the solutions of the past SOEKS, proposed solutions are obtained suggesting possible changes in some subsystems or components of the product.

The system will then compare the alternatives available in the systems and usability module. The best solution is chosen and stored in the decisional DNA of the product innovation as a SOEKS that can be used for solving innovative problem in future. In this way the system also gains some experiential knowledge and, with time, it will behave as an expert innovator/entrepreneur having knowledge equivalent to a group of experts, capable of taking quick and smart decisions.

6. Conclusions

This paper presents a group of knowledge structures able to collect and share collective intelligence/knowledge related to engineering. Such technologies involve Set of Experience Knowledge Structure and Decisional DNA as the basis for more advanced structures: Virtual Engineering Object (VEO), Virtual Engineering Process (VEP) and Virtual Engineering Factory (VEF). Through out the presented case studies, the afore mentioned structures are presented as technologies to manage any collective knowledge based virtual engineering environment. The idea is to make experience shareable and transferable among different manufacturing set-ups as required by the future generation of cyber-physical systems.

References

[1] AutomationML. (May 1, 2016) Available: https://www.automationml.org/.

[2] Bokeh. (April 30, 2016) Available: http://bokeh.pydata.org/en/latest/.

[3] C. Sanin and E. Szczerbicki. 'Developing Heterogeneous Similarity Metrics for Knowledge Administration'. Cybernetics and Systems 37(6), (2006): 553-565.

[4] C. Sanín and E. Szczerbicki. 'Experience-based Knowledge Representation: SOEKS'. Cybernetics and Systems 40(2), (2009): 99-122.

[5] C. Sanin and E. Szczerbicki. 'Extending Set of Experience Knowledge Structure into a Transportable eXtensible Markup'. Cybernetics and Systems 37(2-3), (2006): 97–117.

[6] Chen, K. Z., Feng, X. A. A gene-engineering-based design method for the innovation of manufactured products. J. of Engineering Design. 20. (2009): 175-193.

[7] Chen, K. Z., Feng, X. A., Chen, X. C. Reverse deduction of virtual chromosomes of manufactured products for their gene-engineering-based innovative design. Computer-Aided Design. 37. (2005): 1191-1203.

[8] Hernes M. Nguyen N.T. 'Deriving Consensus for Hierarchical Incomplete Ordered Partitions and Coverings'. Journal of Universal Computer Science 13(2). (2007): 317-328.

[9] Masood, T., R. Roy, A. Harrison, Y. Xu, S. Gregson and C. Reeve. 'Integrating Through-Life Engineering Service Knowledge with Product Design and Manufacture.' International Journal of Computer Integrated Manufacturing 28(1). (2014): 59–74.

[10] MIT Handbook of Collective Intelligence, (May 1, 2016) *What is collective Intelligence?,* Source: http://scripts.mit.edu/~cci/HCI/index.php?title=What_is_collective_intelligence%3F.

[11] Murmann, J. P., Frenken, K. Towards a systematic framework for research on dominant designs, technological innovations, and industrial change. Research Policy. 35. (2006): 925-952.

[12] Nguyen, N.T., Transactions on Computational Collective Intelligence III, Lecture Notes in Computer Science, 3 (2011): Springer.

[13] O'Sullivan, D., Dooley, L. Applying Innovation. Sage Publications (2008).

[14] Sanín, C. and E. Szczerbicki. 'Decisional DNA and the Smart Knowledge Management System: A Process of Transforming Information into Knowledge.' Techniques and Tools for the Design and Implementation of Enterprise Information Systems. A. Gunasekaran. New York: IGI Global, 2008, pp. 149–175.

[15] Sanín, C. and E. Szczerbicki. 'Towards the Construction of Decisional DNA: A Set of Experience Knowledge Structure JAVA Class within an Ontology System.' Cybernetics and Systems 38(8), (2007): 859–878.

[16] Sanin, C., C. Toro, Z. Haoxi, E. Sanchez, E. Szczerbicki, E. Carrasco, W. Peng and L. Mancilla-Amaya. 'Decisional DNA: A Multi-Technology Shareable Knowledge Structure for Decisional Experience.' Neurocomputing 88(0). (2012): 42–53.

[17] Sanín, C., C. Toro, Z. Haoxi, E. Sanchez, E. Szczerbicki, E. Carrasco, W. Peng and L. Mancilla-Amaya. 'Decisional DNA: A Multi-Technology Shareable Knowledge Structure for Decisional Experience.' Neurocomputing 88(0), (2012): 42–53.

[18] Sanín, C., L. Mancilla-Amaya, Z. Haoxi, and E. Szczerbicki. 'Decisional DNA: The concept and its implementation platforms,' Cybernetics and Systems 43, (2012): 67–80.

[19] Shafiq, S. I., C. Sanín, C. Toro and E. Szczerbicki. 'Virtual Engineering Object (VEO): Toward Experience-Based Design and Manufacturing for Industry 4.0.' Cybernetics and Systems 46(1–2), (2015): 35–50.

[20] Shafiq, S. I., C. Sanín, C. Toro and E. Szczerbicki. 'Virtual Engineering Process (VEP): A Knowledge Representation Approach for Building Bio-Inspired Distributed Manufacturing DNA.' International Journal of Production Research. (2016): 1–14.

[21] Shafiq, S. I., C. Sanín, C. Toro and E. Szczerbicki. Towards an experience based collective computational intelligence for manufacturing.Future Generation of Computer Systems, Elsevier. (2016): In print.

[22] Shafiq, S. I., C. Sanín, E. Szczerbicki and C. Toro. 'Decisional DNA Based Conceptual Framework for Smart Manufacturing.' Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part I. L. Borzemski, A. Grzech, J. Świątek and Z. Wilimowska, Springer: (2016): 79–88.

[23] Shafiq, S. I., C. Sanín, E. Szczerbicki and C. Toro. 'Virtual Engineering Factory: Creating Experience Base for Industry 4.0.' Cybernetics and Systems 47(1–2) (2016): 32–47