Application of NX Knowledge Fusion module for the Design Automation of an Automotive Painting Defects Inspection Tunnel

Bernal Z. Fabio A.[1], Rojo V. Alejandro[2] and Tornero M. Josep[3]

[1]Tecnológico de Monterrey, fabernal83@gmail.com
[2]Tecnológico de Monterrey, arojo@itesm.mx
[3]Universidad Politécnica de Valencia, jtornero@isa.upv.es

ABSTRACT

This article shows the application of Knowledge Based Engineering (KBE) methodology for the design automation of the structure of an automotive painting defects inspection tunnel. A previous work reached the design automation through the data transfering from Spreadsheets to Siemens NX CAD software using applications programmed with Visual Basic. With the application of KBE, it was possible to control the geometrical features of every part in the tunnel, to properly manage the component addition to the assembly and to reach a direct interaction between the designer and the CAD tunnel model in a new and more insightful manner. The presented work was developed through Knowledge Fusion (KF), a Siemens NX module which is based in Knowledge Based Engineering (KBE).

Keywords: KBE, knowledge fusion, CAD, design automation.
DOI: 10.3722/cadaps.2012.655-664

1 INTRODUCTION

1.1 Knowledge Based Engineering (KBE)

KBE is a type of knowledge-based system whose main focus is on Engineering Design and its down-stream activities. As engineering methodology represents a fusion of object-oriented programming, artificial intelligence techniques and technologies of computer-aided design, which together generate an automated way to introduce the design requirements, constrains and provides a description of the product. At the same time, knowledge or design intent that experts engineers have selected for your system is stored in the application KBE [1]. Contrary to wanting to replace the CAD modeling, KBE system serves as a supplement providing an environment for automating design activities and also the ways to build an abstraction in which the product knowledge is encoded and stored using an advanced programming language. This language is based on the paradigm of knowledge object-oriented representation (classes, objects, inheritance, polymorphism, etc.). Because KBE systems are closely linked to the functionality of CAD packages, such knowledge may be interpreted by the system to generate product information and 3D models, drawings and others [2].

The necessary knowledge for the design process of a product can be found in several forms such as spreadsheets, books, engineering formulas, drawings, documents or rules of thumb based on the opinion of the Engineer. A KBE system is capable of creating and referencing that knowledge and makes it available as an aid to the engineering process through software tools. Adding this knowledge to the CAD modeling of a product as a result gives us a virtual prototype, which has all the geometric characteristics or attributes of the product as well as non-geometric attributes such as materials, mass properties, stress and deflection characteristics, etc. Once the prototype is created, it can be used by designers to assess the success of the design and modify settings if necessary. The product model represents the engineering intent behind the geometric design. The information contained in this includes physical attributes such as geometry, material and functional restrictions [1].

With the implementation of KBE systems it is possible to reduce the time spent in the design stage of a part. Examples of this are the application developed to reduce time of silencers design made by the Department of Mechanical and Manufacturing Engineering at the University of Manitoba, Canada, where they can reduce design time from one day to few minutes [3] and the work done in the design of a jet engine to Purdue University in conjunction with Rolls-Royce where reduction obtained in time is 88% [4]. Reduction on time in the design stage occurs through automation of routine processes. These processes are tasks whose content can be rules that represent the range of application of the task. Rules can be divided into four categories [1]: a) The heuristics, which include experimental rules of thumb and best practices. These rules are of type if (true condition) then (recommended action) which means that if the rule is accomplished by changing the initial data, the input data for the model will be changed. b) The empirical, based on experimentally obtained curves c) The legislative constrains, which are covered by laws or standards of engineering and d) The physics, based on numerical models which are usually solved in spreadsheets.

KBE answers questions that traditional CAD systems to date have not been able to address, such as knowing the rationale behind the design, identify whether they have violated design constraint, determine the cost of the product, analyze the possibility of manufacturing a part, knowing whether a particular piece to meet expectations of performance or if a design is optimal or there are other alternatives [5],[6]. The gradual introduction of KBE technology from environments such as aerospace and automotive industry to other sectors of smaller volume has led to recognition of KBE systems as an emerging technology with the capacity to provide competitive advantages for the manufacturing industry [7].

1.2 Knowledge Fusion

Knowledge Fusion (KF) is the Siemens NX software module which is based on KBE. Through it you can develop applications that integrate KBE methodologies for developed NX models according to particular interests, modify geometries and models according to a programmed behavior through KF, and manage configurable and adaptive assemblies based on input parameters [8].

KF is usable from NX and allows rules creation for catch the design intentions that can be used in order to control the product design. The KF applications are designed with the intention to reduce the time and make a standardization of the design process, allowing the definition of a specific form to analyze or build and giving the opportunity to improve the work continuously.

KF has the control capability of the micro level of product development by using the obtained entries and handle it at macro level. This is possible by the combination of the generative modeling technology and a powerful system of knowledge bases.

Generative modeling creates geometries using a series of equations and defined geometric operations. Therefore, a sequence of process steps is used to define the geometry. Basic geometric forms and its operators can be combined between them or with other one with the intention to create a huge variety of complex forms and operations. Those can be used also as entries for forms and operations even more complex. On the KBE context, these steps can be considered as rules that are used to create geometries and also for carry out operations on the geometry.
The capability of generative modeling for manipulate geometric entities at micro level (lowest level of component design including primitive geometries) and the potential of the systems based on knowledge of implement automation on the products development using entries of macro level (user entries like final user requirements) can be combined to develop powerful tools that have the qualities of an ideal KBE application.

KF allows the creation of rules that carry out several operations. For example, the rules can be: Create geometries, carry out operations, handle expressions, connect to a database and generate designs based on this information, handle User Defined Features, create assemblies, generate structural or movement analysis on automatic way, generate personalized reports on different formats and upload it to internet and automatically generate CAM operations.

KF programmers can create their own classes (user classes) using the system classes and can bring to a high grade the product development automation. KF language has three characteristics that define its performance and are explained below [5]:

Declarative and Demand Driven: The Knowledge Fusion language is declarative, rather than procedural, which means that the rules can be written without regard for order. The Knowledge Fusion system determines the correct rule firing sequence driven by the dependencies between the rules. Like a spreadsheet, formulas do not have to be written in any particular order. If another formula is referenced (demanded), the system automatically finds and executes it.

Object Oriented: Like modern computing languages, such as C++, the Knowledge Fusion language is object oriented. There are classes and objects (instances), and there are multiple inheritances. A class is the abstraction of objects that have the same characteristics. For example, a block is a class of geometries that are commonly in cubic shape, and have length, width and height attributes. An object is a specific instance of a class. e.g., when you create a block object, you specify the values of its length, width and height to instantiate a specific instance of the block class. You can create as many block objects (instances) as you want, like block1, block2, block3, etc. They can have different dimensions but they all belong to the same block class. The process of creating object from a class is called instantiation.

Hierarchical: In much the same way engineers think about designing products, in Knowledge Fusion. Components are built into sub-assemblies and sub-assemblies are combined to become assemblies and here is consistent treatment at all levels.

1.3 Knowledge Based Engineering vs. Parametric Design

A parametric model is a computational representation of a design built with geometric entities that have Attributes (Values which a name is given. Its value could be taken from a rule expressed as a mathematic formula) that are fixed and others that can vary. Variable Attributes are also called Parameters and fixed Attributes are called constrains. This collection of geometries and relationships are stored in a database. Later, if the designer changes some or all dimensions, the full model fits the new parameters. This will reduce the amount of time spent by designers in the designs modifications, effectively helping make these designs faster to be manufactured [9].

The development of 2D and 3D CAD modeling tools has been focused to support the engineering design, however, its essence is based on geometry and have evolved into design automation (Parametric Design). In the conventional CAD software includes basic options that are close to this philosophy, which are just values that can be referenced by its name, so that they can relate, operate numerically and logically, etc. The potential of design rules that can be implemented using KBE methodology is far superior to working with Parametric Design. With the new KBE modules included in the CAD tools, it does have control of that is called non-geometric knowledge of product design. This analysis aspect of the design process has been commonly managed using procedural languages like FORTRAN, Pascal or C. Procedural languages are powerful tools for engineering analysis applications, but have no inherent geometry [1],[8].

1.4 Painting Defects Inspection Tunnel

After more than six years of research, the Institute of Design and Manufacturing (IDF) from the Universidad Politécnica de Valencia, Spain (UPV), in conjunction with the Ford Motor Company factory in Almussaffes, designed and implemented an automatic Inspection system of paint defects in the Ford vehicle bodies that are manufactured there. Today the system has two patents [10],[11] and is operating properly and perfectly synchronized with the FORD production line in Almussaffes. The positive results generated by this system have driven its implementation in other factories around the world. The painting defects inspection tunnel developed by IDF and FORD has these three systems:

1.4.1 Vision system

Image acquisition: For the image acquisition, the tunnel has 12 high resolution CCD cameras which are assembled with high quality lenses with variable focal distance with respect of the distance to the object. The taken images are transferred to a computer by data acquisition cards which are connected to dedicated industrial computers.

Illumination: The system is based on the analysis of the shape changes of the light reflections generated on the car surface. It uses high frequency fluorescent lamps arranged along the vehicle body in an array of 11 luminous arcs.

Defects detection software: To analyze the images previously scanned and identify defects, software was developed applying computer vision techniques based in classification, localization and segmentation algorithms and geometry extraction combined with feedback from others sensors. It also applies binarization of the captured image and mapping of defects through predefined patterns.

1.4.2 Data display system

After the tunnel there is a repair zone, where is displayed the processed images in displays for that the people that makes the reparation know where are the defects that have to be repaired. In the images, the defects detection software makes a red points for the big defects and green points for the small defects. This classification is agreed with the FORD quality standard.

1.4.3 Structure

To support all the systems mentioned, the tunnel has a structure made of Aluminum. The structure has a fixed part where is mounted the images acquisition system and a mobile part where is mounted the illumination system.

1.5 Justification

The dimension of the tunnel structure and the position of its components are determined by the body that is going to be inspected. Each body has different dimensions and different forms of the metal sheet. The purpose of the cameras is to take images of light reflection on the body, process it, and detect the variations on the reflection form which means that there is a painting defect. Taking in account this situation, the lights and cameras position is going to be different with each body inspected because the cameras have to be positioned in a correct way to take a proper image to the light reflection on the body.

When a new inspection tunnel is going to be developed, the first issue that has to be solved is the cameras and lights position which are based on the bodyės geometry. The cameras position is going to define the external structure position and its dimensions and the lights position is going to define the internal structure position and its dimensions (Figure 1). The issue is that the lights and cameras position is not so easy to determine because there are three variables that have to be combined: the body types that are going to be inspected in the tunnel, the camera focal angle and the light reflection. In order to avoid the ȯtry and errorȯ procedure, is necessary to run CAE simulations of different configurations of the three variables. Every change is a different tunnel configuration and is necessary to run a lot of simulations in order to found the best cameras position and tunnel configuration.

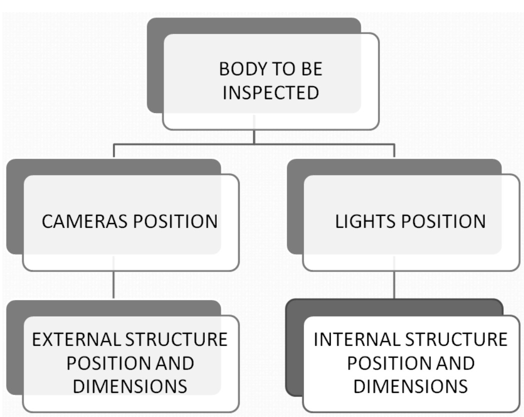

Fig. 1: Dimensional inheritance.

The inspection tunnel design that is already installed in Almussaffes was performed using Excel spreadsheets where are made the math calculations of a geometrical parameterization of the internal structure and were developed Visual Basic applications to transfer this data from the Excel files to NX6 and thereby generate the tunnel CAD model. With the CAD data, other Visual Basic applications are run to simulate the tunnel operation and obtain the images of the light reflections on the body. This procedure has to be repeated until find the better lights and cameras position. For every configuration, the designer has to modify the input parameters on the Excel spreadsheets and run the Visual Basic applications.

The Visual Basic applications were developed for NX6.0 and have to be modified if wants to use it in NX7.0 or NX7.5

With the intention to update the developments to NX7.5 avoiding the use of additional software such as Excel and Visual Basic, have a more direct interaction between the designer and the tunnel CAD model and knowing that it is a project that contains the ideal characteristics to break into the KBE application development, arises from the IDF and the CIMA the KBE application development using NX KF module through which the structure of a new inspection tunnel could be designed, replacing the Visual Basic application programmed and maintaining the current working conditions.

2 WORK DEVELOPED

To develop the KF application, the work already done by the IDF was taken as a reference and from the KF module all the modeling of the parts that make up the tunnel and its assembly was performed again.

The work developed was just the initial part of a bigger project which wants to automate the tunnel design beginning just with the body geometry as an input parameter. This project is focused on the design automation of the internal structure of the inspection tunnel taking as an input parameter the lights and cameras position.

The first action to develop the KF application was to redefine the geometric parameterization which was previously defined based on the dimensions from the geometric center, assuming the vertical and horizontal symmetry, the verticality of tubes number 1 and 5 and horizontality of tubes number 3 and 7 (Figure 2).

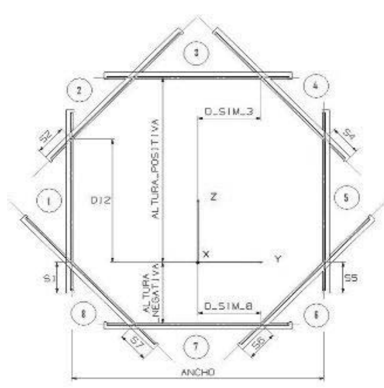

Fig. 2: Previous parameterization.

The new parameterization was defined with the Y and Z axis position of the initial point of the light tube and its inclination angle permitting a higher versatility of the lights positioning (Figure 3).

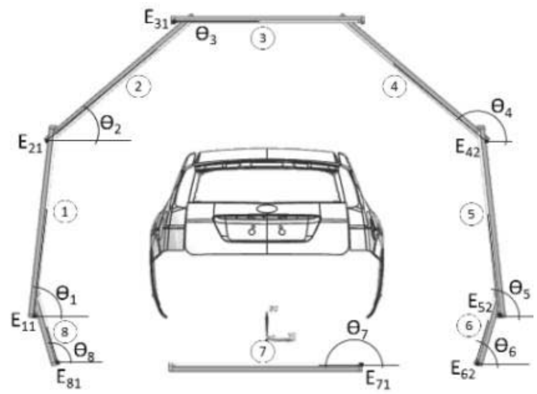

Fig. 3: Latest parameterization.

The Knowledge Fusion application development process was divided into 3 stages:

The first stage was the *Modeling* where with the use of the KF module tools were developed just nine *User Classes* to define each of the 31 pieces that make up the tunnel structure. Its performance shows the software capability to recycling design. The *User Classes* are a Group of *System Classes* and/or *User Classes* that are controlled by input parameters in form of *Attributes*. In this case, the input parameters are the geometric dimensions and physical properties of every part of the tunnel structure.

The second stage was the *Assembly*. In a NX file, *Attributes* were declared containing some of the mathematical calculations that before were made by the Excel spreadsheets. This information is transferred as a value to the *Attributes* of the *User Classes* that defines each piece of the tunnel structure and that were developed in the previous stage. This transfering is reached through a *Child Rule* called *ug_child_in_part*. A *Child Rule* is a special *Attribute* which instantiate a *User/System Class* and generates a CAD data of the part which dimensions depends on the value of its *Attributes*. In this manner, the parts geometry of the tunnel structure is controlled by the mathematical calculations made by the software. In the same way, through the *Child Rule* called *ug_component*, the addition of assembly components was controlled.

The last stage was the *Interface*. In this stage, a *User Interface* was developed using the NX software tool *UIS* (User Interface Styler). This tool allows designing a dialog box where the tunnel designer enters the initial parameters that for this work are the lights and cameras positions. The *Interface* designed was performed with the intention that the designer have in one place access to all options to make changes on the CAD model of the internal structure tunnel (Figure 4).

In the developed Interface, further than the designer can define a new inspection tunnel through the lights and cameras positioning, also can modify the cross section and the material of the internal structure elements, select from a catalogue the type of light tube that want to use, define the arcs quantity that want to model and controlling the visibility of every light tube in order to allow a partial simulation for example just for laterals or the roof.

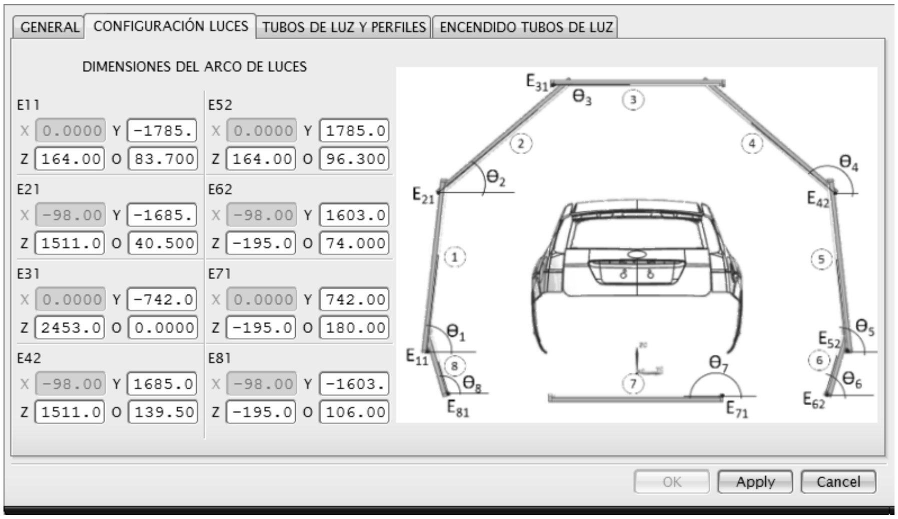

Fig. 4: User interface in NX.

The design scheme that was followed for the KF implementation in this work is shown in Figure 5. There is observed that the communication way between the designer and the software *Interface* was developed using the *UIS* tool. All information entered in the *Interface* is translated into a *User Class* called "tunnel" and stored in its *Attributes*. The "tunnel" *Class* is instantiated into a NX file in which the CAD model of the internal structure tunnel is generated. In this Class, there are other Attributes that contain the other part of the mathematical calculations of the parameterization and the design rules of the tunnel.

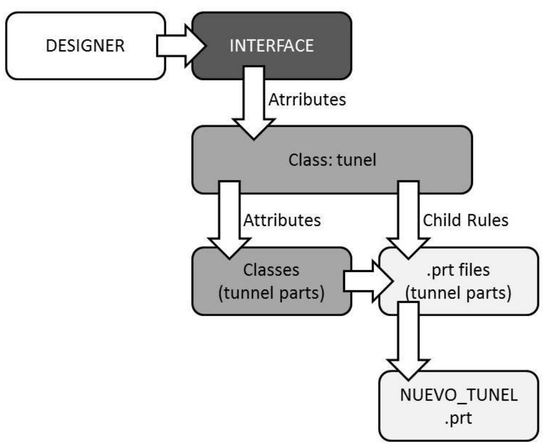

Fig. 5: Design scheme of KF application.

2.1 Results

Once the KF application was developed, it was implemented and it is currently being used by the IDF to do the modeling and simulation of inspection tunnels that are planned to implement in different automobiles production plants.

For CIMA, the knowledge gained can expand work opportunities within the design and manufacturing of components research line.

The implementation of the KF application allowed to automate the design of the moving structure of the defects inspection tunnel and therefore provides a direct communication interface between the designer and the tunnel CAD model by reducing the time spent in the design process.

With this development it was possible to migrate to NX the work that had been done in Excel and eliminate the need to develop Visual Basic applications. Having the information and calculations of the tunnel structure stored within NX ensures that at least as long as no radical changes in the functioning of the Software, the work done can be used and complemented in later versions of NX. The early development of the application was made in NX7 and completed in NX7.5 and found no problem whatsoever with the software versions change.

Although during the application development there were some problems with KF module operation, especially with the development of User Classes, at the end of the application it was possible to obtain 100% module usage. With KBE methodologies implementation in the tunnel structure design, now it is possible to obtain information from non-geometric parameters of the tunnel such as the total weight of the structure. This information for the previous development was unknown to the designer.

3 CONCLUSIONS

During the KF application development it was experienced that using the module is not as intuitive and friendly as other NX modules, it is essential to know at least the bases of object-oriented programming to be followed for applications development. Operations such as Blends or extrusions in KF require more time than is used in the CAD module for the same operations.

As new versions of NX are developed, more operations are integrated to KF module, to date, NX7.5 version still fails KF full integration with other software modules. Because of this, when developing applications, you may find some limitations. An example is the creation of Sketches. KF does not have a specific class to control the elements found in a sketch and it is necessary to resort to the Expressions for its control.

Despite the limitations that may have the module, KF allows us to reach levels that with CAD module is not possible to achieve. The additional time spent on the KF application development is offset by the reduction in time achieved with design reuse.

REFERENCES

[1] Scholz, C.; Egging, N.: Knowledge-Based Engineering (KBE) design methodology at the undergraduate and graduate levels, Int. J. Engng Ed., 16(1), 2000, 21-38.

[2] Bermell-García, P.; Fan, I.: A KBE system for the design of wind tunnel models using reusable knowledge components, VI International Congress on Project Engineering, 2002.

[3] Wong, L. M.; Wang, G. G.: Development of an automatic design and optimization system for industrial silencers, Journal of Manufacturing Systems, 22(4), 2003, 327-339. http://dx.doi.org/10.1016/S0278-6125(03)80041-9.

[4] Oïbrien, W. D.; Chandonais M.; Henderson, R. S.; Prezbindowski, M.; Hartman, N. W.; Rasche, J. A.: Using knowledge-based solid modeling techniques and airfoil design data: a case study in developing an airfoil seed part generator, IJME - INTERTECH Conference, 2006.

[5] Siemens: http://www.plm.automation.siemens.com/es_es/products/nx/prog_custom/kbp.shtml, Siemens NX help, 2010.

[6] Yang, H.-L.; Deng, F.: Knowledge fusion based product visual customization technology, 2nd International Conference on Computer Engineering and Technology, 5, 2010, 556-560.

[7] Vidal N. R.; Bermell-García, P.; Mulet, E. E.; García, F. L.: Definición de una arquitectura para la asistencia en el diseño de productos, 2004.

[8] Sanz, A. J. M.; Prádanos, D. P. R.; Gobernado, E. M.; Rodríguez, A. J.: Diseño en base a reglas con Unigraphics NX, XVI Congreso Internacional de Ingeniería Gráfica, 2004.

[9] Wang, S.-H.; Melendez, S.; Tsai, C.-S.: Application of Parametric Sketching and Associability in 3D CAD, Computer-Aided Design and Applications, 5(6), 2008, 822-830. http://dx.doi.org/10.3722/cadaps.2008.822-830.

[10] International patent (ref.PCT/ES2007/000236) based on artificial vision and distributed computing aspects.

[11] International patent (ref.PCT/IB2010/052193) describing the industrial system already implemented at the Almussafes Ford Factory.