DEPENDENCY IDENTIFICATION FOR ENGINEERING CHANGE MANAGEMENT (ECM): AN EXAMPLE OF COMPUTER-AIDED DESIGN (CAD)-BASED APPROACH

Masmoudi, Mahmoud (1,2); Leclaire, Patrice (1); Zolghadri, Marc (1); Haddar, Mohamed (2)
1: SUPMECA, France; 2: Université de Sfax, Tunisia

Abstract

Engineering change management is a research field in which the goal is to deal with modifications of products and systems. Methods and tools are set up to predict more efficiently the propagation of changes or to assess the consequences of these changes. This paper addresses the fundamental research question of dependency identification in a product. We propose an approach based on the use of specialised support tools and knowledge to map the dependency links among components or parameters of a product. These dependencies are expressed quantitatively, qualitatively or as a polynomial function. The general approach is here applied to a geometric 2D model of a bicycle. The analysis of the obtained dependency graph gives insights to designers for modification of existing products or for creation of new robust products.

Keywords: Engineering Change Management (ECM), Dependency, Computer aided design (CAD), Evaluation, Decision making

Contact:
Mahmoud Masmoudi
SUPMECA
Laboratory of Mechanical Systems and Materials (LISMMA)
France
mahmoud.masmoudi@supmeca.fr

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

A product may be viewed as an association of a structure of inter-connected components and sub-systems. To answer new requirements that can be provided at any phase of the lifecycle, designers have to improve, adapt or upgrade the product by changing one or some of the components or functions of an existing product. The integration of each potential change can be propagated throughout the whole product conducting to extra cost and delay of the product upgrade. Engineering change management (ECM) deals with identifying and predicting change propagation (Jarratt, 2011). Among others, modelling of the inter-dependencies of design parameters or variables is one of the key issues in ECM. There exist methods and techniques in the scientific literature dealing with dependency modelling and the goal of the paper is to contribute to the solution to this central problem. The paper reports on research that used Computer-Aided Design (CAD) packages and their parametric ability to calculate the impact of one component's changes to others, to identify quantitatively and qualitatively the dependency among all components. This allowed the authors to map the dependency graph of the whole product assembly. Mainly, this graph models the propagation patterns and contains several kinds of knowledge that could be extracted from it for further analysis. These analyses give insights to designers in terms of robustness of the design of existing product and future alternatives. A more general goal behind this exploration is to suggest a generic method which makes use of existing formalised knowledge in analysing the change propagation.

This paper is organised as follows: Next section reviews some existing methods and approaches to capture and represent the dependency relations of a product design. In the third section, we propose novel method to characterise the dependency relations, in quantitative, qualitative and functional manners, between elements in a geometrical product design, using CAD package functionalities. The method is applied on a case study of a bike and its geometrical 2D representation. Finally, we discuss the results obtained, the advantages and shortcomings of the approach. Future research opportunities are identified. The general approach is then presented before finishing the paper with some conclusions and perspectives of future research.

2 STATE OF THE ART

In the engineering change (EC) literature, many authors have proposed definitions, tools and methods to characterise, evaluate and propagate changes. This section presents a brief survey of these works. Interested readers could refer to (Jarratt et al., 2011) and (Hamraz et al., 2012) for an extensive state of the art of methods and tools developed to deal with modelling and management of changes.

Wright (1997) defines Engineering change as a modification to a component of a product, after that product has entered production. Huang and Mak (1999) extend this definition considering the design and development phases of product life cycle. They define engineering changes as the changes and modifications in forms, fits, materials, dimensions, functions of a product or a component. Terwiesch and Loch (1999) consider the timing when the change happens; ECs are changes to parts, drawings or software that has already been released. Finally, Jarratt et al. (2003) gave a more complete definition: "an engineering change is an alteration made to parts, drawings or software that have already been released during the product design process. The change can be of any size or type and it can involve any number of people and take any length of time."

It is now largely known that changes can be propagated from one component to its dependent ones.

To deal with predicting change propagations, the key issue is dependency mapping. Some authors rely on Design Structure Matrix (DSM) and Domain Mapping Matrix (DMM) to model and manipulate the dependencies by matrix. DSM is a square matrix with identical row and column labels corresponding to the elements of the system. Each cell in the matrix may contain a numerical or binary representation of the link between two elements. DSMs can aid in identifying the parts of a product or design tasks, and the parametric or precedence relationships between them (Eppinger et al., 1994).

DMM is a second type of matrix-based approach proposed by Danilovic and Browning (2007). Whereas DSMs are strictly square matrices, a DMM is rectangular $(n \times m)$, relating two DSMs representing two different project domains, where n and m are the sizes of each domain. Danilovic and Browning (2007) conclude that using DSMs and DMMs together may enrich the understanding of system's complexity, reduce uncertainty and increase knowledge for analysing dependency between system's elements.

The matrix-based approaches give the possibility to combine mathematical formalisation and algorithmic capacities to design powerful EC decision support systems. One of the most famous ones is the Change Prediction Method, see Clarkson et al. (2004), supported by a tool freely available for research purposes. This method looks for predicting and analysing the risk of change propagation in complex design. CPM analyses change behaviour by developing mathematical models and express the risk of change propagation by using likelihood and impact matrices. Likelihood shows the potentiality of the change propagation from the component j to k while the impact models an assessment of the consequences of such propagation. Finally, the risk is computed by the multiplication of these two parameters. This method was applied on the Westland Helicopters of rotorcraft design as a case study.
Cheng and Chu (2012) propose a method for change propagation modelling based on weighted networks. This quantitative approach is made of three steps: breaking down the product into assemblies or subsystems, identifying the connections between product's elements and developing the network model of product. Cheng and Chu (2012) present three assessment indices: degree-changeability which assesses direct change, reach-changeability that measures indirect change and between-changeability used to judge the parts that will change and be changed dramatically. The approach was illustrated through the example of Roots Blower.
Finally, the last approach presented here is the one presented in Kusiak and Wang (1995). The authors propose a methodology to assist designers in the negotiation of constraints. A dependency network model was used in their approach. This network is a four-tuple $G = (V, E, \Omega, \Psi)$, that contains vertices (V) representing the design variables, the constraints and the design goals of the studied system, edges (E) corresponding to the relationship between vertices, and qualitative (Ω) and quantitative (Ψ) values carried by the edges. A reduction algorithm was developed to represent and derive dependencies (δ_{ij}) between decision variables or specifications, constraints and goals.
Apart from all the methods and tools presented and developed by academics, we can also include very largely used modelling and analysis software packages such as Computer-Aided Design packages, CATIA for instance. As Jarratt et al. (2011) pointed out, they can support the prediction of a change (a geometric dimension for instance) impact by analysing the product geometry and calculating the mismatch to neighbouring components. The major drawback of this approach is that it can only identify the immediate implications of a change rather than the consequences of change propagation and currently do not consider functional relationships.
To conclude this brief state-of-the art on Engineering Change, one can see that most of the approaches for modelling the change use a design structure matrix or, equivalently a graph. Matrix or graph can carry additional information to characterise the nature of a dependency between two elements. This information can be Boolean (existence), numerical (quantitative) or qualitative. Several approaches have been proposed to identify these dependencies, but none of them try to capture the relation as a function; often it is seen as a black box. This brings us to the major shortcomings of such approaches. These approaches are generally developed to be easy to use. The manipulated models make use of simple parameters such as those ones used in CPM: likelihood and impact. These approaches are sometimes necessary to cope with lack of detailed formalised knowledge or when the goal of the study is to analyse the system in a simple way. Otherwise, these approaches cannot be successful or will provide too simplistic results. These shortcomings are therefore:

- Complexity versus feasibility. When the dependency between two parameters of models is searched, often these approaches rely on the field experts to capture the knowledge in terms of evaluation of the model parameters; for instance likelihood and impact. Such pragmatic approaches show nonetheless their limits as soon as the complexity of the product or the number of considered parameters or modules increases; no one can imagine asking for characterisation of dependency between hundreds of parameters due to patent cost and feasibility constraints.
- Expert knowledge versus formalised knowledge. If there exists formalised knowledge about the dependency of two parameters (via formulas or specific tools such as CATIA), these bits of precise knowledge are replaced by some too simplified parameters for change propagation models. These simplifications are too drastic to be acceptable.

The approach proposed in this paper looks to find the right trade-offs between a too much detailed change management models and too much simplified ones and will be discussed at the end.

3 A CAD-BASED APPROACH FOR DEPENDENCY ANALYSIS

In this section, we present our approach to deal with dependency analysis.

3.1 The proposed approach

Four steps are necessary to study change engineering: modelling the system, defining the change by selecting the instigating elements, characterising the dependency between elements and interpreting the results, cf. Figure 1. The first step consists in a construction of an initial model of the studied system. This model contains the necessary information about the system structure/architecture. We also formalise the constraints associated with parameters. In the change definition step, the identification of the change causes and the selection of the instigating element are essential. According to Clarkson et al. (2004), instigating element represents the sub-system or the component corresponding to changes dealing with new product requirements. At the third stage, we will consider the dependencies between every couple of elements of the system's model and we look for characterising the dependency by a function linking them. Finally, all the obtained knowledge about the dependency of parameters and/or components are analysed at the last step of the method.

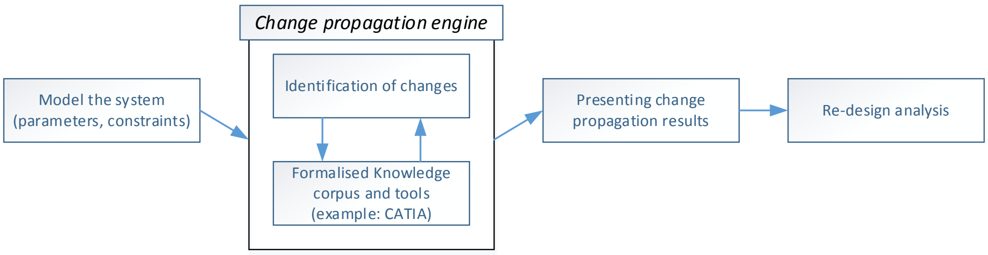

Figure 1. General approach

3.1.1 Modelling the system

In a classic design approach, the modelling is first done only in 2-Dimension (2D), then in a 3D model (Delamé et al., 2011). One can create a geometric model with CAD (Computer-Aided Design) software packages such as CATIA. A 2D model contains $n + m$ elements called "dimensions". Each dimension $\mathrm{D_i}$ can be either a length $\mathrm{L_a}$ or an angle θ_b.

$$\text{Model} = \{\mathrm{D}_i,\ \mathrm{i} = 1..\mathrm{n}+\mathrm{m}\} = \{\mathrm{L}_a,\ \mathrm{a} = 1..\mathrm{n}\} \cup \{\theta_\mathrm{b},\ \mathrm{b} = 1..\mathrm{m}\} \quad (1)$$

To ensure a link between the different segments or points of the model, it is necessary to impose some necessary constraints between them. Constraints are defined as restrictions and associations that are applied to the 2D geometry. The constraints can be divided into two categories. The definition constraints, such as Length Constraints ($\mathrm{C_L}$) and Angle Constraints ($\mathrm{C_A}$), which associate a definition domain with a given dimension. The second class of constraints, called structure constraints, impose relationships between geometrical elements, such as parallelism, coincidence, or concentricity, for instance. These are all necessary and sufficient constraints to define a geometric object in CATIA.

We denote by D_i^0, $\overline{\mathrm{D_i}}$ and $\underline{\mathrm{D_i}}$ respectively the initial value, upper limit and lower limit for each dimension $\mathrm{D_i}, \forall i = 1..n+m$. These limits are determined by designers according to standards, past experiences and blueprints, necessary for satisfaction of needs and requirements.

3.1.2 Defining the change

In order to characterise the dependency between system's elements, we have to study the influence of a change in the value of each dimension $\mathrm{D_i}$ on other $\mathrm{D_j}$, $\mathrm{j} \neq \mathrm{i}$. It is assumed that for a given i, $\mathrm{D_i}$ can vary around $\mathrm{D_i}^0$ in $\left[\underline{\mathrm{D_i}}\,;\,\overline{\mathrm{D_i}}\right]$. We denote δ_i the difference between the new value $\mathrm{D_i}'$ and the initial one D_i^0, such as $\mathrm{D_i}' = \mathrm{D_i}^0 + \delta_\mathrm{i}$ and $\delta_\mathrm{i} \epsilon \mathbb{R}$.

In order to gather the change impacts of dimensions, we use a discretization of the interval $\left[\underline{\mathrm{D_i}}\,;\,\overline{\mathrm{D_i}}\right]$, by taking a number k_i of successive distinct values in $\left[\underline{\mathrm{D_i}}\,;\,\overline{\mathrm{D_i}}\right]$. One can use different discretization methods, such as fixing the sampling step σ_i then computing the k_i values, or fixing the number k_i and then computing the sampling step σ_i. In both cases, the sampling step σ_i is a constant value.

3.1.3 Characterising the dependency

To characterise the dependency, we should first test possibilities of change on different dimensions and then identify the type of dependency between the dimensions pairwise.
Let us suppose that our model contains five dimensions and we choose to change the dimension (D1) and to observe the modifications of others. The results are represented in Table 1. In this example, D1 represents the instigating dimension; the others are the affected dimensions. Each row contains an instance of a change on $D_1 \in [D_1^0 - 2; D_1^0 + 2]$ with a sampling step $\sigma_i = 1$.

Table 1. Example of one change results given different changes on D1

D1	D2	D3	D4	D5
D_1^0-2	D_2^0+2	D_3^0	D_4^0-4	D_5^0+8
D_1^0-1	D_2^0+1	D_3^0	D_4^0-2	D_5^0+3
D_1^0	D_2^0	D_3^0	D_4^0	D_5^0
D_1^0+1	D_2^0-1	D_3^0	D_4^0+2	$D_5^0 - 1$
D_1^0+2	D_2^0-2	D_3^0	D_4^0+4	D_5^0

In this example, D_2, D_4 and D_5 depend on D_1 because changing D_1 modifies them while D_1 and D_3 are patently independent. The Figure 2 provides an overview of these types of dependencies.

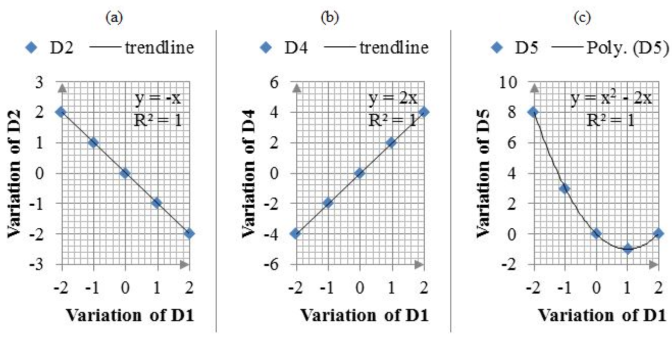

Figure 2. Types of dependency

The two dependency plots linking variation of D_1 with variation of D_2 and D_4 are linear and their equation is given by $y = ax + b$; $a, b \in \mathbb{R}$ (Figure 2.a and Figure 2.b) while a curve (a polynomial relation) models variations of D_5 according to variations of D_1, see Figure 2.c. In this case, the dependency is non-linear. The main goal of this second step is then to characterise as much as possible the dependency link between each pair of model dimensions.

3.1.4 Presenting and interpreting results

The computed dependency data are collected and saved in a three dimension table named $G = (g_{i,j,k})_{1\le i\le n, 1\le j\le m, 1\le k\le \max_i k_i}$. Each $g_{i,j,k}$ represents the gap/variation between the new value of D_j and D_j^0 after the kth modification of D_i, where i, j and k represent the indexes of the instigating dimension, the affected one and the sampling respectively. In order to characterise dependence between dimensions in a more precise way, we convert the table G to the following matrices:

- The Average Dependency Matrix, denoted by $A = (a_{i,j})_{1\le i\le n, 1\le j\le m}$ and defined by:

$$a_{ij} = \frac{1}{k}\sum_{k=1}^{k_i} g_{i,j,k} \; \forall i \in \{1..n\}, \forall j \in \{1..m\} \quad (2)$$

The matrix A presents the average of the variation of each dimension D_j and its initial value D_j^0, after k modifications of D_i.

- The Variance Dependency Matrix, denoted by $V = (v_{i,j})_{1\le i\le n, 1\le j\le m}$ and defined by:

$$v_{ij} = \frac{1}{k}\sum_{k=1}^{k_i} (g_{i,j,k} - a_{ij}) \; \forall i \in \{1..n\}, \forall j \in \{1..m\} \quad (3)$$

The variance characterizes the dispersion of the variation values of each dimension D_j, after k modifications of D_i.

- The Quantitative Dependency Matrix, denoted by $Q = (q_{i,j})_{1\le i\le n,1\le j\le m}$. Its components are defined by the slope of the regression line or the slope of the tangent line according to the regression type T (linear/polynomial) used :

$$q_{ij} = \begin{cases} \dfrac{\sum_{k=1}^{k_i} z_{ik}\left(D_j^0 + g_{ijk} - \frac{1}{k_i}\sum_{k=1}^{k_i}(D_j^0 + g_{ijk})\right)}{\sum_{k=1}^{k_i}(z_{ik})^2}, \text{ if T = Linear} \\ \left(\dfrac{y1 - D_j^0}{x1 - D_i^0}\right), \text{ if T = Polynomial} \\ 0, \text{if } v_{ij} = 0 \end{cases} \quad (4)$$

where $z_{ik} = \left(\underline{D_i} + (k-1)\sigma_i - \frac{1}{k_i}\sum_{k=1}^{k_i}(\underline{D_i} + (k-1)\sigma_i)\right)$ (5)

and $(x1, y1)$ corresponds to a point coordinates of the tangent line passes through (D_i^0, D_j^0).

Q measures the quantitative dependency existing between the instigating and the affected dimensions. It allows comparing and classifying the different dependency relations between system's elements. In addition, qualitative dependency can be concluded from the quantitative dependency.

- The Binary Dependency Matrix denoted by $B = (b_{i,j})_{1\le i\le n,1\le j\le m}$ and defined by:

$$b_{ij} = \begin{cases} 1 \text{ if } q_{ij} \neq 0 \\ 0 \text{ otherwise} \end{cases} \quad (6)$$

The matrix B provides a general information indicating whether a dependency relationship exists or not between two different dimensions.

- The Qualitative Dependency Matrix $S = (s_{i,j})_{1\le i\le n,1\le j\le m}$ and defined by:

$$s_{ij} = \begin{cases} +1 \text{ if } q_{ij} > 0 \\ -1 \text{ if } q_{ij} < 0 \\ 0 \text{ otherwise} \end{cases} \quad (7)$$

The Qualitative Dependency Matrix S is more specific in characterising the dependency than matrix B and shows the sign of variation of D_j when we change the value of another dimension D_i.

- This matrix is important when there are only few available data and the only possible characterization of the dependency is the qualitative using expert knowledge.The Functional Dependency Matrix, denoted by $F = (f_{i,j})_{1\le i\le n,1\le j\le m}$ and defined by the expression of dependency function from D_i to D_j :

$$f_{ij} = \begin{cases} y = \sum_{l=0}^{L} x_l\, x^l, if\ b_{ij} = 1 \\ 0 \text{ otherwise} \end{cases} \quad (8)$$

where L is the highest degree of the polynomial (if T=linear, L=1) and $x_l \epsilon \mathbb{R}\ \forall l = 1..L$.

This matrix contributes to a better description and characterisation of the dependency relationships required for a good understanding of the complex system's element interactions.

3.2 Case Study Research

A real bicycle is used to illustrate the approach step by step.

Step1: Modelling the system. The geometric model, made with CATIA, is a sketcher model of the bike composed of segments and circles. This model contains twenty four dimensions D_i, twelve lengths L_a and twelve angles θ_b, are identified in our model ($n = m = 12$).

$$\text{Model} = \{D_i,\ i = 1..24\} = \{L_a,\ a = 1..12\} \cup \{\theta_b,\ b = 1..12\} \quad (9)$$

Figure 3 shows the twelve lengths created and their initial values (Length.i). An angle θ_b is associated with each length L_a corresponding to the angle formed with this segment and the horizontal axis.

In order to get a feasible bike, we have to integrate required constraints in the model. Structural constraints are defined in CATIA interface by including nine parallelism constraints (C_P), nineteen coincidence constraints (C_C), three concentricity constraints (C_O), three fixation constraints (C_F) and three distance constraints between two lines (C_D). For example, the distance between the saddle and the handlebar is defined as (C_D). Definition constraints, gathering constraints on lengths (C_L) and angles (C_A), are defined for dimensions D_i by associating an interval $\left[\underline{D_i}\,;\,\overline{D_i}\right]$ of possible values for each dimension D_i. Nevertheless, these intervals cannot be directly integrated into the CATIA model; CATIA does not offer the possibility to support the "parametric" definition of dimensions. To integrate them, we propose an a posteriori checking of these constraints according to the Algorithm presented in the appendix of the paper.

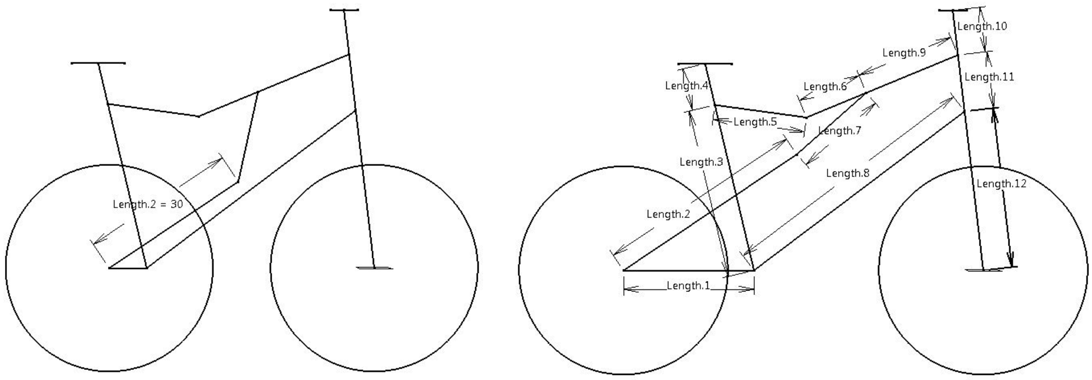

Figure 3. The studied bike and its model

The upper limit $\overline{D_i}$ and the lower limit $\underline{D_i}$ for each dimension D_i are chosen as the critical values of each dimension producing an unfeasible bike. Considering for instance the Length 2, the lower limit $\underline{D_2}$ is the minimum length that allow to obtain a feasible bike, see the upper-left side bike model in Figure 3 where $D_2 = 30 < \underline{D_2}$ generates an unfeasible bike.

Step 2: Defining the changes. In this second step, after indicating the lower and upper limits for each dimension, we choose to fix a sampling step: $\sigma_i = 1\ \forall i = 1\,..\,24$.

Step 3: Characterising the dependency. In this step, we propose a tool to study a desired change. This tool is based on the modification of the CATIA model for characterising the dependency between the dimensions.

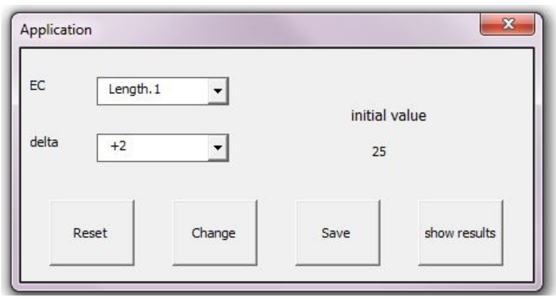

Figure 4. Overview of the developed change tool

This tool has been made by using VB for CATIA. This application contributes to change the selected dimension (EC in Figure 4) from the initial value D_i^0 to a new one $D_i' = D_i^0 + \delta_i$. δ_i in Figure 4, varies between the lower variation $\underline{\delta D_i} = (D_i^0 - \underline{D_i})$ and the upper variation $\overline{\delta D_i} = (\overline{D_i} - D_i^0)$. From these results, for each instigating dimension D_i we plot the variations of its dependent dimension (affected ones) D_j. Figure 5 shows a sample of the dependency linking L3 with other dimensions, characterising the type of the dependency linear/ polynomial in each case. Finally and after repeating these operations for all the twenty four dimensions from $\underline{\delta D_i}$ to $\overline{\delta D_i}$, the G matrix contains all the collected data.

Step 4: Presenting and interpreting the results. As mentioned before, the elements of the three matrices B, S and Q are calculated from the table G. The left side graph of Figure 6 is obtained from these matrices. Every node corresponds to one dimension. An edge shows the existence of dependency between the two connected nodes; its colour indicates the strength of dependency. The graph shows the complexity of this type of analysis, with twenty four nodes and 162 edges. These results therefore need to be interpreted with caution.

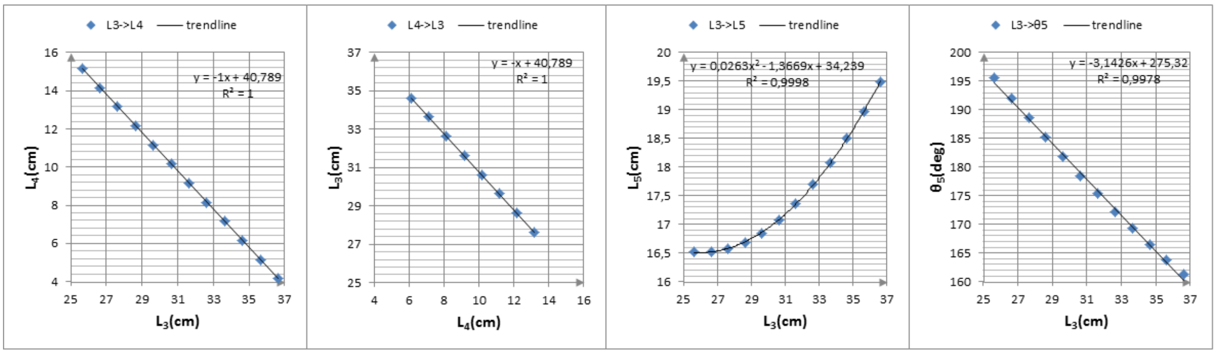

Figure 5. Example of dependency plots

3.3 Discussion about results

Let us analyse the dependency graph more in detail. Observing attentively this graph, it may be noticed that nodes can be classified according to the number of their incoming and outgoing edges into two types: dominant or dominated. θ_{12} is a dominant node because each change made in this node causes changes to lots of other nodes (17 in total). On the contrary, the node L_5 is a dominated one as it can be affected almost with all of the other nodes.

However, it can be observed that the graph should be screened for easier analyses. Screening methods should be identified in order to allow for instance to represent only those meaningful dependency links (using threshold techniques for instance), linear ones and non-linear ones. Non-linear dependencies require a careful analysis to detect the sensibilities of one parameter regarding the other one. This is the case for instance the curve representing the variations of L_5 according to variations of L_3. First, there is a horizontal part of the curve (close to 25cm), which shows a local independency between them while passing 27cm, the variations of L_5 are quite important; i.e. local dependency. This shows that the variational analysis can be performed by intervals. Moreover, one can use asymptotic analysis in order to simplify decision-making based on the real observed behaviour.

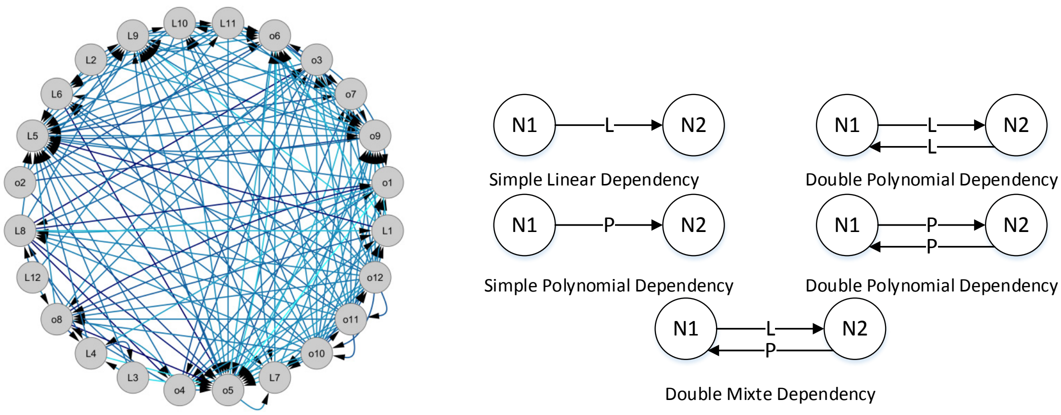

Figure 6. Dependency graph and taxonomy of dependencies

In the example mentioned in Figure 5, we can identify a double linear dependencies (between L_3 and L_4), a simple linear dependency (between L_3 and θ_5) and simple polynomial dependency (between L_3 and L_5). The dependencies between nodes therefore could be simple (oriented in one way) or double, see right side of Figure 6. However, one may hardly do this analysis on the graph. It does not allow noticing these interesting characteristics related to mutual dependencies. Indeed, our study demonstrated that different situations of dependency between two dimensions can exist. The determination of existence or not of such mutual dependencies can be performed using appropriate algorithms manipulating the matrix G.

Mutual dependency links constitute an important driver for the analysis of the change propagation. If the mutual dependency links are both linear, it is not necessary to pay attention to the order in which the changes are applied to nodes. The superposition principle can be applied to predict the final change propagation behaviour. The existence of a mutual dependency made of one linear and one non-linear link eliminates this possibility. For instance changing the value of L_1 first and then $\theta_3$does not generate the same value of L_5 than the changing first θ_3 and then L_1. It is therefore necessary to determine that the right order of changing values of the parameters which limits best the propagation of changes throughout the whole set of parameters.
The general approach that authors are working on, see Figure.1, stipulates that during the determination of dependency graph, we can use not only expert knowledge available but also available specialised tools and methods. This allows to launch change prediction and change propagation in a more precise way. The tool can be a behaviour simulator and if not we could make use of available knowledge and formulas, cf Kusiak and Wang (1995). The second box shown in Figure.1, called the change propagation engine, is therefore composed of one change generator (the tool such as that one developed in VBA) and combined with a field-specialised tool (CATIA in our example). In this case, in an iterative way, changes are suggested to this latter tool which has to compute the consequent modifications on other parameters. The iterations are performed to browse the whole possible variations space. The obtained results are then collected for further analyses. The used illustrative example gives promising results while shows some limitations and necessary improvements.

4 CONCLUSION

The main goal of the current study was to suggest a new approach of engineering change prediction to characterise the nature of dependency relationships between components of a product.
A graph and different matrices are proposed for change impact assessment. This characterisation can be Boolean, quantitative, qualitative or functional. The contribution of this article is to use CAD package as a simulation environment to rebuild the dependency functions between components, from the geometrical point of view. The analyses we can make based on qualification and quantification of dependencies give insights in terms of change propagation behaviour and can be used by designers as a key indicator in terms of choice of changes to implement and the order in which they are applied. The dependency graph allows determination of the most dominant and most dominated nodes. Again this can be used as a driver for the determination of changes that should be avoided due to the numerous impacts that it will create and that have to be implemented in their turn.
Different authors provide qualitative and quantitative dependency values. But in our approach, a functional dependency is introduced which procures a good analysis capability. Even though, qualitative dependency aids to predict either the increase or decrease of the affected dimension's value after a change in instigating dimension's value, functional dependency offers an effective way for characterising the relationship between two dimensions and translate it into a function which opens lots of analysis possibilities such as sensibility analysis.
The sample size can be reviewed, optimised and calculated according to the level of the necessary precision. The study of the non-linear dependency has to be pursued to identify appropriate techniques for the analysis of their proper behaviours and their impacts of multiple propagations. This issue also relates to the determination of the parameter σ_i which defines the analysis level. The choice of a constant value of σ is not relevant for non-linear dependencies: a too small value generates too much computation and decreases the computation efficiency while a too big value could ignore lots of fine-grain propagation behaviours.
The right choice of the parameter L (the highest degree) of the polynomial equations (see equation 8) introduces another question that has to be studied jointly with the value of sampling step σ. Further works must be done to improve the quality of the sampling of variation of parameters and to limit the general dimension of the solution space which could be a serious trap. These works have to take account of several steps propagation too. In fact, the propagation wave generated by one applied change has to be simulated till a stopping point which has to be precisely defined according to some criteria to determine.
Authors are working on the possibility of generalization of the approach for the complete product development life-cycle in further works, they focus also on considering the dependency between not only two parameters but several ones. In Kusiak and Wang (1995) for instance, the authors simplified

the n-dependencies as a set of n 2-dependencies. Further researches are necessary to be able to provide evidence to this possibility or to suggest new techniques allowing such multilateral dependencies consideration.

REFERENCES

Cheng, H. and Chu, X. (2012) A network-based assessment approach for change impacts on complex product. Journal of Intelligent Manufacturing, Vol. 23, No. 4, pp. 1419-1431.

Clarkson, P. J., Simons, C. and Eckert, C. (2004) Predicting change propagation in complex design. Journal of Mechanical Design. Vol. 126, No. 5, pp. 788-797.

Danilovic, M. and Browning, T. R. (2007) Managing complex product development projects with design structure matrices and domain mapping matrices. Int. J. of Project Management, Vol. 25, pp. 300-314.

Delamé, T., Léon, J.C., Cani, M.P. and Blanch, R. (2011) Gesture-based design of 2D contours: an alternative to sketching?. Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling, pp. 63-70.

Eppinger, S. D., Whitney, D. E., Smith, R. P., and Gebala, D. A. (1994) A model-based method for organizing tasks in product development. Research in Engineering Design, Vol. 6, No.1, pp. 1-13.

Hamraz, B., Caldwell, N. H. M. and Clarkson, P.J. (2012) A multidomain engineering change propagation model to support uncertainty reduction and risk management in design. J. Mechanical Design, Vol. 134, No. 10.

Huang, G. Q. and Mak, K. L. (1999) Current practices of engineering change management in UK manufacturing industries. International Journal of Operations & Production Management, Vol. 19, No. 1, pp. 21-37.

Jarratt, T. A. W., Eckert, C. M., Weeks, R. and Clarkson, P. J. (2003) Environmental legislation as a driver of design. In: Folkeson, A., Gralen, K., Norell, M., Sellgren, U. (eds) 14th International Conference on Engineering Design - ICED'03, Stockholm, Sweden, pp. 231–232/ CDROM.

Jarratt, T. A. W., Eckert, C. M., Caldwell, N. H. M. and Clarkson, P. J. (2011) Engineering change: an overview and perspective on the literature. Research in Engineering Design, Vol. 22, No. 2, pp. 103-124.

Kusiak, A. and Wang, J. (1995) Dependency analysis in constraint negotiation, IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 9, pp. 1301-1313.

Terwiesch, C. and Loch, C. H. (1999) Managing the process of engineering change orders: the case of the climate control system in automobile development. J. Product Innovation Mgt, Vol. 16, No. 2, pp. 160-172.

Wright, I. C., (1997) A review of research into engineering change management: implications for product design. Design Studies, Vol. 18, No. 1, pp. 33-42.

APPENDIX

Overview of algorithm

Input: integer n, m; real $D_i^0, \underline{D_i}, \overline{D_i}$, σ_i; $\forall i = 1..(n+m)$

Output: Matrices G, A, B, Q, S, F

Declare variables $D_i, \delta D_i, \underline{\delta D_i}, \overline{\delta D_i}, k_i$; $\forall i = 1..(n+m), \forall j = 1..(n+m) \forall k = 1..k_i$

For each instigating dimension $i \in \{1, n+m\}$ do

{$k_i = 1$

While ($D_i + (k-1)\sigma_i < \overline{D_i}$) {

For each dimension $l \in \{1, n+m\}$ do {

$D_l = D_l^0$// initialisation of the model by CATIA

}

$D_i = \underline{D_i} + (k-1)\ \sigma_i$

Verify that all dimensions D_l belongs to $\left[\underline{D_l}\ ;\ \overline{D_l}\right] \forall l \in \{1, n+m\}$

For each observable dimension $j \in \{1, n+m\} - \{i\}$ do {

$g_{i,j,k} = D_j - D_j^0$

}

$k_i = k_i + 1$

}

}

For each $(i,j) \in \{1, n+m\} \times \{1, n+m\}$ do{

Compute a_{ij} and v_{ij} according to (3) and (4)

If $v_{ij} \neq 0$ then {

$t_{ij} = 1$

C**ompute** R_{ij}^2 , the coefficient of determination for a linear regression

While ($R_{ij}^2 < 0.9$) do {

$t_{ij} = t_{ij} + 1$

Compute R_{ij}^2 , the coefficient of determination for a polynomial of degree t_{ij}

}

If $t_{ij} = 1$ then q_{ij} =slope of regression line

Else q_{ij} =slope of the tangent line

End If

If ($q_{ij} \neq 0$) then {$b_{ij} = 1$

$s_{ij} = sign(q_{ij})$

$f_{ij} = \{y = \sum_{l=0}^{t_{ij}} a_l\, x^l\ \}$

Else $q_{ij} = 0$, $b_{ij} = 0$, $s_{ij} = 0$, $f_{ij} = 0$

End If

}