INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

DEVELOPMENT OF A MULTILAYER CHANGE PROPAGATION TOOL FOR MODULAR PRODUCTS

Roberto Raffaeli[1], Michele Germani[1], Serena Graziosi[1] and Ferruccio Mandorli[1]
[1]Polytechnic University of Marche, I

ABSTRACT
In this work we focus our research on the product design related aspects; currently we deal with modularity, product architecture and change propagation issues along the design process. In order to apply abstract concepts to design practise different approaches and tools have been proposed; anyway presently concrete software solutions and applications examples are still lacking.
Companies modify their products for a number of reasons and rarely start from new ideas when designing. Due to the lack of suitable tools and methodologies designers are not aware of modifications impacts and propagations when trying to change or update a product.
In this paper we present our research efforts in developing a methodology and the related software tool to support change management during the product redesign. It is conceived as guiding tool based on a product multilevel representation: from functional contents to implementation design; currently the designer can obtain a complete presentation of the product parts characteristics and their relations. In this way the resulting graphical model becomes a company tacit knowledge repository about the product. Operational functionalities are provided to support the designer during his activities.
This work has been carried out and tested on the redesign process of a washing machine in collaboration with an Italian company, leader in house working appliances.

Keywords: Change Propagation, Product Architecture, Modular Structure

1 INTRODUCTION
From development to production, a product has already lost part of its market value: the level of excellence and quality immediately starts decreasing due to other competitive products on the market. Only with continual improvements a company can extend its products life, trying to adapt them as much as possible to customer needs. Managing the change process is very difficult because of the enormous factors to consider and especially because of the responsibilities involved. Modifying a product requires a huge redesigning and administrative efforts, which can be reduced only if the process is well managed. For instance, using products, processes and technologies already known it can surely reduce project risks and costs. It is clear how the management of modification impacts is a key issue that should not be underestimated.
One of the main factors that influence the way a change propagates is the product architecture. In order to estimate this spread is essential to obtain accurate and well-structured product architecture. Modifications occur for two principal reasons [1]: product enhancement and problem correction. In the first case the change is done in order to meet new expectations or raise the standards of the current product, for example new customer requirements, new legislation, technological progress, etc... In the second case the change is necessary in order to correct errors, reduce costs or plan a new production process.
The application of the modularity concept in the product development is already known from several years: it gives to the company the possibility of reaching a high level of flexibility thanks to the creation of a productive structure able to sustain the launching of customised product families on the market. On the other hand, the change process is very expensive in terms of time and money and, despite of the research efforts [2], [3], [4], suitable tools and methodologies to manage the problem do not exist. The most critical aspect is to know as soon as possible the change effects on the redesigning and manufacturing time and how these changes interfere.

2 PRODUCT ARCHITECTURE AND CHANGE PROPAGATION MANAGEMENT

Engineering Changes (ECs) can be defined as modifications in dimensions, shape, materials, etc… of products or constituents after design product realization [5]. According to Jarrat et al. [6], it is possible to identify two different types of changes: initiated and emergent. Initiated changes arise from the market and are considered as necessary in order to maintain the position of the product. The others refer to errors and malfunctions that appear along the product life and must be removed to let the product work properly. The change process becomes problematic when a component modification spreads all over the product because exceeding the margins of tolerance between the parameters that binds the components. The early evaluation of the impact produced by a change is fundamental because only after this analysis is possible to estimate costs, time and resources necessary to implement it.

The way a change can afflict a product depends essentially on its architecture that is the scheme by which the function of a product is allocated to its physical components [7]. Into modular product architecture, ideally, every physical component carries out only one function reported in the functional structure and the interfaces between them are de-coupled. In contrast to modular architectures, the mapping of function and component is not respected in the integral product structure where a component can carry out more than one function: this is defined function sharing [8]. It is hard to identify simple relations between the functional and the physical structures in integral products. The great part of products is situated in the spectrum between full modularity and full integration. Indeed whether a product can be considered modular or integral depends upon the level at which it is examined. Products can be composed of sub-systems that are modular in the way they link together, but each one can be highly integrated.

However, circumscribing the change management problem to the modular product, we consider two change typologies: local changes [9], which just involve one component or system, and interfaces changes, which imply a modification of the connections between modules. Once the interfaces need to be modified, the complexity of the issue strongly increases. Generally in a modular product the changes are implemented inside a module and then they can propagate only in it without afflicting the other modules. In the adaptation of modular product the interfaces are kept unchanged until they become outdated and a complete new product structure is required.

Even if it is evident that computer-based tools are essential to support changes, a lot of companies rely on a team of experienced people that keeps in mind product history. Such approach generates two orders of problems: the effective management and storage of a large quantity of information generated from strategic decisions (for example, why a change occurs, what are the change repercussions, and so on), the difficulty of the tacit knowledge transmission to the novices engineers.

On the one hand the current commercial PDM/PLM software packages, even if they are not able to assess the collective effect of several changes with their impact, offer solutions for supporting product and process information sharing. On the other hand, the advanced CAD systems, once a product has to be modified, can only predict the geometric behaviour without predicting the extended effect of changes; moreover the analysis can be carry out once the change has been modelled and a large part of the investment has already been made.

In conclusion, currently, there are not ready-to-use software tools supporting the management of the engineering change process. In the academic research world an interesting solution is the Change Prediction Method (CPM); this is an approach for calculating the combined risk that a change in one component affects another and for analysing indirect changes [10], [11]. This method predicts the likelihood, impact and resulting risk of a change propagation based on a product component breakdown using the Design Structure Matrix (DSM), which is a square matrix able to show the relations among the product components. An algorithm calculates the change propagation risks based on a combination of direct and indirect impacts and the change propagation likelihood between components. The problem of this tool, as stated by its authors, is the complexity and computational effort it requires. In addition, it does not take in consideration all the existing information coming from different departments: design, marketing and production. Moreover each component is characterized in terms of different aspect like the involved company sectors, costs, times, resources and so on, and then, all this different information can not be encompassed in a number. As a consequence it is necessary to deeply characterize each component using standard techniques able to collect and catalogue the information considering the different functions, collocation, main parameters and so on.

Our purpose is to create a tool that analyses the change propagation issues not in a statistic way. In our opinion statistic values can be influenced by lot of factors and are usually based on product experience matured on the past that might not be available. Research methods describing products quantitatively, use index and matrices so that a product is merely described in terms of numbers rather than in term of its functions. Product designing has changed considerably over the years but we think that engineers do not have to lose product functional structure because it represents a solid base point to start from for operating all sort of product analysis.
This is the motivation to analyse the product in a more deterministic way, having as objective not only the development of supporting tools but also the proposition of a new approach: a new way of thinking. We consider necessary to find a method really able to assist the designer in considering the change propagation implications of alternative solutions before investing considerable efforts into any of them. In order to support the engineers to find the best solution is mandatory to obtain an estimate of the likely workload, cost and duration of the change implementation. It should offer a wide vision of the product ranging from design activities till production ones.
These connections among components could be useful for the engineer to better understand the role that his competence area plays inside the product design activity. Moreover the tool should capture information from the experienced engineers in order to elicit knowledge about relations among components. It means to have the possibility to generate and store information on how engineers have taken decisions and then create an important database for the company growth.

3 APPROACH TO CHANGE PROPAGATION

Our proposed approach is based on a preliminary product functional configuration. Resulting structure analysis leads to modules identification. This process allows the definition of a product architecture, which can be browsed in order to identify change propagation paths. Property and relation concepts have been introduced to manage multilayer product characteristic and knowledge through a dedicated IT platform.

3.1 Modular products and product architecture

The identification of a common architecture within the company product lines has been studied to manage the product portfolio [12]. The literature describes both theories and methods to support the modularity concretisations through the product platform use [13]. Practical tools to support the implementation and their application are widely described [14], [15], [16].
A product can be defined modular and configurable if, on the basis of the product platform, it is possible to accomplish a rapid and low-cost design modification that fully satisfies the customer specifications.
The product platform can be represented, at high level, as a set of functional modules that are collected to achieve a generic overall function. A product is defined through a black box, which identifies just the main function of the system by means of input/output relations with the external environment. The main function is then represented trough the definition of sub functions and flows of material, signals and energy (SEM). This process is repeated at different level of detail maintaining the whole functional analysis as distant as possible from any implementation principle.
Once the functional model has been carried out, an analysis of the chains is performed to group together functions. Stone’s heuristic method [17] has been used to form modules that minimize interactions between them. Therefore SEM flows are split into internal and external flows.
The next step involves the introduction of solution principles to embody abstract tasks in real products. The product architecture consists of modules arrangement and all necessary relations between them. Briefly this architecture could represent a sort of database in which are stored all the useful information to completely characterize a product.
We believe as important to have a double vision of the product: one more abstract represented by the functional structure and a second one represented by the allocation of abstract functions to physical components. Our point is to maintain as much clear as possible the correlation between the functional structure and the structural one in order to consider that modifying a product does not only mean to eliminate or change one of its components, but also that this change could alter the its functional role.

3.2 Properties and relations

The central idea of this work is the definition of proper relations and constraints among components, in particular between their characteristics (we have called them “properties”), usable in a change management software tool. From this observation we have considered as necessary the creation of a method for classifying the product information and the relations among components: our objective is to create an original information network, which could be navigable in an active way.
Before starting the explanation of the relation structure is necessary to give some useful definitions.

- *Block* is a physical part of the product ranging from components, subassembly to an entire module itself. This definition allows maintaining a high level of flexibility in the approach. In fact some properties are referred to large assemblies while low geometrical details have to be associated with single components.
- *Property* is the information that represents a particular characteristic of a block which it belongs to. In this case we are considering also the possibility of using this technique to represent all sort of product architecture (modular or integral). It is obvious that every component or module could have one or more property, without any limit.
- *Relation* is a conceptual or physical linkage between blocks. While Jarratt et al. [6] define a linkage as a direct relation or connection that exists between product physical parts, we define a relation between two generic blocks as a constraint between couples of their properties. It is possible to specify the direction of the relation in order to understand which of the two properties constraints the other. This aspect is very important to create a network that can be navigated. As for blocks, each relation is detailed by properties. How explained before, a property is defined inside a component or module while the property relation specifies the information necessary to represent the relation between properties blocks.
- *Internal relation* is a relation that links two properties belonged to the same block. The information about the internal relation will be contained inside the internal property relation. In this case this type of property belongs to the internal relation, which is defined inside the product element.

Elements of the product architecture can be instantiated and put in relation each other. It is obvious that a block without any properties cannot have any relations with the other elements.
The concepts introduced until now, especially property and relation, represent the base of the change management method. To classify relations and properties we have introduced the following attributes: *category*, *type* and *name*. Table 1 presents an example of the categories with the related types and names utilized to classify a property. We aim to a generic classification for properties and relations in order to share a common terminology between different products.
The category is the attribute that identifies the ambit of the cataloguing. A correct choice of the category mainly provides the designing area of competence. The next step is the choice of property type: it is necessary to select a particular aspect of the chosen category. The choice of the type allows a deeper view on the product information classification, but it is not enough to completely characterize the product. In fact the main characteristic of a property is its name. Choosing a name means to select a parameter, which can identify a particular aspect of the product, which has been bounded by its category and type. A name could vary from geometrical information, like diameter, to production ones, like the manufacturing process that is necessary to realize the component. A name represents the specific circumstance in a given engineering situation. While the category and the type contain information that transcends the product, on the contrary the name is its characterizing element.
Besides, three different levels of information facilitate the relations insertion and filtering of specific ones. It is fundamental to operate a selection of the information choosing only what the designer is interested in, because it has influence on the system under reviewing.
Also for the relations cataloguing it has been utilised the same subdivision in category, type and name as done for the properties (Table 2). In this case, we have identified two categories: *physical linkage* that requires a means and *conceptual linkage*. Physical linkages are then split in direct and indirect. For example, a welding needs a direct contact between the parts while in an electric connection the means is represented by an electric cable without any contact between the two components.

Table 1. Cataloguing of the information: Properties

Category	Type	Name
Competence Area	Fluid dynamics	Referent
	Structural	Referent
	Material	Referent
	Electronic	Referent
		….
Geometric	Position	Tilt
		Position
		Orientation
		….
	Dimension	Width
		Thickness
		Volume
		Diameter
		Depth
		Height
		….
Technical Information	Technical Parameters	RPM
		Frequency
		Gear Ratio
		Quantity
		Material
		….
	Layout	Heads number
		Ways number
		…
	Experimental Tests	Heating
		Scratching
		….
Supply Chain/Production	Purchase	Supplier
		Cost
		…
	Internal	Time
		…
Aesthetic/Comfort	Aspect	Colour
		…

Table 2. Cataloguing of the information: Relations

Category	Type	Name
Physical Linkage	Direct	Welding
		…..
	Indirect	Wiring
Conceptual Linkage	Position	Parallelism
		Perpendicularity
		……
	Technique Decisions	Project Variants
		Quantity
		Material
		....
	Aesthetic	Color Matching
	Purchase	Supplier
		.....
	Production	Production Lines
		....

3.3 Software tool

Having defined a methodology to classify the information about the product, the next step has been the implementation of a software tool able to represent the product architecture, to store information, and in particular, to efficiently manage the product engineering changes. Component or module properties and their relations with other parts can be defined using the cataloguing explained before. The software tool has been implemented in a windows-based environment using MS.NET programming language.

While CAD systems create only a geometric representation of the product the developed software gives a logic representation of it. The final output is not the three-dimensional model but an information network generated by the linkage among components or modules properties. Therefore a product architecture consists of a group of *blocks* which are strictly connected each other (figure 1). Besides, to give a visible representation of the hierarchic structure of the product architecture father-son relationships have been used.

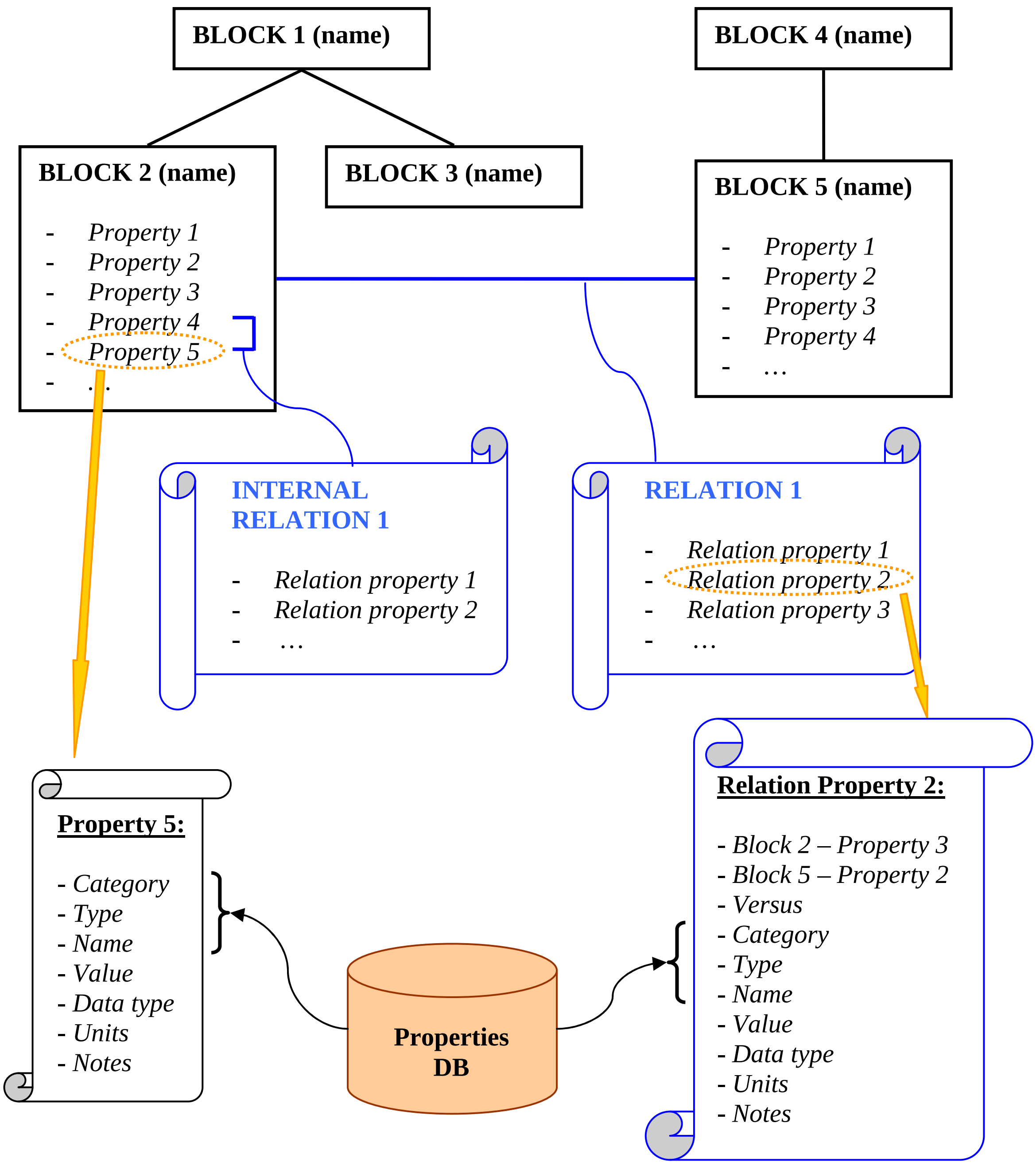

Figure 1. Product architecture representation in the software tool

The product structure based on more hierarchic levels gives the possibility to the user to decide how deep the product representation is. A chromatic scale underlines different levels of granularity. While father-son linkages have hierarchic meaning, relations are the means for identifying how changes in a block property spread through the structure.

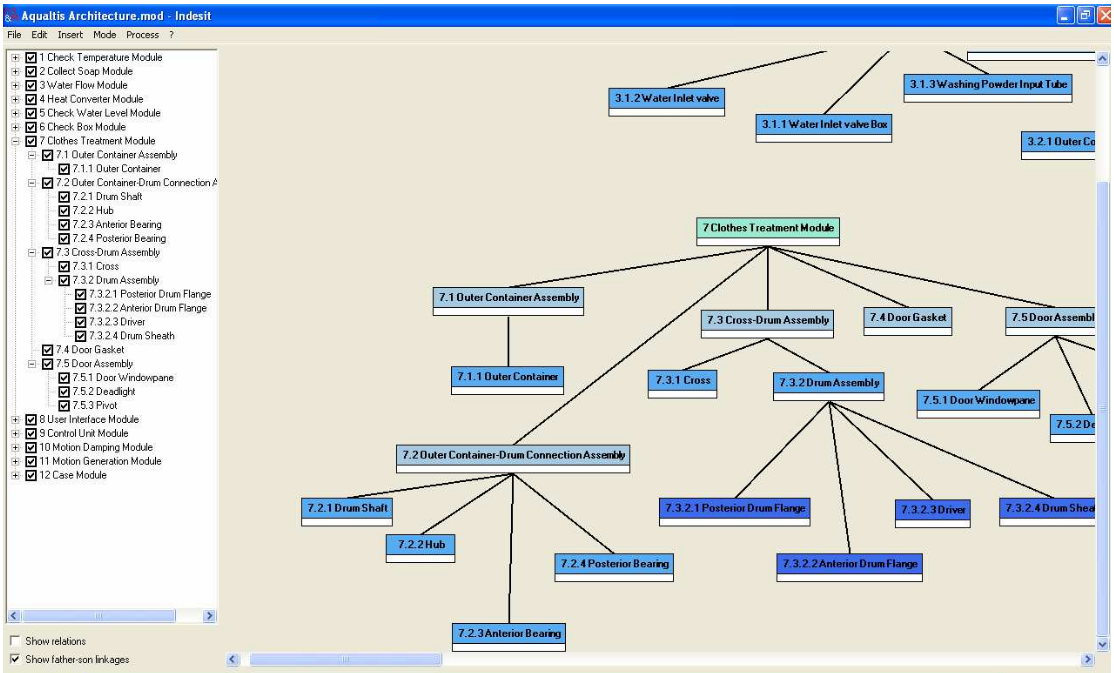

Figure 2. Screenshot of the software tool: en example of blocks hirerchical structure

Once created the product structure, the user has to define components and modules properties, and then he puts relations among them. Relations have to be specified by their properties in which are collected information about the relation direction and the reason of the linkage. The user can also define internal relations that connect properties of the same block.

The blocks are the means to sketch the product architecture. The black lines in the figure 1 represent the father-son linkages by which the module structure is organized while the blue ones are the relations among blocks. The hierarchic structure that the user has defined is well represented by the chromatic gradation chosen to represent blocks. Every block is identified by a name and a number, which represent its level of collocation inside the structure. A tree graph appears in the user interface to synthesize the structure as shown in figure 2.

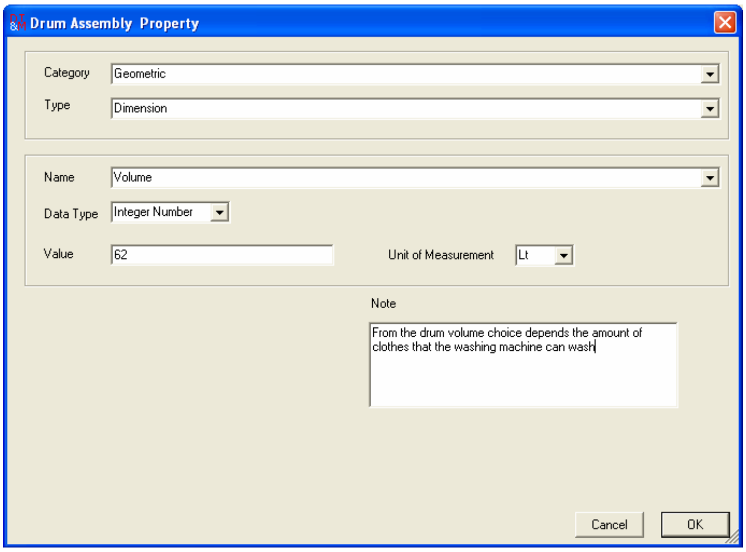

Figure 3. The property form

Through this representation technique, a user can choose how deep the product representation level is. The user can introduce relations represented by blue lines only if blocks he wants to connect have properties defined inside. For this reason we have organized the information definition phase using forms like the ones reported in figure 3.
This form is basically made up of two parts: general information and specific ones. On the upper part of the figure, there are two combo boxes to choose the category and the type of the property. On the lower part of the figure there are other combo boxes by which the user can choose the property name, the data type and there are also text boxes to introduce some values or comments. With the combo box related to data type, users can specify what type of value they want to introduce choosing among enumerative, text, integer and decimal numbers. With the enumerative data type users can assign specific values to characterize the property. In addition it is possible to assign the unit of measurement. Finally, using the text box identified with "Note" the user can insert same comments, for example specify why the property has been defined.
Combo boxes are connected to a database so the user has the possibility to choose categories, types and names already collected in it. Our purpose is to make the users share the same language. Software forms act as knowledge repository because the data collected are recorded in the product model file for future use.

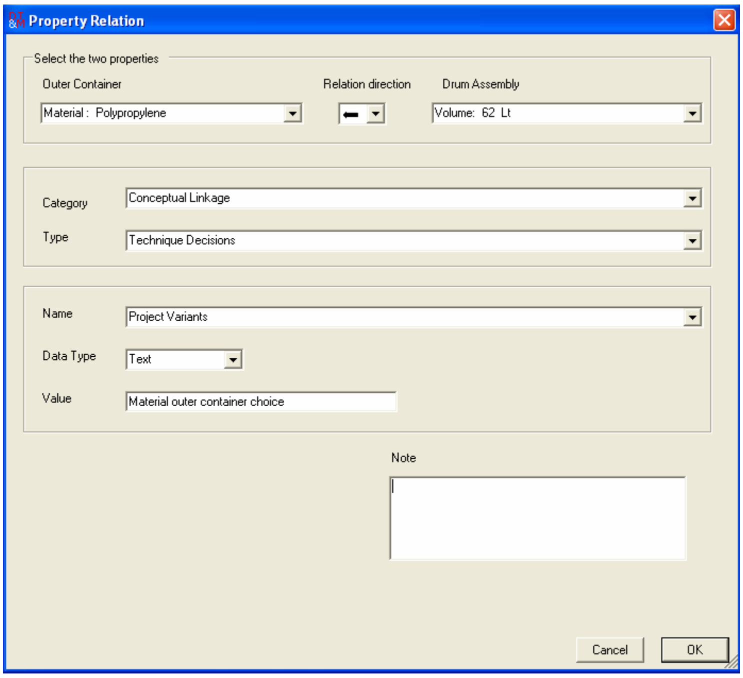

Figure 4. The relation property form

In the same way, the user can define a property relation. In this case he has to specify which properties of the two blocks he wants to connect and the relation direction choosing between direct, indirect or bi-directional (fig. 4).

3.4 Change propagation management

We have defined different change strategies, which can be classified into two different groups: the changes connected to the phase of the architecture definition and the other connected to the phase of the change propagation analysis. In the first group we consider the insertion of a new functional module and the change of an assembly position from a module to another. In the other group we consider the modification of a module or subassembly property and the change of properties relations.
To start the analysis of how a change can propagate through a product the user has to specify the name of the block that he wants to modify and, in particular, the property that needs to be changed. Since relation is a linkage between two specific properties, when one of these is changed the system considers it as invalid. To change back the relation status into valid it is necessary that also the other property involved is changed. In addition, if a relation to a third property connects the second one, also

this relation becomes invalid until the other property will not be changed and so on. If internal relations have been defined, change can propagate also inside the blocks.
Change propagation stops when to validate a relation is necessary to change a property, which has no more relations. In this way is possible to identify change propagation paths. Therefore the software can show how many elements are involved and how deep is the change. By knowing this information the user can have an idea of the redesign time.
Different level of analysis can be performed. For instance, the system may not consider the relation direction to find the change propagation paths. This leads to a wider level of propagation. Besides change propagation is due to the amount of aspects that the program must take into consideration. For example a property can have more than one relation with different blocks and more than one relation can connect two properties of two different blocks.
The information about the change propagation path is reported in a text file that is the current software output but it can be also graphically visualised by the user interface.

4 TEST CASE

The experimental work as been carried out and tested on the redesign process of a typical electromechanical product, a washing machine, in collaboration with an Italian company Indesit Company S.p.a., leader in house working appliances.
After having carefully analysed the product we have defined the product architecture using the technique described in the third paragraph.
In particular we have compared the product modular structure identified adopting Stone's methodology with the company product structure. We have analysed the product bill of materials and carried out interviews with engineers and designers with a wide range of experience within the company. Experienced engineers have given many examples of past changes, which had occurred in washing machines, mostly describing situations where the implementation of a change had not gone efficiently. In particular the key issue that has arisen is that designers find it hard to fully evaluate the complexity of linkages between parts that could cause changes to propagate.
Starting from these considerations we have realized a modular representation of the product classifying each washing machine component. In this way we have realized the product architecture, which can have various representations at different level of detail depending on the designer needs. Components like screws, wirings and so on have not been collocated inside modules because they have been evaluated as relations between components.
Once the functional product structure has been identified, we have focused our attention on the linkage analysis between components. With the help of the company engineers we have carried out this analysis from future-changes perspective. After having identified the product linkages we have catalogued the information, as explained in the third paragraph, in order to obtain a rational and well-structured representation of these linkages.
At this point the developed software tool has been used to represent the product architecture (figure 2), collect the product information and individuate the change propagation paths in order to automate the change impact analysis.

4.1 Product changes examples

Up to this point, the designer has completed the structure and has captured the knowledge related to a specific product. Since we suppose a new product definition like a number of changes from an existing one, the actual product structure is the starting point and the new product structure is the final result.
The result is reached through a number of block property modifications. For example, we suppose that the user wants to modify the central agitator (referred as drum) volume. The product analysis has shown that changing this parameter means a strong impact on the washing machine. In fact if the user wants to increment this value, he wants to wash a bigger quantitative of clothes and then the washing machine will need a different dynamic equipment, a different motor, a larger amount of water and soap. As a consequence of this change, we expect the software tool to invalid a lot of relations and the change propagation to be very wide.
At the end of the change propagation figured in this example, the system produces the report represented in the figure 5 in which are listed the relations invalidated due to the change and their amount.

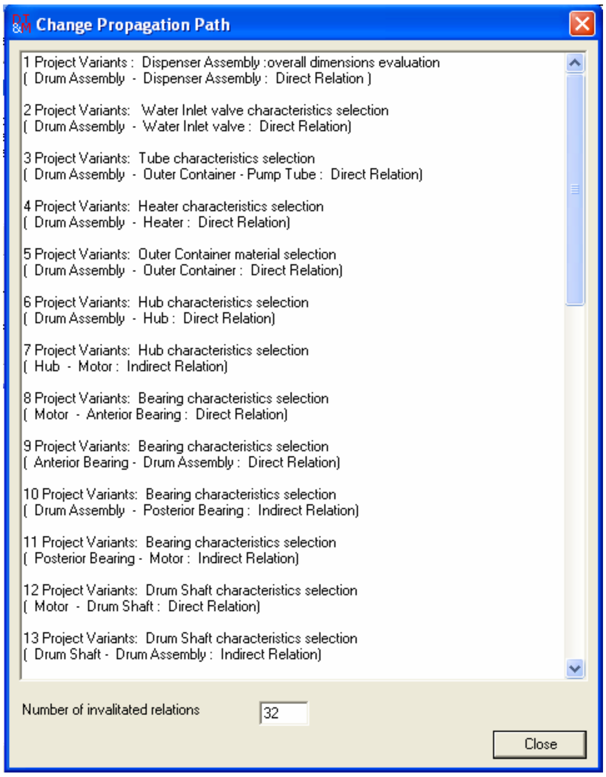

Figure 5. Change impact analysis: the report

Each paragraph of the report is composed by the name of the relation which is no more valid, the value users has given to it, the connected components and the direction by which is possible to understand which of the two blocks constrains the other.

The software system gives also the possibility to user to have a graphical representation of the change propagation path by changing in red the color of the invalidated relations in the structure network as in figure 6.

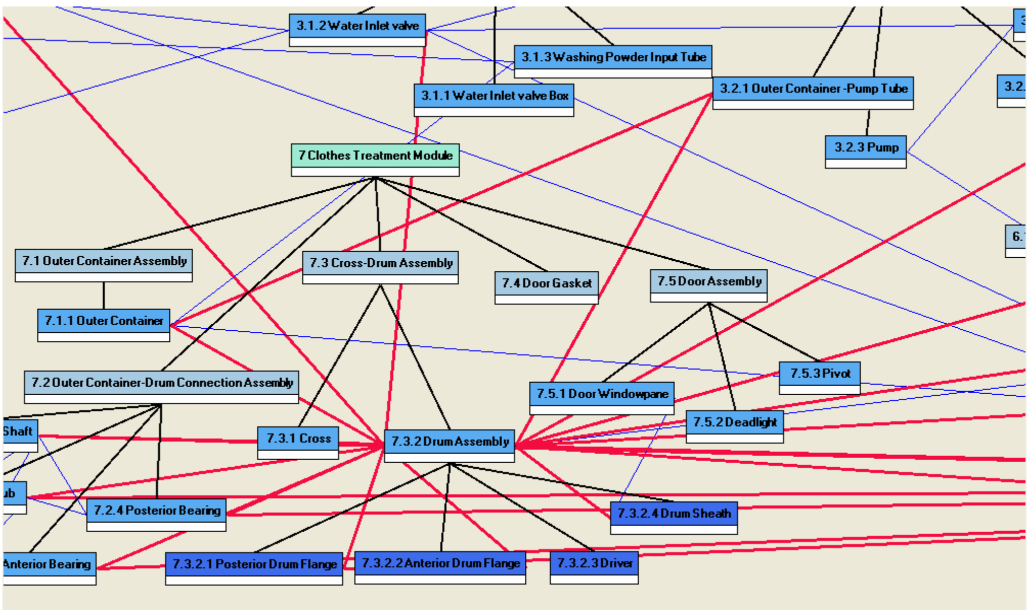

Figure 6. Change impact analysis: the change propagation path

Further analysis tools can be used to evaluate the change impact. We have tested our software in past washing machine changes with the help of company engineers. During the tool validation activity engineers have realized that the model could have been of use in many of the change situations encountered by the company. They have noticed that the change propagation paths obtained using the software and the product architecture we have identified, was very similar to the redesign activities they have had to do to implement these changes. They have also evaluated as very interesting the possibility to interface this tool with other decision support system, such as CAD, PDM, PLM, etc. In addition several designers have commented that the tool makes them consider their product in a different light, as they said "more rational but not sterile" and that they have found in it a perfect balance between functionality and usability.

5 CONCLUSIONS AND FUTURE DEVELOPMENTS

This paper introduces an innovative method to identify the linkages between components and assemblies within a product in order to gain a better understanding of change and potential propagation paths. However, the method has further potential uses because it can correctly represent the product functional structure and the product architecture as well as the relations between its parts. In addition it can represent a database of the product information and then designers can take advantage of this information that normally is time consuming to retrieve because implicit.

The experimental work has been performed on washing-machines design. Several architectures have been implemented using the software tool. We have used different products as examples of variability and source of changes. The developed system has produced propagation graphs and reports of relations between parts subject to changes. Currently we are working on the experimental results to define criteria for correlating the system output with the indices of impact evaluation (cost, time, etc.)

The preliminary results show that the approach is a valid support to evaluate the impact and possibly the cost of a modification, it provides a roadmap of the operations to perform and therefore it can help engineers to choose the correct product configuration among different design alternatives: predicting the paths of change propagation can also give aid to manage the design process and increase product reliability.

Our final purpose is to help the designer to think about product in a logic way putting evidence on the relations among components. It does not mean to make the design process sterile or extremely practical but to create a logic base from which an engineer can start to develop his creativity without forgetting the practical constraints of the product.

REFERENCES

[1] Eger T., Eckert C.M., Clarkson P.J. Towards a change process planning tool. In *CD-ROM Proceedings of International Conference on Engineering Design, ICED '03*, Stockholm, August 2003

[2] McAdams D., De Crescenzo A., Stone, R. Visual representation as an aid to concept generation. In *CD-ROM Proceedings of International Design Engineering Technical Conferences-CIE 2006*, Philadelphia, September 2006

[3] Flanagan T. L., Eckert C. M., Smith J., Eger T., Clarkson P. J. A functional analysis of change propagation. In *CD-ROM Proceedings of International Conference on Engineering Design, ICED '03*, Stockholm, August 2003

[4] Germani M., Mandorli F. Self-configuring components approach to product variant development, *AIEDAM - Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Special Issue: Platform Product Development for Mass Customization*, 18, 1, pp. 41 - 54, 2004

[5] Huang G.Q., Yee W.Y. and Mak K.L. Current practice of engineering change management in Hong Kong manufacturing industries. *Journal of Materials Processing Technology*, 2003, 139, pp. 481-487

[6] Jarrat T., Eckert C., Clarkson P.J. and Schwankl L. Product Architecture and the propagation of engineering change. In *Proceedings of International Design Conference, DESIGN 2002*, Dubrovnik, May 2002, pp 75-80.

[7] Chen K.M. and Liu R. J., Interface strategies in modular product innovation. *Technovation*, 2005, 25, pp. 771-782

[8] Fixson S. K. Product architecture assessment: a tool to link product, process, and supply chain design decisions. *Journal of operations management*, 2005, 23, pp. 345-369

[9] Lindemann U., Kleedorfer R. and Gerst M. The development department and engineering change management, in *Designers: The Key to Successful Product Development*, pp. 169-182, 1998 (Frankenberger E., Badke-Schaub P. and Birkhofer H., Springer-Verlag, London).

[10] Jarrat T., Eckert C. and Clarkson P.J. Development of a product model to support engineering change management. In *Proceedings of Tools and Methods for Competitive Engineering conference TMCE 2004*, Lausanne, April 2004, pp 331-342
[11] Keller R., Eger T., Eckert C.M. and Clarkson P.J. Visualising change propagation. In *CD-ROM Proceedings of International Conference on Engineering Design, ICED ’05,* Melbourne, August 2005
[12] Dahmus J., Gonzales-Zugasti J., Otto K. Modular product architecture. In *Proceedings of ASME 2000 Design Engineering Technical Conference DETC'00,* Baltimore 2000
[13] Simpson T. Product platform design and optimization: status and promise. In *CD-ROM Proceedings of ASME 2003 DECT/CIE Conference,* Chicago, Illinois, September 2003
[14] Ishii K., Yang T.G. Modularity: international industry benchmarking and research roadmap. In *CD-ROM Proceedings of ASME 2003 DECT/CIE Conference*, Chicago, Illinois, September 2003
[15] Germani M., Bonaventura L., Raffaeli R., Mandorli F. Development of a collaborative product development tool for plants design. In *CD-ROM Proceedings of International Conference on Engineering Design ICED '05,* Melbourne, August 2005
[16] Fujita K. Product variety optimisation under modular architecture, *Computer Aided Design,* 2002, 34, pp.953-965
[17] Stone R., Wood K., Crawford R. A heuristic method to identify modules from a functional description of a product. In *Proceedings of ASME 1998 Design Engineering Technical Conference DETC/DTM,* Atlanta, Georgia, September 1998

Contact: Roberto Raffaeli
Polytechnic University of Marche
Department of Mechanics
via Brecce Bianche
60021, Ancona, Italy
Tel.: +39-712204797
Fax.: +39-712204801
E-mail: r.raffaeli@univpm.it
URL: http://www.dipmec.univpm.it/disegno