INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED03 STOCKHOLM, AUGUST 19-21, 2003

A METHODOLOGY TO SUPPORT DESIGNER CREATIVITY DURING THE CONCEPTUAL DESIGN PHASE OF INDUSTRIAL PRODUCTS

F Bruno, F Giampà, M Muzzupappa and S Rizzuti

Abstract

In the recent past several studies have been conducted, and several methodologies have been set up to aid the designer in product design development. Another useful development in this field would be tools able to support the designer when he/she is manipulating abstract elements that roughly approximate their final shape and placement in the definitive layout.

The paper describes a methodology and a prototype software to support the designer during the conceptual phase. It is based on the creation of a 3D environment, the "design space", where the functional representation of the problem, that the designer has outlined, is increasingly clarified and solved in an architectural lay out of a product.

Keywords: Early phases of design, Conceptual design, Creative design.

1 Introduction

Conceptual design is the crucial phase in the activity of product design and in the whole product development cycle. Although CAx systems are widely used in many phases of product development (modelling, analysis, simulation, optimisation), they are not employed in the conceptual phase. Although several attempts at this have been made [1-11], it is difficult to support conceptual design with software.

The present paper proposes an approach to conceptual design, supported by a CACD tool, that starting from a functional representation of a product allows the designer to devise a 3D layout, assembling existing components and trying several alternatives in order to generate new possible solutions. The whole time the designer operates in a so called "design space": a 3D environment where he/she initially places functional blocks connected by links, that develop from conceptual forms to real 3D objects. This sequence in concordance with the top-down approach, can be easily reversed and a bottom-up sequence can be followed. Both approaches are attractive, in that new ideas can arise in unforeseen ways, and the designer must have the possibility to reappraise everything that has already been developed and make changes in the most flexible way.

The paper focuses on the generation of the “design space”, in which the functional nets are continuously updated. The meaning given to functional blocks and links, already present in other approaches [12-16], will be clarified. The links that at the beginning connect functional blocks, when these are transformed into real components, are directed to their functional surfaces. The analysis of the consistency of two functional surfaces connected by a link allows the designer to check the presence of critical points and, select the most appropriate solution.

Software, written in C++ employing the TGS Open Inventor 3.0 library, has been developed to support the decisions that the designer has to take during the design activity. The tool, that does not curb the generation of concepts, reminds the designer about the state of the current design conception and the problems that still have to be solved. The user works in an easy to use 3D environment where he/she can generate the product layout managing the visual representation of physical components and functional blocks. The components can be chosen from a set of predefined objects or imported in VRML format. It is possible to associate to every object, whether abstract or physical, a data set containing the information defined during the design process. In addition, the software helps the user to define the connections between 3D objects at every level of abstraction, and, on the basis of information related to functional surfaces, facilitates the assembly phase of the components.

The last section of the paper reports an example where the method is employed to study the functional aspects of a screw-driver and its subsequent fulfilment in an engineered solution. Even if the test was carried out on a well known device, it illustrates how the method and the tool can be employed. The tool has been conceived as an aid for the designer during the conceptual phase, to support him/her in a manner that he/she feels free to choose and try out any old or new idea.

2 Conceptual Design - The state of the art

At present, a universal approach to conceptual design does not exist, and this is a positive thing. To resolve engineering problems, the designer must be technically experienced and must also have common sense and, where necessary, a developed aesthetic sense. Nevertheless, in the last few decades, without taking individual competence into consideration, several studies have been conducted and several theories and approaches elaborated in order to improve this phase. The reason for this were the time reduction for product development, to ensure the identification of the essence of a product, and to support team or collaborative work.

2.1 Methodologies

Commonly the conceptual phase is set out in the steps of generation of ideas and selection of the main solution to be developed. Researches have proposed different methodologies to support the creative steps of the conceptual design phase: the generation of solutions.

The suggested approaches, for the most part, can be classified into two types: knowledge based and functional approaches.

Knowledge based approaches [1-7, 10, 17] allow the users to generate trial solutions first, supplying information about components. These systems provide a record of known solutions, frequently presented by morphological charts [18], which can be a stimulus to generate new, alternative ideas, even if based on existing examples. In these the “reuse” of solutions derived from previous designs is the recurrent theme.

Functional approaches are oriented toward the identification of the functions that a device has to perform. Recently, in [15], the functional approaches have been classified into paradigm [19] and systematic types. This latter was introduced by Palh and Beitz [12] and extended by others [13, 14, 20, 21]. However, both do not guarantee the generation of new concepts because, generally, a new solution can be reached as a combination of already known solutions. In fact, the solution can be searched for decomposing the function into sub functions and a solution is achieved collecting the sub solutions for each sub problem into which the problem has been decomposed.

The attempt solutions in [22], that try to integrate the functional and knowledge-based approaches, is interesting. A conceptual environment has been devised where a solution materializes, starting from its first representation in functional blocks, substituting these with real objects. Following this a knowledge system helps the designer for dimensioning, selecting materials, etc., by extracting information from a data base of similar cases.

Another approach is the Russian Theory of Inventive Problem Solving, TIPS or TRIZ [23]: a design method that identifies system conflicts and suggests solutions.

Other research [24, 25] proposes methods able to capture the designer intent from 2D sketches that are transformed in 3D objects and producing product layouts .

2.2 Tools

The theoretical research in conceptual design has been followed by a set of software prototypes able to support the designer in his/her work.

Among the knowledge based systems, CDIS [1] (Conceptual Design Information Server) has been designed to support concurrent engineering activities throughout the conceptual design phase, allowing the designer to access a set of stored experiences at different levels of abstraction. “Concept Generator” [2] is a tool that identifies potential components for a new product manipulating a series of matrices, starting from a functional description of the problem. Similar are the software tools ICEDMP [3], that, moreover, allows the designer to make DFX considerations and CADET [4, 5], that allows the user to evaluate possible solutions and then to select the best design choice.

Schemebuilder [6, 7] is a design tool aimed at guiding designers approaching problems from a functional standpoint and providing rapid access to relevant information, assisting in the production of specifications, producing preliminary lay outs and models for simulations etc.

ICM [8] provides a shared graphical modelling environment for collaborative design. Co-Designer [8] is a framework that supports localised design agents in the generation and management of ideas in the conceptual design phase.

Model Enhancer [9] is able to capture the knowledge in CAD models, to manage it and therefore make it reusable at a later date.

The aim of PROSECCO [10] is to provide access in a structured way to all the design information to facilitate conceptual design activities. The system orders all the available data, beginning from the automated preparation of the requirement list.

The WeBid system [11] is addressed at supporting the involvement of suppliers in new product development.

The software tools NODES [8] and DOME [8] allow the expression of knowledge in mathematical terms by associating numerical relationships.

3 A tool for conceptual design

According to different approaches in literature [12-14], the functional approach to conceptual design, described in this paper, is based on the integration of the classic top-down and bottom-up approaches, by which a product is studied as a set of functional elements (functional blocks) properly connected in order to form a so-called "functional net".

The draft of the functional structure of the design generally does not follow a rigid framework because designer creativity cannot be constrained within defined and structured schemes. Therefore, because the embodiment process is not unequivocal, the designer should have freedom of action. In other words, the definition of the functional structure should not be exclusively tied to the top-down or the bottom-up sequence, but must rather be an activity that integrates these two strategies. In this way it is possible to describe in more detail further the first concept (top-down approach) in order to obtain the final description of all components; otherwise, once a functional structure has been created, it is possible, if necessary, to review the decisions taken and to group single components in blocks that could be used in a wider functional context (bottom-up approach).

Because the final goal of the design is the definition of a geometric three-dimensional model, a 3D environment has been developed to support our functional approach.

In this 3D environment, different "graphic primitives", specifically to aid the designer in the conceptual phase, are available. He/she can use these primitives to represent both functional blocks and standard mechanical components.

At the end of the modelling phase, the design space is full of geometric shapes that represent possible solutions of the product.

3.1 Graphic Primitives

A functional block is a kind of black box that has a different meaning depending on the current level of detail in the design process. In the initial phase, a functional block represents a specific function derived from an analysis of the functional requirements. In the final definition phase each functional block is substituted by a single component or group of components that are put in relationship with the other blocks. Decomposing the conceptual schematisation in blocks, a higher level of detail can be gradually obtained. During this process the functional blocks change both their semantic meaning (from a concept-function to a constructive solution) and their shape. In fact when a functional block becomes a real component, its graphic representation changes from the initial block (Figure 1/a) into a shape that represents the component, a so-called "archetype" (Figure 1/b). All the predefined archetypes are collected in a library.

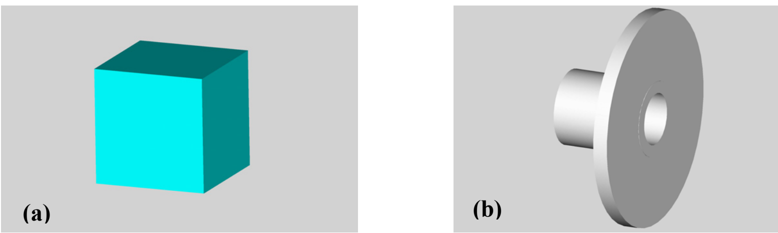

Figure 1: Graphic Primitives: a) functional block; b) The archetype of a pulley

3.2 Links and functional net

In the proposed approach the interactions between functional blocks are represented by links. Four type of links have been used, each one characterised by a different colour (red, yellow, green and blue) that represent respectively the energy, material, signal and force flow (Figure 2/a).

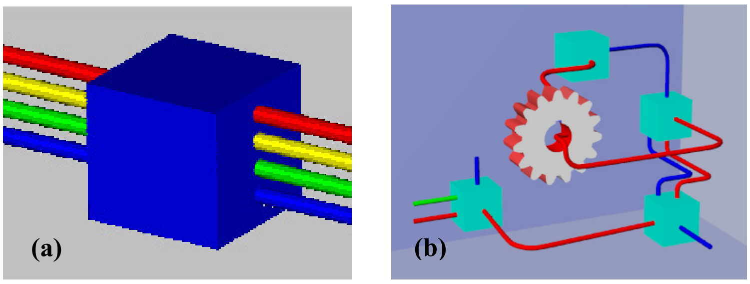

Figure 2: a) A functional block with all possible links, b) An example of a functional net

In Figure 2/b a functional net is represented. The net is composed of a certain number of functional blocks and predefined components that are connected by the links. In other words the functional net represents a possible product layout defined by functional blocks, archetypes and links. The designer employs the functional net in order to create and evaluate different product solutions.

3.3 Functional surfaces

When a possible constructive solution has been defined, a functional block "evolves" into an archetype of mechanical component to which a specific geometric model corresponds. The archetype is a smart object characterised by a set of semantic data about its functionality and interaction with the other archetypes.

On every archetype, all possible contact surfaces with other archetypes are identified. These "functional" surfaces are characterized by a geometry (plain, cylindrical, conical, either internal or external) and by the degrees of freedom locked through the assembly. The set of functional surfaces give the archetype the same behaviour as the real component in the device.

In figure 3/a, the archetypes of two mechanical parts (shaft and gear) are shown, where the functional surfaces (cylindrical for the joint and planar for the shoulder) are displayed by the suitable colour. In figure 3/b the single parts are assembled because the compatibility of the functional surfaces has been verified.

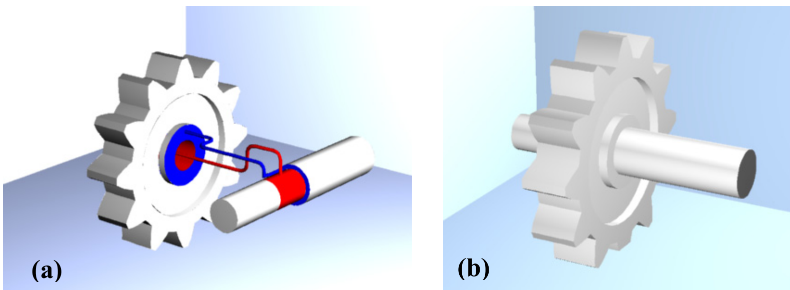

Figure 3: a) The archetypes of shaft and gear with functional surfaces. b) The assembly of two components.

4 The first prototype of the software

A prototype of the software has been implemented. The software is based on a framework in which the designer opens many 3D windows used to design the product at the various levels of abstraction. Each window can contain both functional blocks and detailed components.

4.1 The design space

The software is a multi document application. This means that the user can open different documents in the same application. Every document has its own window that represents the 3D scene in which the user can add and manipulate 3D objects that represent both the functional blocks and the archetypes chosen from a library. The user can navigate in the scene using either the mouse in conjunction with the Alt and Ctrl keys or three sliders placed on the window frame. The user can also use some other navigation tools that allow him/her to reset the viewpoint to the default position, to bring all the objects into view, to move the camera to see a specific detail etc. Other standard functions allow the user to change the lights on the scene, edit the object colour and material, set the type of visualization (shaded, wire frame, hidden line etc.) and the type of the camera (perspective or orthographic) etc.

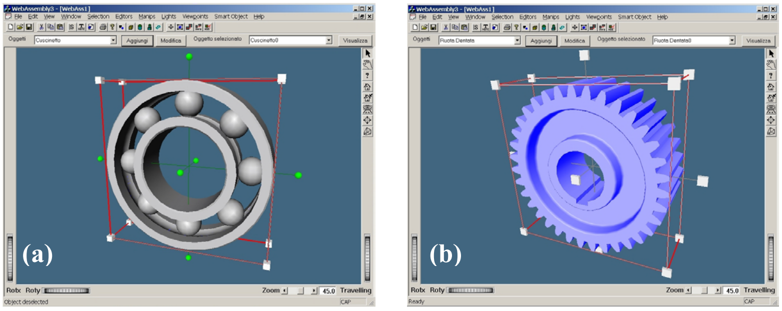

Figure 4: The two interfaces for manipulating the object in the design space

The objects, both the functional blocks and the archetypes, can be rotated, translated and scaled in the scene using two types of manipulators. A manipulator is a 3D interface made of some items (lines, cubes and spheres) that the user can drag to perform a specific transformation. The manipulator shown in Figure 4/a is used to translate the object on each of the 3 planes defined by the bounding box of the object, to rotate the object dragging the green spheres and to scale uniformly the object dragging the white boxes. In Figure 4/b the manipulator used to scale the objects either uniformly or along a single axis is shown.

4.2 Building a functional net

The draft of the functional structure of the product can be built employing the functional blocks and the links that connect them. The functional blocks are represented as 3D boxes that can be transformed using the manipulators and are characterised by a name and a description assigned by the user. The links are represented as pipelines where the colour indicates the type of the link (energy, force, signal or material flow). When the user creates a link he/she has to specify the type of the link and can assign it a description. In any case, the data about the blocks and the links can always be modified. Each functional block can be detailed in a new empty window. When the user finishes modelling it in the new window he/she can close it and return to the previous window. Then the user can explode the content of the functional block inside the functional net. In other words the content of the new window is inserted in the previous one substituting the corresponding functional block. If the user does not explode the functional block, this will be represented as a semitransparent box containing a simplified representation of the new window. The user can duplicate the functional net in order to preserve the representation with the functional blocks and the links. In this way, the copy of the scheme can be used to explode the functional blocks in order to assemble all the detailed components (used to specify the functional blocks) together and to build an approximated lay out of the product.

Currently the main limitation of the software is related to the lack of a proper method to support the bottom-up approach described in the model. In fact, if the user modifies something at a lower level of abstraction, the corresponding upper levels are not automatically updated.

4.3 Building a constructive solution

The software has a library of predefined components (archetypes) that the user can employ in the product design. The components are smart objects created on the basis of the definition reported in section 3.3.

The assembly of components can be performed automatically when functional surfaces, related to different components, recognise each other. When the user moves a smart object near another one, if they have a set of compatible surfaces, he/she is informed of the possibility of assembling these components. If the user chooses to link the objects with these surfaces the moving object is snapped in order to allow contact between the two surfaces (figure 3/b). Moreover, in some cases, the system asks the user which way he/she wants to connect the two surfaces, suggesting some possible solutions and giving some information for choosing the best one. It might be necessary that the two surfaces need some "devices" to establish the connection. For example to connect a shaft to a gear the user can adopt a splined profile, a key, etc.

4.4 Creating an archetype

In the current version of the software there is also the possibility to create new archetypes which can be added to the predefined library. The interface for creating the predefined components was initially intended only for use by the software developers. Currently we are evaluating extending this interface to also allow the user to add new predefined components to the library.

The first step for creating a new archetype is to model its shape in a CAD system able to export the model in VRML or Inventor file format. Then, the model has to be loaded in the software and a name has to be assigned to it. The next step is the definition of the functional surfaces. These can have three geometrical shapes: planar, cylindrical or truncated conical. The geometrical shape of the surfaces defines in which way the snap will be made, as explained in Table 1. In order to identify a functional surface on the object the user has to declare the type of surface and pick a point on it with the mouse. For each functional surface the developer also has to specify the compatibility with the surfaces of the other objects and, eventually, insert knowledge about the connection.

Table 1. Rules controlling the snap between functional surfaces

Surface	***Snap – How***	***Snap - When***
Planar	the centres and the normals of the planes have to mate	The distance between the two plane centres and the angle between the normals are lower than two fixed thresholds
Cylindrical	the axes have to align	The angle between the axes and the minimum distance between them is lower than two fixed thresholds
Truncated Conical	the generatrixes have to coincide	The angle between the generatrixes and the minimum distance between them are lower than two fixed thresholds

5 An example

In order to better explain the advantages of the methodology described in the previous sections, a simple example will be reported in which the software prototype is employed to design an electric screw-driver.

In Figure 5/a the first functional net is shown, containing only one functional block that represents the entire product. The links with the outside are the contact with the user's hand, the contact with the screw and the activation signal. As shown in Figure 5/b this functional block is detailed in a possible functional net. The links with the outside have not been changed, and new links have been added to establish the connection between the new internal functional blocks.

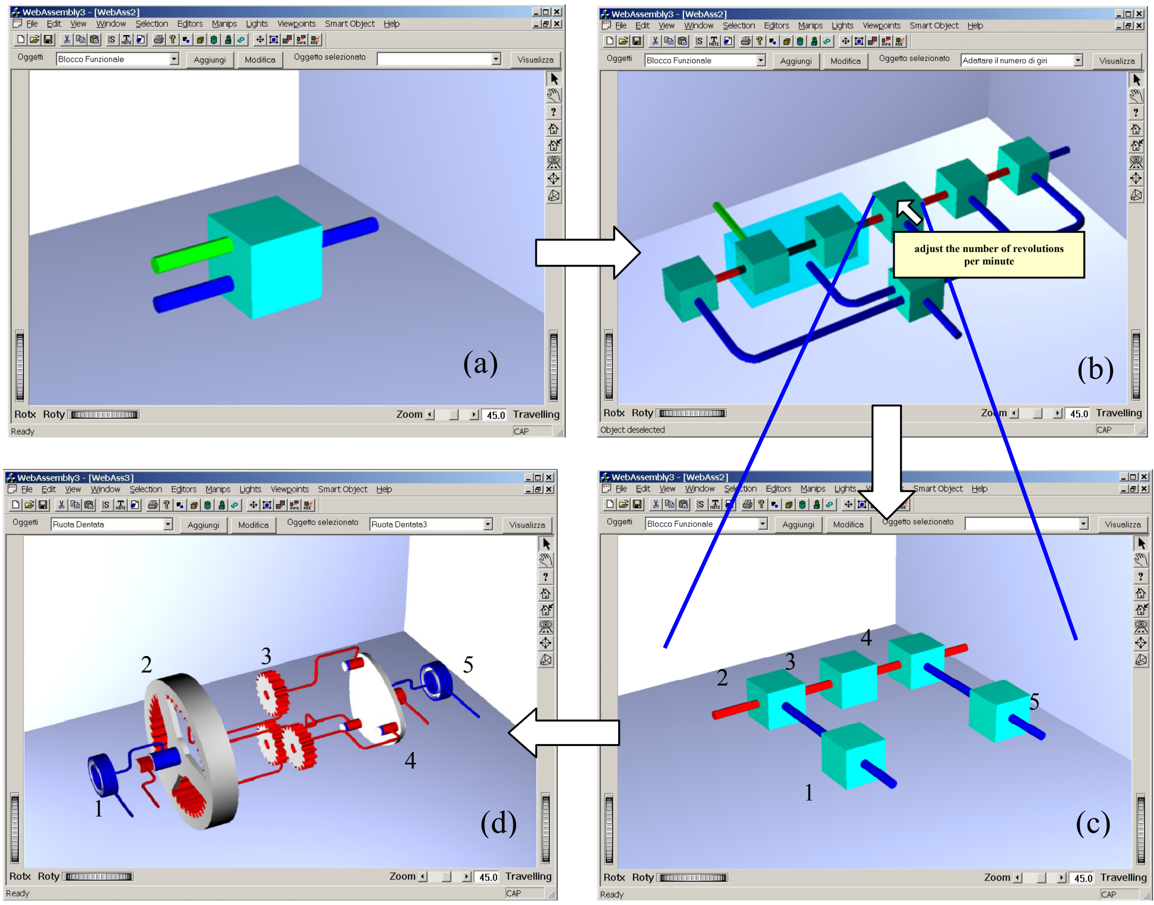

Figure 5: a) First functional block. b) Functional net of an electric screw-driver. c) Functional net of the "adjust the number of revolutions per minute" function. d) Functional blocks are substituted by component archetypes

Each block in the design space is identified by a name that appears in a callout when the pointer touches it. The semitransparent cyan block represent a multifunction block, that can be further detailed using other blocks. For our example, only the specification of the "adjust the number of revolutions per minute" function has been shown (Figure 5/c). In any case the other blocks are detailed in the same way. When the functional net has reached a sufficient level of detail, the functional blocks are substituted by the corresponding predefined components chosen from the library (Figure 5/d). The functional surfaces of the components have a specific colour according to their nature and the links that connect with them. Now, the user has to scale and move each component in order to define an approximate layout of the product. In this phase he/she is supported by the surface snap that allows an easy assembly of the components.

6 Conclusions

The methodology, described in the paper, has been tested on several cases. It offers an aid for the conceptual design phase, integrating the top-down and bottom-up approach. In this way the creativity of the designer is not constrained in a too rigid and structured scheme. The software offers the designer the possibility to integrate standard components, abstract functional blocks and their relative data in a single 3D design space.

In the near future the software will also be extended to support the bottom-up approach of the model, by implementing a method that automatically maintains the consistency between the functional net at the various levels of abstraction and the assemblies. The approach consists in the automatic construction of a 2D graph that contain both the functional net at the various level of abstraction and the detailed representation of each functional block. In this way the user has an explicit visualization of the design history and, for each functional net, he/she can have a parallel representation containing the detailed components.

References

[1] Wood III W.H. and Agogino A.M., “Case-based conceptual design information server for concurrent engineering”, Computer-Aided Design, Vol. 28, 1996, pp. 361-369.

[2] Strawbridge Z., Mc Adams D.A., Stone R.B., “A computational approach to conceptual design”, Proc. of DETC02, ASME 2002 Design Engineering Technical Conference, Montreal, 2002.

[3] Chen K.H., Chen S.J., Lin L., Changchien S.W., “An integrated user interface (GUI) for concurrent engineering of mechanical parts”, Computer Integrated Manufacturing System, Vol. 1, 1998, pp. 91-112.

[4] Rodgers P.A. and Huxor A.P., “The role of artificial intelligence as ‘text’ within design”, Design Studies, Vol.19, 1998, pp. 143-160.

[5] Rodgers P.A., Huxor A.P., Caldwell N.H.M., “Design support using distributed Web-based AI tools”, Research in Engineering Design, Vol. 11, 1999, pp. 31-44.

[6] Bracewell R.H., Bradley D.A., Chaplin R.V., Langdon P.M., Sharpe J.E.E., “Schemebuilder, a design aid for the conceptual stages of product design”, Proc. of ICED93 International Conference on Engineering Design, The Hague, 1993.

[7] Sharpe J.E.E., “Computer tools for integrated conceptual design”, Design Studies, Vol. 16, 1995, pp.471-488.

[8] Wang L., Shen W., Xie H., Neelamkavil J., Pardasani A., “Collaborative conceptual design – state of the art and future trends”, Computer Aided Design, Vol. 34, 2002, pp.981-996.

[9] Wang F., Mills J.J., Devarajan V., “A conceptual approach managing design resource”, Computers in Industry, Vol. 47, 2002, pp. 169-183.

[10] Frankenberger E., “Computer-supported systematic design and knowledge management in the early design phase”, Proc. of ICED01 International Conference on Engineering Design, Glasgow 2001, pp. 115,122.

[11] Huang G.Q. and Mak K.L., "WeBid: A web-based framework to support early supplier involvement in mew product development", Robotics and Computer Integrated Manufacturing, Vol. 16, 2000, pp. 169-179.

[12] Pahl G. and Beitz W., "Engineering Design– a systematic approach", Springer, London 1996.

[13] Ulrich K. and Eppinger S., "Product Design and Development", McGraw-Hill, New York, 2001.

[14] Cross N., "Engineering Design Methods", John Wiley & sons, New York, 2000.

[15] Chakrabarti A. and Blight T.B., "A scheme for functional reasoning in conceptual design", Design Studies, Vol. 22, 2001, pp. 493-517.

[16] Stone R. B. and Wood K., "Development of a Functional Basis for Design", Journal of Mechanical Design, Vol. 4, 2000.

[17] Wood W., Yang M., Cutkosky M., Agogina A., "Design Information Retrieval: Improving Access to the Informal Side of Design," Proc. of DETC98/DTM-5665, ASME1998 Design Theory and Methodology Conference, 1998.

[18] Chiou S.J. and Kota S., "Automated conceptual design of mechanisms", Mechanism and Machine Theory, Vol. 34, 1999, pp. 467-495.

[19] Yoshikawa H., "General design theory and a CAD system", IFIP Man-machine communication in CAD/CAM 1981.

[20] Suh N., "The Principles of Design", Oxford University Press, 1990.

[21] Hubka V. and Eder W.E., "Engineering Design", Heurista, Zürich, 1992.

[22] Terpenny J. P., Nnaji B. O., Bohn J. H., "Blending Top-Down and Bottom-Up Approaches in Conceptual Design", Proc. of 7th Annual Industrial Engineering Conference, Banff, Alberta, Canada, May 1998.

[23] Altshuller G., "Creativity as an Exact Science," Gordon and Branch Publishers, Luxembourg, 1984.

[24] Rodgers P.A., Green G., McGown A., "Using concept sketches to track design progress", Design Studies, Vol. 21, 2000, pp. 451-464.

[25] Qin S.F., Wright D.K., Jordanod I.N., "From on line sketching to 2D and 3D geometry: a system based on fuzzy knowledge", Computer-Aided Design, Vol. 32, 2000, pp. 851-866.

Sergio Rizzuti
Professor
Dept. of Mechanical Engineering
University of Calabria
Ponte Pietro Bucci 44/C
87030 Rende (CS), ITALY
TEL: 39-0984-494601 FAX: 39-0984-494673
E-mail: rizzuti@unical.it