Rational Approach for Creative Products Design

Waleed El-Ganzoury[1] and Chun-ju Tseng[2]

[1]Dansco Dairy Products Ltd, wganzoury@dansco.co.uk
[2]Ling Tung University, ktseng@mail.ltu.edu.tw

ABSTRACT
When a client approaches a designer with a product need, it is unlikely that his/her need will be expressed very clearly. The client simply might have a problem that needs a solution. This paper will set up a rational/systematic methodology for creative designs, which delivers more customised products with shorter lead times, perfect quality and reliability and at lesser prices. The design method possess the capability to employ the product's overall required function provided by customer, to produce a set of embodiment designs that are capable of obtaining this function.

Keywords: Systematic design, modular products design, rational design, creative design

1. INTRODUCTION

It has been indicated from various studies that over 70% of the production cost of a product is determined during the conceptual design stage [1]. However, the design phase itself accounts for only 6% of the total product development cost [2-5]. Therefore, devoting greater effort during the design phase is a necessary step towards optimising product design.

Products design goes through a number of processes starting from clarifying the design objectives, establishing its function, setting the function attributes, determining the engineering attributes, generating the alternative embodiment designs that meet these specification, evaluating these alternatives, and finally developing the product detailed design [6, 7].

A knowledge based system has been developed in the Rapid Product Development Research Centre (RPDRC) at De Montfort University (DMU). The system possess the capability of developing product designs that follow the above mentioned procedures, the system is also integrated with a Computer Aided Design (CAD) solid modelling system which gives that developed system the capability of demonstrating the developed detailed design. The system is also integrated with the material selector software and a finite element analysis package to facilitate the detailed design process within the developed design method [8]. In this paper the procedure of developing alternative product designs will be explained.

2. CONCURRENT CONCEPTUAL/EMBODIMENT DESIGN

The main principle of the developed design approach is that it assists inexpert designer to use experts' design knowledge to develop creative, original designs. Design knowledge has to be collected. This knowledge is a combination of expert designers' knowledge, which could be obtained by analysing existing designs. This knowledge is simply a set of modules, their functions, and their assembly ability.

The overall function of a product is expressed in terms of its functional input and output, and could be denoted as the overall design problem, see Fig. 2. Based on Fig. 1 in a rational design method, each overall problem has an adequate overall solution and both of them are composed of number of sub-problems and sub-solutions respectively. The overall solution is the required product's embodiment design, and the sub-solutions are the modules that this embodiment design is composed of. The functions of the modules that a design is composed of are the sub-problems, and the overall problem is the overall conceptual design. The overall problem is broken down based on modules' functions and these modules' assembly ability, structuring the overall solution – embodiment design-.

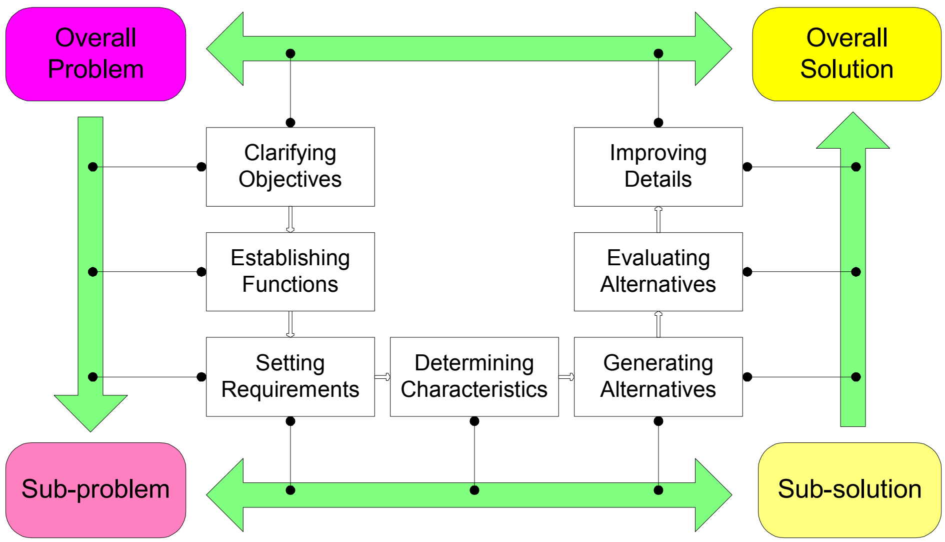


Fig. 1. Symmetrical Problem/Solution model [6].

In Fig. 3 the conceptual design is represented with the sets of inputs and outputs (*i*,*o*) and the blocks are representing the modules of the product.

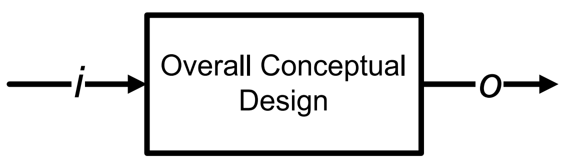


Fig. 2. Overall conceptual design.

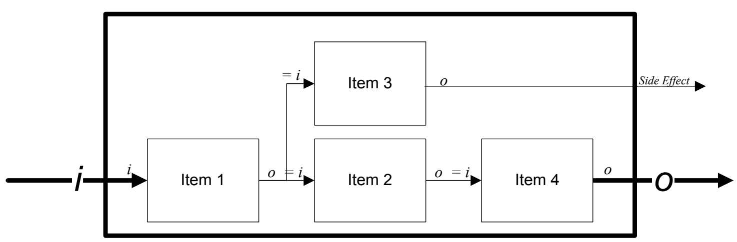


Fig. 3. Conceptual/Embodiment design.

This great design effort during the conceptual and embodiment design phases always occurs in modular products design, as they elaborate the corresponding function structure in a way to fulfil their overall functions through the combination of the distinct modules.

The modules generated by the developed design approach adjust their parameters (parametric/functional attributes) to fit with the current design problem, which is classified based on Ulrich and Tung [9] as Fabricate-to Fit modules. Four types of functions recur in the developed design approach and can be combined as sub-functions to fulfil different overall functions. These function definitions was originally obtained from Pahal, and Bitz [10]:

1. Basic function, which fulfils the overall function simply or in combination with other basic functions;

2. Auxiliary function, which is implemented within the developed design approach as a side effect of the basic modules. - No Auxiliary modules are defined in the form of (*i,o*);
3. Adaptive function, which is necessary to adapt the proposed design to other systems and to marginal conditions. Adaptive modules dimensions are not fully fixed in advance and hence allow for different circumstances;
4. Special functions, which if they present will present the side effects of the basic functions.

To facilitate design externalisation, a tree view should be structured to guide designers to the right, feasible, applicable functions. This would ensure that the designer could visualise what he/she really wants if his/her needs were quite vague. On the other hand the tree ensures that the selected required input\output functions have adequate modules that are capable of obtaining them, which in turn increase the chance that the requested product is applicable to be designed or generated by the available knowledge. Fig. 4 illustrates the tree view for precise definition of required functional input for designing a hairdryer.

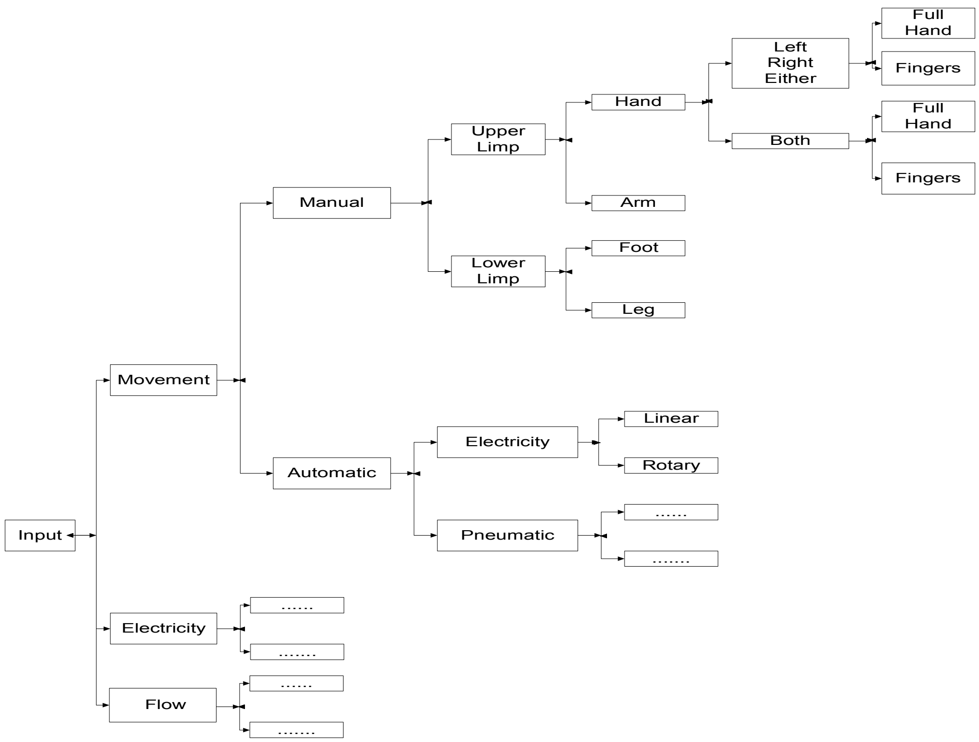


Fig. 4. Required functional input definition for designing a hairdryer.

2.1 Overall Function Decomposing Process

Reasoning within the developed design approach is based on the formulation of the developing conceptual/embodiment designs as a search. In this formulation the description of the desired design is the goal and the set of embodiment designs leading from initial conditions to the goal is viewed as the search space.

Design knowledge has to be collected. This knowledge is a combination of expert designers' knowledge, which could be obtained by analysing existing designs. This knowledge is simply a set of modules, their functions, and their assembly ability. Tab. 1 demonstrates an example of the required design knowledge.

Since the overall design problem provided by the user is the overall conceptual design in its format of product required functional input and output, and all the modules are also defined by their functional input and output. Therefore, the search process starts by looking for a module to function as an input module ($module_n$) for the required product, this module's functional input should match the required functional input requested by the client. The search process chains forward starting from the initial condition ($module_n$) through the knowledge domain based on the assembly ability data available for the designer, for the event ($module_n$, $module_{n+1}$) where ($module_{n+1}$) is another module that is applicable to be assembled to ($module_n$). Whenever this event is executed the functional output of $module_{n+1}$ is then checked, and if it is the same as the required functional output, this module is considered as an output module and the set of modules and their representative functions are considered as the embodiment and the conceptual design respectively. If the required functional output was not satisfied, the search process keeps assembling modules structuring the *function chain* of the product until the required functional output is satisfied.

	Input module	**Output Module**	**Side Effect**	**Additional Input**
1	Pedal	Wire		
2	Wire	Carburettor		X
3	Carburettor	Engine	X	X
4	Engine	Cultch		
5	Engine	Torque converter		
6	Engine	Exhaust	X	X
7	Cultch	Gearbox		X
8	Torque converter	Gearbox		X
9	Gearbox	Chain		
10	Gearbox	Propeller shaft		
11	Gearbox	Differential		
12	Propeller shaft	Differential		
13	Chain	Two wheels		X
14	Differential	Four wheels		X
15	Air cleaner	Engine	X	X
16	Battery	Switch		
17	Switch	Ignition coil		
18	Switch	Bulb		
19	Ignition coil	Distributor		
20	Lever Control	Gearbox		X
21	Fuel Tank	Fuel Pump		
22	Fuel Pump	Carburettor		X
23	Steering wheel	Steering shaft		
24	Steering shaft	Steering gear		
25	Steering gear	Four wheels		X
26	Distributor	Spark plugs		
27	Spark plugs	Engine	X	X

Tab. 1. Assembly ability knowledge.

A set of conditions was developed to prevent the search from going to close loop search. However, whenever the search reaches a dead end (no more modules to be assembled to ($module_n$) and the required functional output is not satisfied) the search process should move backward substituting ($module_n$) from the function chain, and searches for another ($module_n$) to be assembled to ($module_{n-1}$).

2.2 Branching Process

In complex products design, one module might demand more than one input, – additional input- or would produce more than one output, –side effects-. In this case the function chain starts to branch shaping a net of several inputs and outputs, (see Fig. 5)

After the whole chain is completed, and the overall required function is satisfied, the branching process is executed by checking all the modules in the chain, and if any of these modules produce an side effect, the system searches for the a side effect module that should be assembled to the main chain to facilitate this side effect. The search process then carry on assembling modules to side effect module in the same manner using the assembly ability knowledge until it reach to an output module, and the output of this output module is considered as a side effect for the whole product. On the other hand, if an additional input is required, the additional input module is obtained from assembly knowledge, and then assembles modules to it.

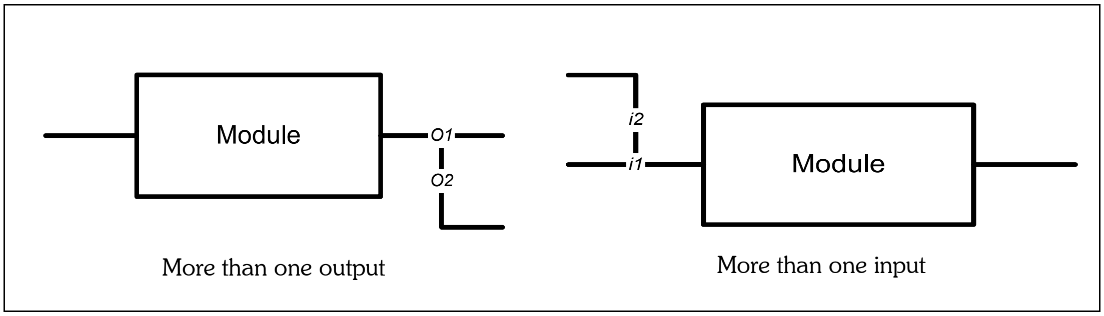
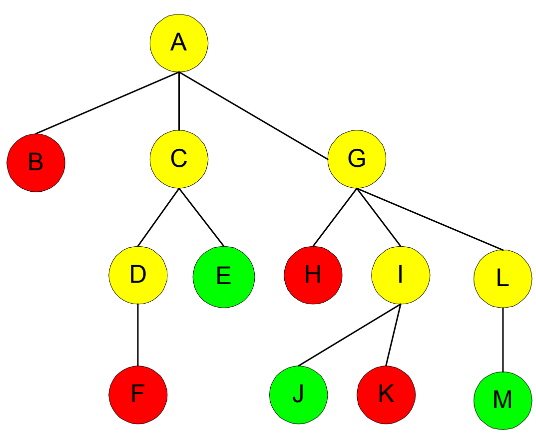


Fig. 5. Concept branching.

2.3 Reasoning Mechanism

The search process look for a module to function as an input module for the required product design ($module_n$), this module functional input should match the overall required functional input requested by the user. The search starts forwardly from the initial condition (input module) to the desired conceptual/embodiment designs in a data-directed approach. The assembly ability between the product modules must be set in advance from experts' knowledge, which guides the search process.



Search State	Search Results
A	Check B, Check C, Check G
B	Fail
C	Check D, Check F
D	Check E
E	Fail
F	Success
G	Check H, Check I, Check L
H	Fail
I	Check J, Check K
J	Success
K	Fail
L	Check M
M	Success

Fig. 6. Depth first search tree.

Considering the systematic order in which the nodes of the solution space maybe considered, the search method is depth-first searches, which consider the successive solutions before considering the alternatives.

Fig. 6 shows an example of the depth first tree search, which is obtained by the reasoning mechanism. Each node in the tree represents a module, and each line represents an assembly ability event between the two modules. States (A, B), (A, B, D), (A, G, H), and (A, G, I, K) are the states where the composed design does not satisfy the overall required function, since the functional output of modules B, D, H, and K do not satisfy the required functional output. State (A, C, F), (A, G, I, J), and (A, L, M) represents the design alternatives, where the overall required function is satisfied.

3. CASE STUDY

The aim of this case study was to demonstrate the capability of the system to generate complex function nets with their various inputs and outputs, based on the overall required function. A small prototype of the knowledge frames is implemented for this case study to narrow down the search space for simplicity. For paper size limit not all the modules incorporated in the product design and all additional inputs and side effects are considered in the cast study.

3.1 Product Specification

This is a design problem of designing a vehicle. The designer starts from the client's vague requirements formalising the product over all conceptual design (overall required function) describing it using a pair of input and output (*i-o*). The designer considered that the required input for the vehicle is the linear movement the driver apply to the accelerator pedal, and the required output is the movement front-back left-right with stability. The overall conceptual design of the vehicle obtained by the user is demonstrated in Fig. 7.

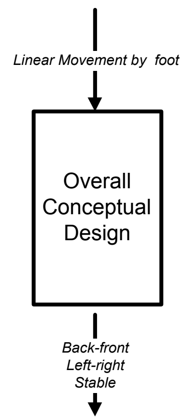


Fig. 7. Overall conceptual design.

3.2 Generating Alternatives

The search starts forwardly from the initial condition (the input module) to the desired conceptual/embodiment designs in a data-directed approach. The input module is a pedal that accepts the operator's foot linear movement. Tab. 1 illustrates the parts of the assembly ability farms, which is implemented as heuristic data for this search problem.

Stage 1:

This state is reached by executing assembly ability events 1, 2, 3, 4, 7, 9, and 13, (see Tab. 1). Fig. 8 illustrates the design structure of this state. The functional input of this design is the same functional input requested by the user (linear movement by foot), where it's functional output (movement front-back left-right with no stability) is different than the requested functional output.

These events are executing sequentially based on satisfying the assembly ability events. Where event 1 satisfies the functional input requested by the user, that both of them share that the same functional input is linear movement by foot. The output module of this event is the wire, which the next search process is based on. Event 2 is the next event to be executed since the value of its input module slot is the same as the output module of the composed design (the wire and this example it is the only event in this knowledge base for simplicity). In the same manure event 3 is executed. The output module of the composed design is the engine, which is the starting point of the next search process. Three events satisfy this search i.e. events 4, 5, and 6. However, event 4 would be the first event to be executed since it is a sequential search process. The output module of the composed design is the cultch, which could only executed by event 7 producing an output module of gearbox, which again could be executed either by event 9, 10, or 11. However, event 9 would be the first alternative to be executed. The output module of the composed design is the chain, which is only executed by event 13 producing an output module 'two wheels', which is an output module but does not satisfy the overall required function.

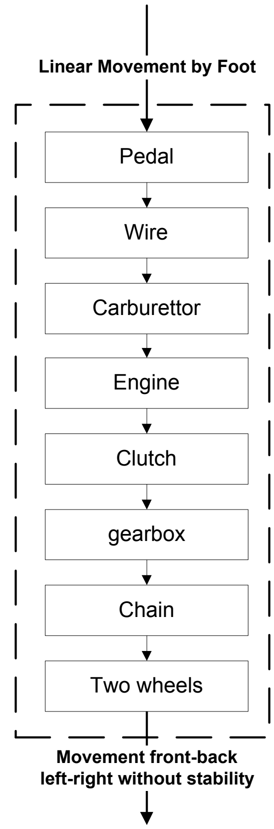


Fig. 8. First search stage.

Stage 2:
The first state is reached by executing assembly ability events 1, 2, 3, 4, 7, 10, 12, and 14, (see Tab. 1). Fig. 9a illustrates the design structure of this state. After the search process reached an end point without satisfying the requested functional output, it moves back ward substituting the last module been added to the search process, which was the two wheels, searching for another module that could be assembled to the chain to complete the search process. Where there is not another event in the knowledge base farms contain the chain in the input module slot, the chain is substituted as well from the composed design leaving the gearbox as an output module. After event 9 has been executed it is event 10 turn to take place, producing an output module of propeller shaft. After this event 12 and 14 are executed suggesting the four wheels as output module, which satisfies the required functional output, and the overall design satisfies the overall required function, making this state is the first suggested alternative.

After the overall required function is satisfied, another search process is executed to convert the produced function chain to a complete function net with its several additional inputs and side effect. Each module in this function net is checked and if additional input is recommended or a side effect is produced. From Tab. 1 the carburettor requires additional an input. Another search process is executed in the special knowledge frames to obtain the additional module to be assembled to the carburettor. After obtaining this module from the additional module frames, the search process is carried out in the assembly ability frames in Tab. 1 until it reach an input module, but in this case the search start by looking at the output slots to find the input module for the composed design. In the carburettor case the fuel tank is obtained to be an additional input to the carburettor. This is followed by executing event 22 where the fuel pump is the output module to the fuel tank. The whole net is illustrated in Fig. 9b.

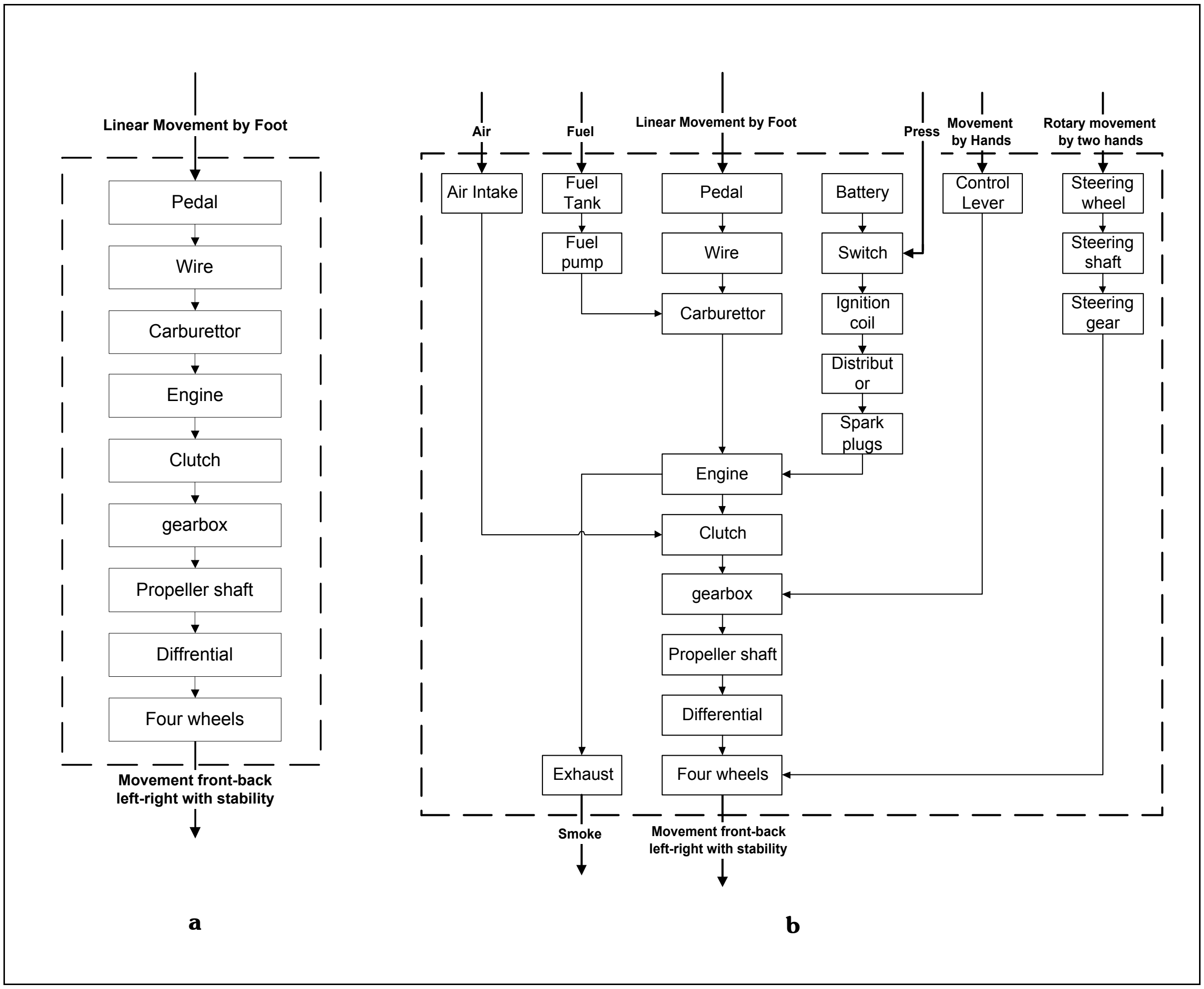


Fig. 9. First design alternative.

Stage 3

This state is reached by executing events 1, 2, 3, 4, 7, 11, and 14, (see Fig. 10 illustrates the design structure of this state. The proposed design satisfies the overall required function and full net is produced with its additional inputs and side effects. It is considered as the second design alternative.

For paper size limit the other four generated alternatives are not explained in details. However, Fig. 11 illustrates the search tree for this design problem search, where lined circles indicate successful alternative, and doted circles indicates failure alterative.

4. CONCLUSIONS

The developed design approach has been described in detail in this chapter. The knowledge base system roles were reviewed and the external software activities were represented. The developed system contains many original features, which together produces a novel approach for systematic product design, these are as follows;

- The recommendation of implementing a creative method for transforming Product Design Specification (PDS) to an overall product conceptual design. This enhances the chance of developing innovative designs, and gives the developed design method a generalised image, which is extremely necessary for a successful design method.

- The system uses the overall required function to systematically obtain the set of alternative designs that are capable of satisfying it.
- State of the art functional reasoning strategy, which is implemented by decomposing the product overall required function based on the function of the available appropriate modules and their assembly ability.
- The developed system defines the required product function net with its several inputs and outputs and set up the main modules that are capable of performing these functions.
- The system produces Fabricate-to Fit modules that are capable of obtaining basic, auxiliary, adaptive and special functions.
- The developed conceptual design fulfils the overall required function requested by the designer.

As a result, the developed design approach has the combination of the above unique features that have not been addressed previously by other approaches. These features provide a rational approach that supports creative product design.

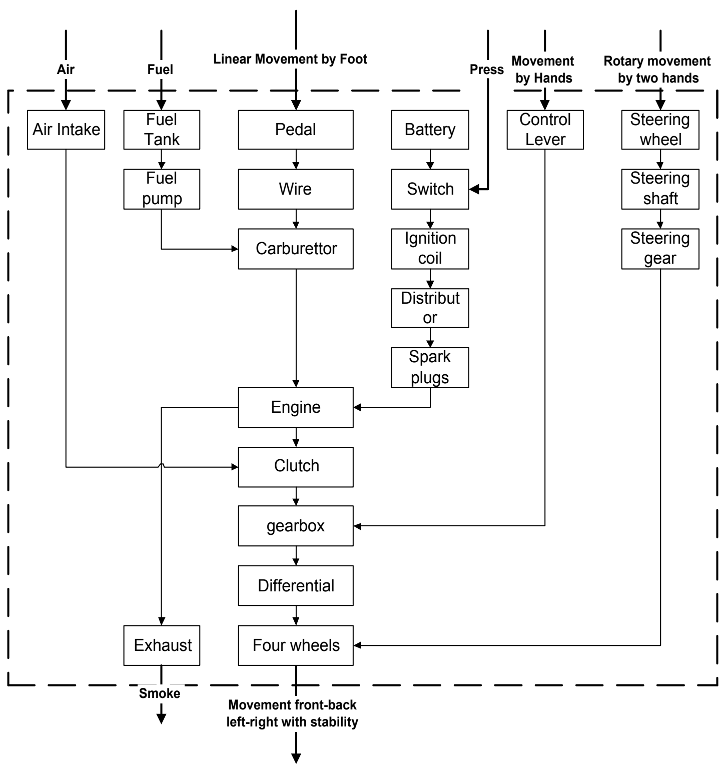


Fig. 10. Second design alternative.

5. REFERENCES

[1] Shehab, E. M. and Abdalla, H. S., A design to cost system for innovative product development, *Proceedings of the Institution of Mechanical Engineers Part B- Journal of Engineering Manufacture*, Vol. 216, 2002, pp 999-1019.

[2] Hundal, M. S., Rules and models for low cost design, presented at ASME design for Manufacturing Conference, 1993.

[3] Abdalla, H. S., Concurrent engineering for global manufacturing, *International Journal of Production Economics*, Vol. 60-1, 1999, pp 251-260.

[4] Shehab, E. M. and Abdalla, H. S., Manufacturing cost modelling for concurrent product development, *Robotics and Computer-Integrated Manufacturing*, Vol. 17, 2001, pp 341-353.

[5] Gayretli, A. and Abdalla, H. S., A prototype constraint-based system for the automation and optimization of machining processes, *Proceedings of the Institution of Mechanical Engineers Part B- Journal of Engineering Manufacture*, Vol. 213, 1999, pp 655-676.

[6] Cross, N., *Engineering Design Methods, Strategies for Product Development*: John Wiley and Sons, 1994.

[7] Baxter, M., *Product Design: A practical guide to systematic methods of new product development*. London: Chapman & Hall, 1995.

[8] Tseng, K., El-Ganzoury, W. and Abdalla, H., Integration of Non-networked Software Agents for Collaborative Product Development, *Computer-Aided Design & Applications*, Vol. 2, Nos. 1-4, 2005, pp 377-386.

[9] Ulrich, K. T. and Eppinger, S. D., *Product design and development*, McGraw-Hill, Inc., 1995.

[10] Pahl, G. and Beitz, W., *Engineering Design*, Springer-Verlag, 1984.

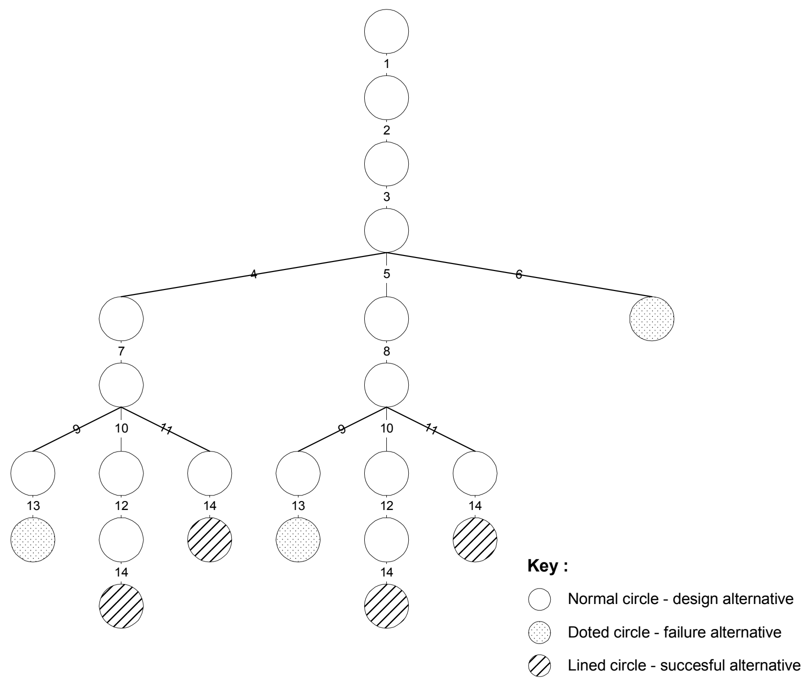


Fig. 11. Search tree.