Identification of Optimised Open Platform Architecture Products for Design for Mass Individualisation

Ravi K Sikhwal[1] and Peter R N Childs[1]
[1] Dyson School of Design Engineering, Imperial College London, South Kensington, London, SW7 2AZ, United Kingdom
(r.sikhwal15@imperial.ac.uk)

Abstract. Mass Individualisation is a new product design paradigm that comprises an open hardware platform and multiple independent modules for end-user's selection that are integrated with the platform. Open platform architecture products (OPAP) are the key enablers for this paradigm. Based on explorative literature analysis, with practical insights from an industrial questionnaire survey, an Innovation toolkit for the end-user has been developed. This provides a means for selecting an optimal OPAP. The design of the Innovation toolkit has been approached in four different steps: Modelling of OPAP Products, Modelling of evaluation measures and evaluation indices with end-user preferences, Identification of the optimal module options for every configuration, and Configuration optimisation. Two case studies have been presented to demonstrate the effectiveness and to illustrate that the Innovation toolkit can readily be applied to these types of product development to obtain highly individualised and optimised OPAP.

Keywords: Design optimisation·innovation toolkit·mass individualisation·open platform architecture products.

1. Introduction

Industrial product design has changed significantly over time, inspired either by market conditions or the consumers' desire for the product offering. With the industrial revolution, the idea of individually crafted designs was replaced by product design for mass production, followed subsequently by product design for mass customisation. Mass customisation aims at customisation of products and services for customers at a mass production price and efficiency [3]. Traditionally, most products are designed by professionals working for the underlying firms in design teams [4]. However, a significant shift has been observed over time, with technological advancement. Innovation technologies (IvT) [5] have facilitated new strategies for product design and development. New technologies have democratised the tools for both invention and production [6]. Anyone with an idea can use advanced and accessible technologies and turn it into a product. Some users are able and motivated enough to share their innovative ideas with firms. Ninan and Siddique [7] proposed configuration tools to optimise and assess the feasibility of customer choices.

The growing saturation of markets and continuously increased aspiration levels of customers are the primary drivers for the development of customer individualised products [8]. These products draw on a new set of strategic decisions related to how value is created and captured, how the relationship with conventional business partners such as suppliers are redefined [9]. These changes in user aspirations and inclination towards more individualised product offering are the basis for a relatively new product design paradigm, known as Product Design for Mass Individualisation (MI). Explorative literature analysis and practical insights from an industrial questionnaire survey, conducted among consumer product companies, shows that end products in MI are highly individualised and technologically advanced [10].

1.1. Product design for Mass Individualisation (MI)

Product design for MI is based on open platform architecture products (OPAP) that comprises of an open hardware platform and multiple independent modules. The open hardware platform is integrated with different modules as per end user's needs, using the interactive design program. This paradigm is named "Mass Individualisation" as products are mass produced, but each one is tailored to the needs of the individual buyer [11].

In the framework developed, it is envisaged that large manufacturers will provide the platform of the product along with interfaces for adding modules. These interfaces/modules can be satisfied by different module options. Smaller companies/3rd party module vendors will invent and produce modules options. Different module options will have different parameters to fulfil the requirement. Thus the basis of competition shifts from discrete products to modules and product systems consisting of interfaced modules on the product platform.

The variability that MI creates in traditional product design, end-user needs, regulations from different authorities and standards can be challenging. Given the benefits MI provides to all the actors, these challenges are worth addressing. As earlier work [10] suggests, MI could be beneficial in a range of markets, but consumer electronics and furniture markets are well-known sectors that can benefit readily from the end-users' perspective. MI with OPAP can be implemented in various products such as smartphones, smartwatches, individualised furniture.

Although MI has been considered a promising industrial product design paradigm to meet the increased aspiration level of today's customers, it also faces many challenges due to multi-dimensional variations of end products. To model these variations and capture innovation from different actors, a systematic approach and tools are required. Different constraints from so many actors have to be taken into account while solving these models. Xie, Henderson [12] developed modelling for engineering product configuration problems and solved them by constraints satisfaction. Once the modelling of these individualised products is done, the next step is to identify the optimal configuration with optimal module options. Hong, Hu [13] used genetic programming to identify the optimal product configuration and its parameters for one-of-a-kind production. In this paper, an Innovation toolkit is presented to identify the optimised OPAP for product design for MI.

2. Open platform architecture products (OPAP)

Open platform architecture products (OPAP) are the key enablers for Product design for MI. OPAP are based on an open hardware platform with many interfaces for module integration. Fig. 1 illustrates a typical schematic representation of an OPAP skeleton with interfaces, specific and unknown module options.

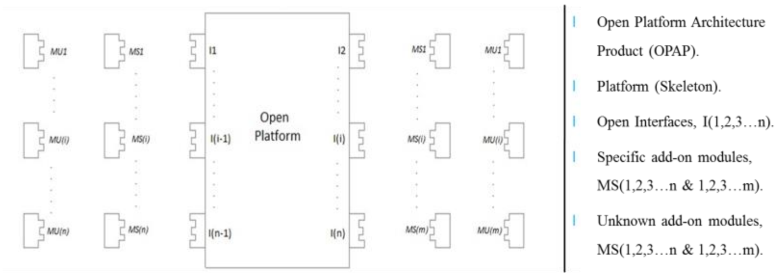


Fig. 1. Schematic representation of an OPAP with platform, interfaces and module options

Specific module options are the modules selected at the time of first use of the product, where unknown module options demonstrate adaptability or modules added in future as per users change in requirement. In this work, only specific module options are the primary focus for the development of the Innovation toolkit.

2.1. Innovation toolkit for OPAP

A networked Innovation toolkit describes a design environment which enables actors to formulate their requirements iteratively and transfer these into a producible solution by an iterative process with continuous live networked support from other actors in the OPAP ecosystem. The function of one module or module system can be optimised with other related modules or module systems with this Innovation toolkit. A multi-level optimisation model is developed for this Innovation toolkit to identify the best design configuration with optimal module options which satisfies all the requirements of the end-user. Fig. 2 depicts the framework for the Innovation toolkit including roles of different actors and optimisation model.

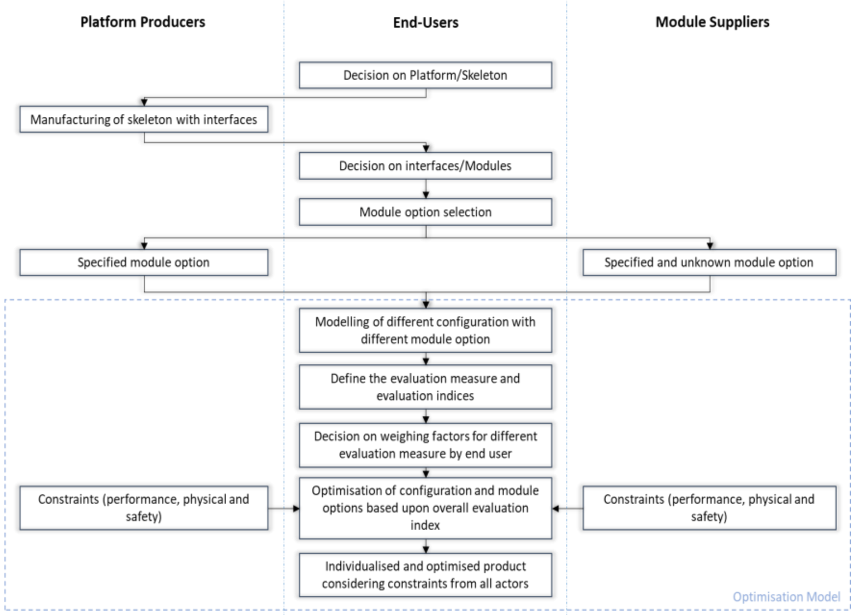


Fig. 2. Framework for the Innovation toolkit

The design of the optimisation model for the Innovation toolkit has been approached in four different steps: Modelling of OPAP, Modelling of evaluation measures and evaluation indices with end-user preferences, Identification of the optimal module options for every configuration, and Configuration optimisation.

The following assumptions are used for the development of the model:

- The end-user acts as a lead to decide on the platform.
- Adaptability and cost of the all feasible configurations with different module options are comparable.
- The Primary requirement of the end-user can be represented by the module options of each module/interface, and it is only allowed to configure a product that offers higher-order module options than the customer requirements.

The end product is a result of participation from many module option suppliers and the end-user. This multi-directional participation causes many variations in the end product. These variations include two kinds of variation: variation of configuration in terms of different interfaces used for modules and variation of module options for selected interfaces/modules. Different module options can be denoted by different parameters. After selecting particular modules for skeleton interfaces, the second choice will be to select module options in terms of desired parameters for modules. So a new method to model the variations of OPAP product configuration and the variations of product parameters in terms of module options is required.

3. Modelling of OPAP with evaluation measures and evaluation indices

Compared with traditional product customisation approaches, the variation of configuration and parameters is too high in product design for MI. Therefore, a sophisticated automated Innovation toolkit is required for modelling of OPAP with variations. Different product configurations are modelled by an AND-OR tree, as shown in Fig. 3. The OPAP structure can be decomposed into different sub-structures (module), connected with an AND relation. Every sub-structure can be satisfied with different module options, associated with an OR relation. Each module option in the AND-OR tree is further modelled in terms of parameters.

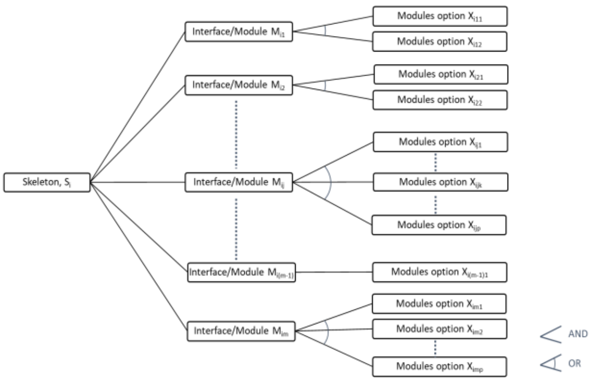


Fig. 3. AND-OR Tree diagram for modelling different OPAP Configuration

A feasible individualised OPAP can be obtained from the AND-OR tree through a tree-based search [14], described by a collection of nodes (Skeleton, Interfaces, modules options). In this work, the following conditions are used to generate different feasible design configurations:

1. The first node should be the root node, to be selected.
2. After selecting the root node, all the sub-nodes should be selected, if all its sub-nodes are connected with an AND relation.
3. After selecting the root node, only one of the sub-nodes should be selected, if all its sub-nodes are connected with an OR relation.

If a module node for the i^{th} design configuration S_i *(i=1, 2,…..n)* is defined by M_{ij} *(j=1, 2,….m)*. This design configuration can be described as follows:

$$S_i = (M_{i1}, M_{i2} \ldots\ldots M_{im}) \ , \ i = 1, 2, 3, \ldots\ldots. n \tag{1}$$

A module node is associated with the different module options nodes. These module option nodes represent different design parameter choices. The k^{th} design parameter X_{ijk} of the module node $M_{i,j}$ is defined in the form of $M_{ij}.X_{ijk}$. Therefore the parameters of a module node, *Mij*, can be described as follows:

$$X_{i,j} = \left(M_{ij}.X_{ij1}, M_{ij}.X_{ij2} \ldots\ldots M_{ij}.X_{ijk}\right) \ , \ i = 1, 2, 3, \ldots\ldots. n, \text{ and } j = 1, 2, 3, \ldots\ldots m \tag{2}$$

The parameters for the i^{th} design configuration considering all involved nodes are defined by

$$X_i = (X_{i1}, X_{i2} \ldots\ldots X_{ik}) \ , \ i = 1, 2, 3, \ldots\ldots. n \tag{3}$$

The complete design solution of this configuration can be then defined,

$$D_i = (S_i, X_i) \ , \ i = 1, 2, 3, \ldots\ldots. n \tag{4}$$

If only i^{th} design configuration is considered in terms of parameters, then

$$S_i = \left(X_{i1}, X_{i2} \ldots\ldots X_{in_i}\right) \tag{5}$$

Different product configurations are evaluated by customer satisfaction measures and indices.

3.1 Evaluation measures & Evaluation indices

An evaluation measure can be either a constant, a monotonic or a non-monotonic function of life cycle time. For this research work, these measures can be classified into two categories: performance measures, P_i, and cost measures, C_i. Performance measures include efficiency, speed, resolution, etc., whereas cost measures include product cost, module replacement cost, maintenance cost, etc.

For a product configuration, *S*, with *n* parameters, evaluation measure in the i^{th} evaluation aspect (measure) is defined by,

$$E_i = E_i(X_1, X_2, X_3, \ldots\ldots. X_n) \tag{6}$$

Different evaluation measures are in different units, so these evaluation measures need to be converted into comparable evaluation indices between 0 and 1, which represents different levels of satisfaction [15]. Customer (End-user) satisfaction has been selected as an evaluation index in this work. The evaluation measure and evaluation index can be related by a linear or a nonlinear relation.

The customer satisfaction index, in the i^{th} evaluation aspect, is defined by,

$$CS_i(X) = F_i[E_i(X)] \tag{7}$$

The overall customer satisfaction index can be modelled as follows:

$$CS(X) = \frac{1}{W_1+W_2+W_3+\ldots\ldots+W_m}\left[W_1 CS_1(X) + W_2 CS_2(X) + W_3 CS_3(X) + \ldots\ldots. + W_m CS_m(X)\right] \tag{8}$$

where W_1, W_2,……..,W_m are *m* weighting factors for *m* evaluation indices, selected by end-users, according to their individual requirements and preferences.

4. Identification of the optimal OPAP configuration with optimal module options

Since a large number of design configuration with different module options can be selected to fulfil the individualised requirement, a multi-level optimisation is employed to identify the best design configuration with optimal module options. This will maximise the satisfaction of the end-user requirements within the constraints provided by other actors of the OPAP ecosystem. Platform producers will define some constraints including functional, safety and assembly constraints. The module options providers will also provide some constraints based on their manufacturing capability, spatial and other constraints.

The overall customer satisfaction index can be considered as the optimisation objective function. The average-case in which the average evaluation index is used as the objective function method is generally the most suitable for the optimal design of OPAP. The optimisation is conducted at two levels: the module options level and the configuration level.

4.1 Module option optimisation

The first level of optimisation is conducted at the module options level, i.e. selection of optimised module options into chosen interfaces, for a given configuration. In this work, module option optimisation is done with penalty-based optimisation [16] method. In the presence of constraints provided by different actors, penalty functions are used to convert a constrained optimisation problem into an unconstrained optimisation problem. The optimal parameter values for a product configuration, *S*i, defined by its parameters $(X_{i1}, X_{i2} \ldots\ldots X_{in_i})$, using constrained optimisation approach, can be obtained as follows:

$$\underset{\text{wrt } X_{i1},X_{i2}\ldots\ldots X_{in_i}}{\text{Max}} \; CS(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) \tag{9}$$

Subject to:

$$X_{ij}^{L} \leq X_{ij} \leq X_{ij}^{U}, \qquad j = 1, 2, 3, \ldots \ldots . n_i \tag{10}$$

$$h_{ij}(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) = 0, \qquad j = 1, 2, 3, \ldots \ldots \ldots \ldots k_i \tag{11}$$

$$g_{ij}(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) = 0, \qquad j = k_i + 1, k_i + 2, \ldots \ldots \ldots \ldots m_i \tag{12}$$

Such a constrained optimisation problem can be converted into a non-constrained optimisation problem by adding a penalty term to the objective function mentioned in the equation (9). The modified objective function with a penalty term can be defined as follows:

$$UCS_i(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) = CS_i(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) - \alpha . p_i(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) \tag{13}$$

where, UCS_i represents the non-constrained form of CS_i, $p_i(X_{i1}, X_{i2} \ldots\ldots X_{in_i})$ is the penalty term for the unconstrained objective function and φ is a multiplier constant that determines the magnitude of the penalty. The penalty term is defined as follows:

$$p_i(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) = \sum_{j=1}^{k_i}\left[h_{ij}(X_{i1}, X_{i2} \ldots\ldots X_{in_i})\right]^2 + \sum_{j=k_i+1}^{m_i}\left[g_{ij}(X_{i1}, X_{i2} \ldots\ldots X_{in_i}) + \left|g_{ij}(X_{i1}, X_{i2} \ldots\ldots X_{in_i})\right|\right]^2 \tag{14}$$

4.2 Configuration optimisation

The second level of optimisation is conducted at the configuration level, i.e. selection of optimised OPAP configuration for the end product. The following optimisation model is used:

$$\underset{wrt\ S_i^*}{\text{Max}} \; CS(S_i^{\,*}) \tag{15}$$

Subject to: $1 \leq i \leq p$ (16)

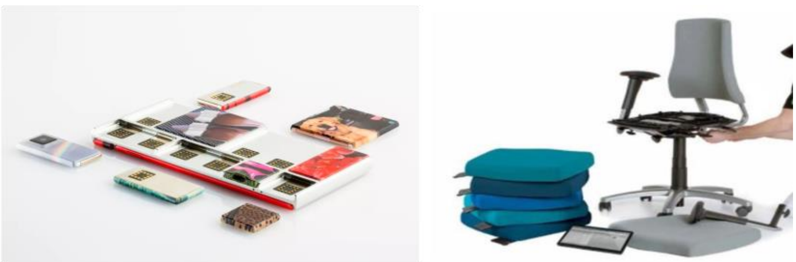
Where S_i* is iterated among the feasible configurations with the optimised module options, *i* represents the i^{th} design configuration candidate, and *p* is the number of all feasible design configuration candidates.

In this work, genetic programming [17] is used for configuration optimisation. Genetic programming is based on genetic algorithms and is an evolutionary method to solve an optimization problem when solutions to the optimization problem can be modelled by tree data structures. In genetic programming, multiple solutions are considered in the population of a generation. These solutions are called chromosomes which evolves with better evaluation measures.

The genetic programming used in this work is inspired by the work done by Hong, Hu [13]. It is formulated in the following steps:

1. Create the initial generation with *n* individuals. Each individual represents a feasible configuration, created randomly from the configuration AND-OR tree.
2. Obtain the overall customer satisfaction index of each individual in the current generation with optimal module options from first level optimisation. This is used as the fitness of the corresponding individual.
3. Create a new generation from the current generation by repeating the following steps until the number of individuals in the new generation reaches *n*.
   Reproduction. Select two parent individuals in the current generation according to their fitness measures using the roulette wheel selection method.
   Crossover. Calculate the crossover probability. If a crossover operation is required, cross over the two selected parent individuals to form two new offspring individuals. Otherwise, no crossover is required.
   Mutation. Calculate the mutation probability for each of the two offspring individuals. If a mutation is required, mutate the new offspring individually.
4. Select the newly created generation as the current generation.
5. If the average fitness of a generation cannot be significantly improved in the last *m* generations (i.e. the improvement is less than a pre-defined small number $\mathcal{E}$"), or the pre-defined maximum generation, g_{max}, has been reached, the evolution process should be stopped, and, the best individual in the current generation is selected as the optimal product configuration.
6. Go to Step (2).

5. Case studies



(a) (b)

Fig. 4. (a) Google ARA, A smartphone based on OPAP [1] **(b)** An individualised chair [2]

The concept of product design for MI can be implemented in the market with a variety of products, but our earlier study suggests that consumer electronics and furniture industries would be a good point to start. Following this suggestion, a consumer electronics product, OPAP Smartphone (based on Google ARA) and an individualised chair (based on Axia Smart Chair from Nomique) are used as case studies for our work, as shown in Fig. 4(a) and Fig. 4(b), respectively.

5.1. An OPAP smartphone (Google ARA)

Information available in the public domain for ARA has been used to formulate the optimisation problem for an OPAP smartphone. Information is gathered from MDK (Modular development kit), a guide for the development of modular technology that Google has provided to developers [1]. Due to variations of OPAP, selected products are not the optimised one with optimal modules. Once the end-user puts forward the choice for the required module type (e.g. battery module), different smaller companies will provide different module options (e.g. different capacities). Hence, the Innovation toolkit will be employed to find the end product which provides a smartphone with optimal OPAP for the given requirements. Two feasible product configurations can be created with an AND-OR tree, as shown in Fig. 5.

$$S_1 = (\text{Camera Module}, \text{Battery Module}, \text{Screen Module}) \tag{17}$$

$$S_2 = (\text{Speaker Module}, \text{Battery Module}, \text{Screen Module}) \tag{18}$$

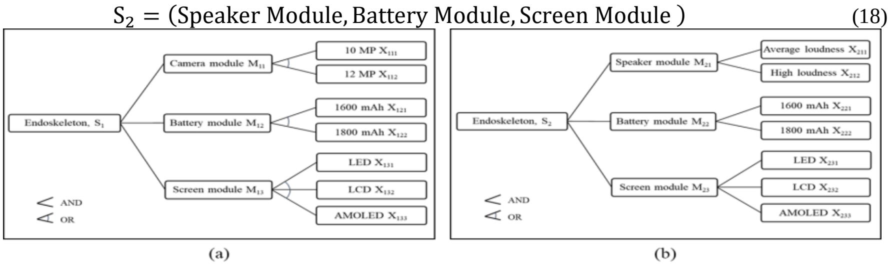


Fig. 5. Two different feasible configurations based upon interfaces selected by end-users

From the configuration S_1, different sub-configurations, X_1 and X_2 can be obtained with different module options, shown in Fig. 6.

$$X_1 = (10MP, 1600\text{mA h}, \text{AMOLED}) \tag{19}$$

$$X_2 = (12MP, 1800\text{mA h}, \text{LED}) \tag{20}$$

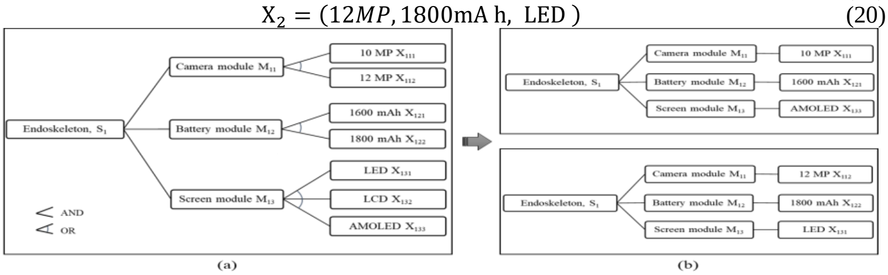


Fig. 6. (a) OPAP smartphone configuration, S_1 **(b)** Feasible product sub-configurations.

To obtain an optimal configuration for S_1, first level optimisation is employed. Various evaluation measures for this case study are shown in Table 1. The product cost for different configurations can be determined based on individual cost from different module options suppliers. These three evaluation measures C_p, P_w and P_{bb} are converted into three customer satisfaction indices, I_p, I_w and I_{bb}, respectively.

Table 1 Customer evaluation measures selected for OPAP smartphone

Evaluation measures		Unit	Representation
Cost evaluation measure	Product Cost	GBP (£)	C_p
Performance evaluation measure	Product Weight	Grams (g)	P_w
	Battery backup	Hours	P_{bb}

If the weighting factors provided by end-users are x_1, x_2, and x_3 then the overall customer satisfaction index,

$$CS(X) = \frac{1}{x_1+x_2+x_3}[x_1 I_p + x_2 I_w + x_3 I_{bb}] \tag{21}$$

This equation will be used for the optimisation of customer satisfaction index with optimal module options' parameters as per equation (9). Once both configurations are with optimal module options, a genetic algorithm is employed to obtain a highly individualised and an optimal OPAP smartphone. For a larger number of variations, automated Innovation toolkit can be used.

5.2. An individualised chair (Axia smart chair by Nomique)

Another product, the Axia smart chair from Nomique, see Fig. 4(b) was selected as a case study to demonstrate the application of the OPAP and the Innovation toolkit. Different evaluation measures, e.g. chair cost, chair weight were selected for this case study and converted into respective evaluation indices to get the overall customer satisfaction index for the first level of optimisation (for optimal module options). The second level optimisation is then employed to obtain optimised and highly individualised smart chair. This case study is presented briefly in this paper just to demonstrate the effectiveness of introduced Innovation toolkit in range of products and the arising configuration are illustrated in Fig. 7.

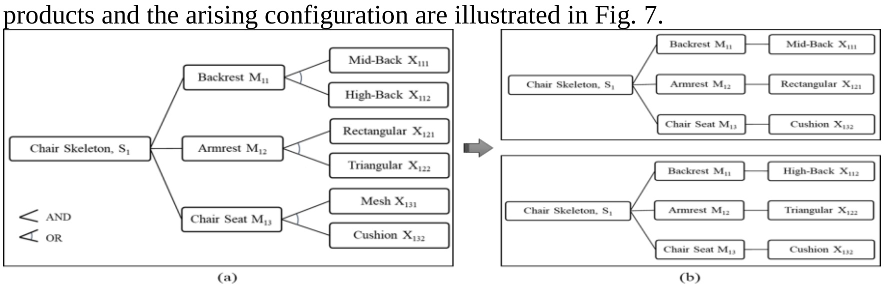


Fig. 7 (a) OPAP smart chair configuration **(b)** Feasible product sub-configurations.

6. Conclusion

An Innovation toolkit for identifying the optimal OPAP has been introduced. Variations in product configurations with different module options in an OPAP are modelled by nodes in an AND-OR tree. The AND-OR with different nodes for module options provides a systematic framework to model large variations of OPAP configurations. The optimal module option for every interface with maximum overall customer satisfaction index is identified by constrained optimisation, followed by configuration optimisation to identify optimal OPAP configuration out of all the feasible configurations. Two case studies are used to demonstrate the applicability of this Innovation toolkit. These case studies show that the Innovation toolkit developed in this work can readily be applied to this type of product development to obtain a highly individualised and optimised OPAP.

Product design for MI is a relatively new area where much research has to be done. To realise and implement this new approach in the market, many issues need to be addressed including optimisation of module option during the product operation stage and development the Innovation toolkit further considering the same. Different monetary aspects, IP rights, acceptance of this approach by existing designers are also need to be tested before implementation in the market.

Acknowledgements. The case studies used in this paper are based on the information available in the public domain about Google ARA and Axia smart chair by Nomique.

References

1. Project ARA by Google. 2016 [cited 2016 20-05-2016]; Available from: https://twitter.com/projectara?lang=en
2. Axia Smart Chair by Nomique. 2018 [cited 2018 10-02-2018]; Available from: http://nomique.com/products/task-ergonomics/axia-smart-chair-2/.
3. Tseng, M.M. and J. Jiao, *Mass Customization*, in *Handbook of Industrial Engineering: Technology and Operations Management*, G. Salvendy, Editor. 2001, John Wiley & Sons, Inc.: Canada. p. 684-709.
4. Ulrich, K.T., *Design: Creation of Artifacts in Society*. 2011: Trustees of the University of Pennsylvania.
5. Dodgson, M., D. Gann, and A. Salter, *Think, Play, Do: Innovation, Technology, and Organization*. 2005: Oxford University Press.
6. Anderson, C., *Makers: The New Industrial Revolution*. 2012: Crown Business. 272.
7. Ninan, J.A. and Z. Siddique, *Internet-based framework to support integration of customer in the design of customizable products.* Concurrent Engineering-Research and Applications, 2006. **14**(3): p. 245-256.
8. Holle, M. and U. Lindemann, *Design for Open Innovation (DfOI) — Product structure planning for open innovation toolkits*, in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*. 2015: Bandar Sunway, Malaysia.
9. Porter, M.E. and J.E. Heppelmann, *How Smart, Connected Products Are Transforming Competition.* Harvard Business Review, 2014. **92**(11).
10. Sikhwal, R.K. and P.R.N. Childs, *Design for Mass Individualisation: Introducing Networked Innovation Approach*, in *World Mass Customization & Personalization Conference (MCPC 2017)*. 2017: RWTH Aachen, Germany.
11. Koren, Y., et al., *Product Design for Mass-Individualization.* Design Conference Innovative Product Creation, 2015. **36**: p. 64-71.
12. Xie, H., P. Henderson, and M. Kernahan, *Modelling and solving engineering product configuration problems by constraint satisfaction.* International Journal of Production Research, 2005. **43**(20): p. 4455–4469.
13. Hong, G., et al., *Identification of the optimal product configuration and parameters based on individual customer requirements on performance and costs in one-of-a-kind production.* International Journal of Production Research, 2008. **46**(12): p. 3297–3326.
14. Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2002, Upper Saddle River, NJ: Prentice Hall.
15. Yang, H., D. Xue, and Y.L. Tu. *Modeling of Non-Linear Relations Among Different Design Evaluation Measures for Multi-Objective Design Optimization*. in *ASME 2005 IDETC/CIE*. 2005. CA, USA: American Society of Mechanical Engineers.
16. Arora, J.S., *Introduction to optimum design*. 4th ed. 2016: Elsevier Academic Press.
17. Koza, J.R., *Genetic programming: on the programming of computers by means of natural selection*. 1992: MIT Press. 680.