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

MODULE DESIGN BASED ON INTERFACE INTEGRATION TO MAXIMIZE PRODUCT VARIETY AND MINIMIZE FAMILY COST

Tsuyoshi Koga and Kazuhiro Aoyama
Department of Environmental and Ocean Engineering, The University of Tokyo

ABSTRACT
This paper proposes a design method for platform modules by considering the variety of the product family and the design and production cost. Computational and graphical models of the product family architecture are proposed as the summation of all line-up products. Based on the product family model, the design problem of interface integration is formalized as the generation of a variety of products. The platform module design is represented as the generalized module architecture design that is commonly used in all products in the product family. A decision making system is developed considering the value of the product family, variety of line-up products, cost reduction effect of the platform modules, and cost of design change. The proposed design method addresses: (1) how to computationally formalize the product family and platform modules; (2) how to evaluate the difficulty of design change, advantages and cost of the platform module, and the variety of the line-up; and (3) how to design the platform module considering the product family strategy.

Keywords: Platformization, common architecture, product family model, module, cut-set matrix

1 INTRODUCTION

1.1 Background
Various products are required to satisfy various customer needs in niche market and mainstream markets. It is difficult to achieve a good balance between product variety and cost, since manufacturing a variety of products necessitates design and production costs. Modularization and platformization considering a product family contributes to improvements in both variety and cost. The use of common modules and platforms in a product family reduces design and production cost. An interface integration generates the product variety, because generalized interface creates new combination of modules.
For example, the vehicle platform strategy is one of the important strategies for motor companies, since it determines the characteristic and a profit ratio of the company. The production and maintenance of press dies for the body and chassis is expensive. The use of a common platform for producing both the body and chassis reduces the number of press dies required. Presently, the cost competition is so intensive that manufacturers that have several dozens of products actually use only a few platforms in their lineups. Therefore, the appropriate design of platforms is highly desired.
This study defines a module and a platform module as follows:

- Module: *A group of components integrated from a certain viewpoint.*
- Platform module: *A group of modules commonly used in a product family.*

Platform design is one of the important decisions, because it determines the common architecture of products in the product family. Not only the variety and cost of products but also the evolution of technology and the organizational architecture are significantly dependent on the common architecture. Organizations aim to maximize their profits and the value of the product family, as well as to achieve an efficient division of labour by designing an appropriate common structure by the platform design.
Determining the appropriate platform is very difficult because the platform module is significantly dependent on the topological structure of the products in the product family. Computational design support contributes to the logical platform design. Hence, this study proposes a design method that enables the designer to design the platform modules by considering the variety and cost of products

based on the computational representation model of the products in the product family, modules and product structure.

1.2 Recent Works on Product Family Design

Presently, several methodologies have been developed for designing product families and platform modules. Neison proposed the multicriteria optimization method for the product platform, this method aimed to realize benefits through reduction of inventory, proliferation of different parts, and design lead-in time for a product [1]. For the purpose of the topological architecture design, it is necessary to introduce a product and platform model that can represent the product structure.

Simpson developed an interactive web-based platform customization framework as an extension for the product family design and presented a prototype system [2]. Our study attempts to propose the meta-model of platforms and product family based on their framework.

Raghothama proposed a topological framework for a parts family [3]. They addressed the issues of how to generate members of a parts family and determine which parts family a given object belongs to. Their method is effectively used in the detailed design stage, and its application to the early and strategic design stages is desired.

The design method for the combinations and attributes of modules by using the optimization method is proposed by Fujita, Akundi et al. [4] [5]. This study mainly focuses on the design of the topological structure of the platform through the products based on their attribute optimizations.

Shiddique presented a reasoning method for the product family architecture by considering the product family architecture and manufacturing process [6]. Based on their definition of a product module model, representation of product options, and production process model, we propose a product family model to discuss the interface integration and parts generalization.

Jonathan proposed the evolution model of products in the product family [7]. This study represents Jonathon’s product family evolution model as a result of designer-based computation and virtual trial and error steps based on the proposition of the product family and platform model.

1.3 Purpose of this research

The reduction in cost and increase in product variety are both necessary to increase the value of the product family. To explain the increase in the value, this study proposes a model of product variety increase based on the interface integration. This study formalizes the design of the platform modules by taking into consideration the product family based on the common architecture model and interface integration. Furthermore, we propose a formalization-based platform design method for increasing and maximizing the value of the entire product family.

In order to maintain a good balance between variety and cost, it is necessary to estimate and consider (1) devising options by unifying the parts and interfaces, and (2) reducing the design and production costs by using the common platform modules. The design of the platform module is one of the chance discovery issues by defining many constraints and searching vast space of solutions. Hence, a system design method that models all the design objects and constraints [A] as one system and estimates various factors [B] is required for the development of an appropriate platform module design.

[A] Design objects and platformization constraints

Product structure, common architecture, unified parts, unified interfaces, options, unified modules, and platforms

[B] Factors contributing to the value of the product family

Variety of products, design change cost, and reduction in production cost

The existing methodologies proposed in earlier studies are incomplete. There exists no platform design method that can consider all the objects and factors given in [A] and [B]. Generally, the common architecture that significantly depends on the product structure has a longer lifespan than a single product. The product family evolution must be considered at the platform design stage. Hence, the purpose of this research is stated as follows:

Research purpose:

To propose a platformization method that can support the design of appropriate platform modules based on the trial and error of the objects and constraints [A] by using the evaluation results of the product family factors [B], taking into consideration the product family evolution

Based on the Siddique’s product option model and production model [6], this study proposes the option generation model by adding the unification model for parts and interfaces. An assembling process calculation method using a cut-set of the product representation graph proposed by Mantripragada is expanded as the method for calculating the unification and the platformization processes [8]. This study expands the proposed systematic modularization method by employing multi-stage decomposition proposed by the authors [9] from the early design stage to the detailed design stage.

2 PLATFORM DESIGN METHOD

2.1 Product Family Model

The product family consists of line-up products. The product family model that represents both of the product structures and the relationships between the products is required for designing the platform module. The product family model used in this study is shown in Figure 1.

A single product is modelled from the entity model, attributes, and their relationships. This product comprises components, parts and sub-systems. These are represented by the entity *E*. This entity is represented by the attributes *A*, e.g., its shape, material, and cost. A single product comprises a set of entities. The structural connections between these entities are represented as the link *F* between them. The single product model is represented as a graph whose nodes and links represent entities and structural connections respectively.

There exist strong relationships between the products in the product family. The existence of these relationships implies that the commonly used parts and interfaces can potentially be unified. In order to describe the possibility of unifying the parts and interfaces, a unification link U is introduced in the product family model. The unification link connects the entities of different products and represents the possibility of the integration of the entities at both ends. The integration cost of the entities is described at the unification link. A functional module that is commonly used across the different products in the product family is defined as a platform module *P*. *P* is represented as a subgraph of the product family model that comprises the entities and connection links.

The entire product family is represented as the summation of the graphs of the products on a common architecture element *C*. The common architecture comprises the common architecture element *c*, and represents the common structure of the products in the product family. The entity represents the product element that is not included within the common architecture element.

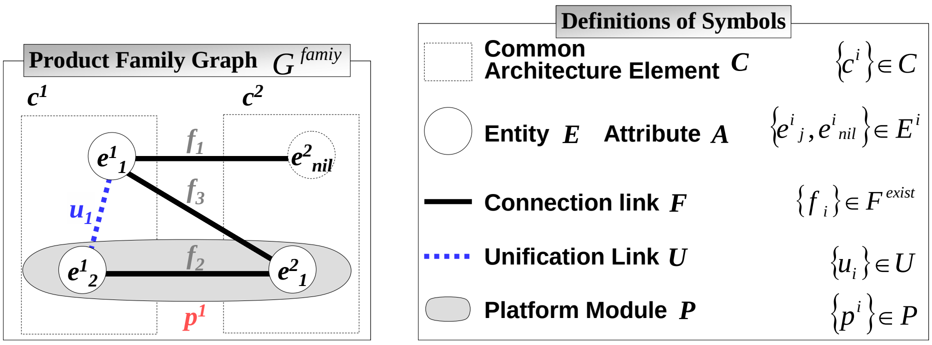

Figure 1. Product Family Model

2.2 Formulation of Product Family Model

Based on the definition of the product family model in Figure 1, a formulation of the product family is proposed as a series of formulas (1)-(6). The product family graph G^{family} comprises the product graphs $G^{product}{}_i$, and is represented in terms of *C*, *E*, *A*, *P*, *F*, and *U* as follows:

$$G^{famiy}(G^{product_1}, G^{product_2}, \cdots, G^{product_n}) = \{C, E, A, P; F, U\} \qquad (1)$$

C comprises the entities that belong to different products. Each entity must belong to one common architecture element:

$$c^i = \{e^i{}_1, e^i{}_2, \cdots, e^i{}_{n_i}\}, \quad E = \sum_{i=1}^{n} E^i \qquad (2)$$

All connection links are defined between entities that belong to different common architecture element:

$$F \subseteq E^i \times E^j, i \neq j \qquad (3)$$

The unification link must be defined between entities that belong to the same common architecture element:

$$U \subseteq E^i \times E^i \tag{4}$$

The entity contains the connection links, unification links and attributes:

$$e^i{}_j = \{F^i{}_j, U^i{}_j, A^i{}_j\} \qquad F^i{}_j \subseteq F, U^i{}_j \subseteq U, A^i{}_j \subseteq A \tag{5}$$

A platform is defined as the set of entities that belongs to the different common architecture elements:

$$p_i = \{e^s{}_l, e^t{}_m, \cdots, e^u{}_n\}, \qquad s \neq t \neq u \tag{6}$$

The product family is formulated from the graph representation in Figure 1 and the information structure and relationships shown in the formulas (1) - (6).

2.3 Formulation of Interface Integration and Parts Unification

This section formulates the operations of the interface integration and parts unification. This study defines the interface as a component between parts or modules, and describes both the part and interface as an entity. Hence, both interface integration and parts unification are represented by the integration of entities. The unified entities must belong to different line-up products and same common architecture element. The unification operation requires a unification link to be present between the entities to be integrated. The new entity that is generated by the integration inherits the connection links, the unification links, and attributes that the integrated entities possess by the 'and' inheritance rule. The operation of the parts and interfaces unification is formulated as given in (7):

For every two entities $e^s{}_i$ and $e^t{}_j$ that belong to the same common architecture ($s = t$) and that have a unification link between them ($U^s{}_i \cap U^t{}_j \neq \phi$), there exists a unification operation $h^{s,t}{}_{i,j}$ that generates the unified entity $e^s{}_{i,j}$:

$$h^{s,t}{}_{i,j} : (e^s{}_i, e^t{}_j) \mapsto e^s{}_{i,j}$$

$$\textit{where: } F^s{}_i \in e^s{}_i,\ F^t{}_j \in e^t{}_j,\ F^s{}_i \cup F^t{}_j = F^s{}_{i,j} \in e^s{}_{i,j} \tag{7}$$

The operation of the platform module design (referred to as platformization) is represented by the combinations of the operation of the interface integration and parts unification.

2.5 Overview of Platform Module Design Method

An overview of the platform design for the variety and cost of the product family is shown in Figure 2. Detailed procedures of this method are described in section 3 through a case study, and a support prototype is introduced in section 4. The main flowchart consists of the following steps: input the current products <A>, the common architecture design and the generation of a product family model <B>, interface integration and parts unification <C>, platform module design <D>, evaluation of the line-up <E>, and an evaluation of the product family <F>. The design and evaluation steps <A> to <F> are performed iteratively; this implies that the evolution spiral of the product family is designed based on the computational models of the product family and the platform modules. The detailed information regarding each procedure is provided below:

Design step <A> Input current products

The designer inputs the current products based on the existing line-up or the result of the iterative design loop. The entities, attributes, and connections of each product are fed as inputs to the computer. Based on the current products, the variety, costs, and profit ratio of the current product family are estimated. Figure 2 <A> shows examples: sedan, van and compact.

Design step <B> Product family model calculation

The designer introduces the common architecture based on the summations of the current product graphs. The product family model is generated based on this common architecture by the detection and definition of the unification links or the connection links.

Design step <C> Parts and interfaces unification

The design system automatically calculates the entire space of the unification plans of the parts and interfaces as a unification tree. The designer selects the unification plan and the unification scenario. Furthermore, the integrated parts and interfaces are designed.

Design step <D> Platform module design

The design system automatically calculates the entire space of the platformization plans as a platformization tree. The designer selects the platformization plan. The platform modules are designed by selecting the platformization plan.

Design step <E> Line-up Evaluation

The design system automatically calculates the line-up products, and the sharing ratio of the platform modules. Figure 2 <E> shows that the line-ups of Family Plan A and B comprises eight and three products, respectively. The platform module A-1 that designed in step <D> is commonly used in four products in the line-up, therefore the sharing ratio of module A-1 is 50%. Similarly, the sharing ratio of A-2, B-1, and B-2 is 50%, 66%, and 33%, respectively.

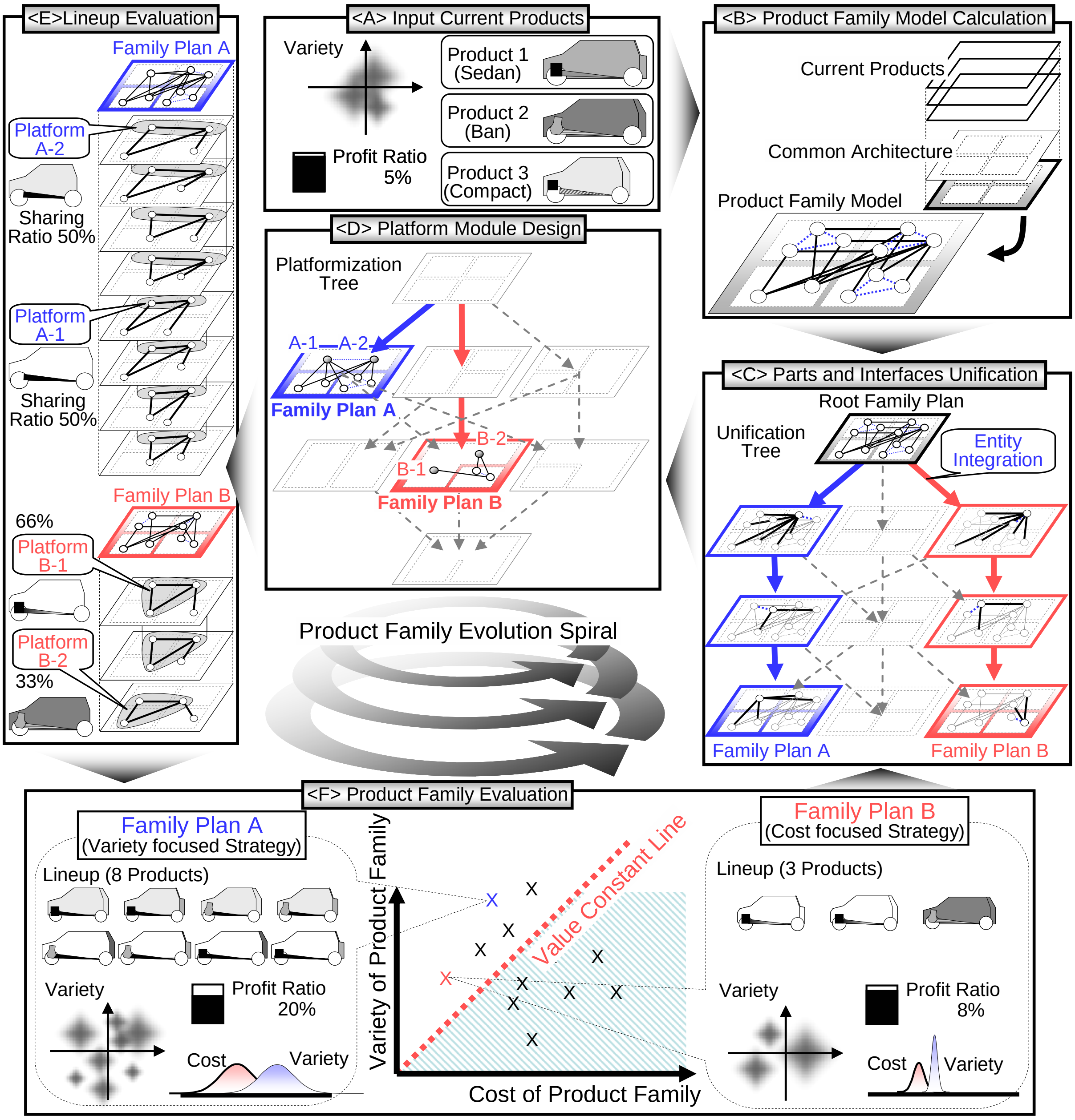

Figure 2. Overview of the Platformization Method

Design step <F> Product family evaluation

In this step, the product family plans are evaluated and compared based on the design system, and selected by the designer. The designed unified parts and interfaces, and platform modules in the product family plan determine the feasible products in the line-up and the design change, and production costs. Based on the evaluation result of the feasible products, the variety of product family is calculated. The integration costs and the number of modules determine the design and production costs. The graph in Figure 2 <F> shows the value of the product family plans. The points in the graph represent product family plans individually. The vertical axis represents the evaluation results of the product family variety. The horizontal axis represents the total cost of the product family, which is calculated by adding the summation of the design change cost and the reduction in the production cost. The plans of upper area of the value constant line have a higher value than the current product family plan. The proposed platform design method enables the designer to compare and select the family plan more appropriately based on the strategy of the product family.

3 INFORMATION PROCESSING PROCEDURES

This section introduces information processing procedures for the platform module design based on the example of the car platform design.

3.1 Generation of Product Family Model

The product family model comprises of the line-up products. Figure 3 shows the information processing procedures for the generation of the product family model, these procedures accept three current products as input and output the product family graph. The detailed procedures are as follows:

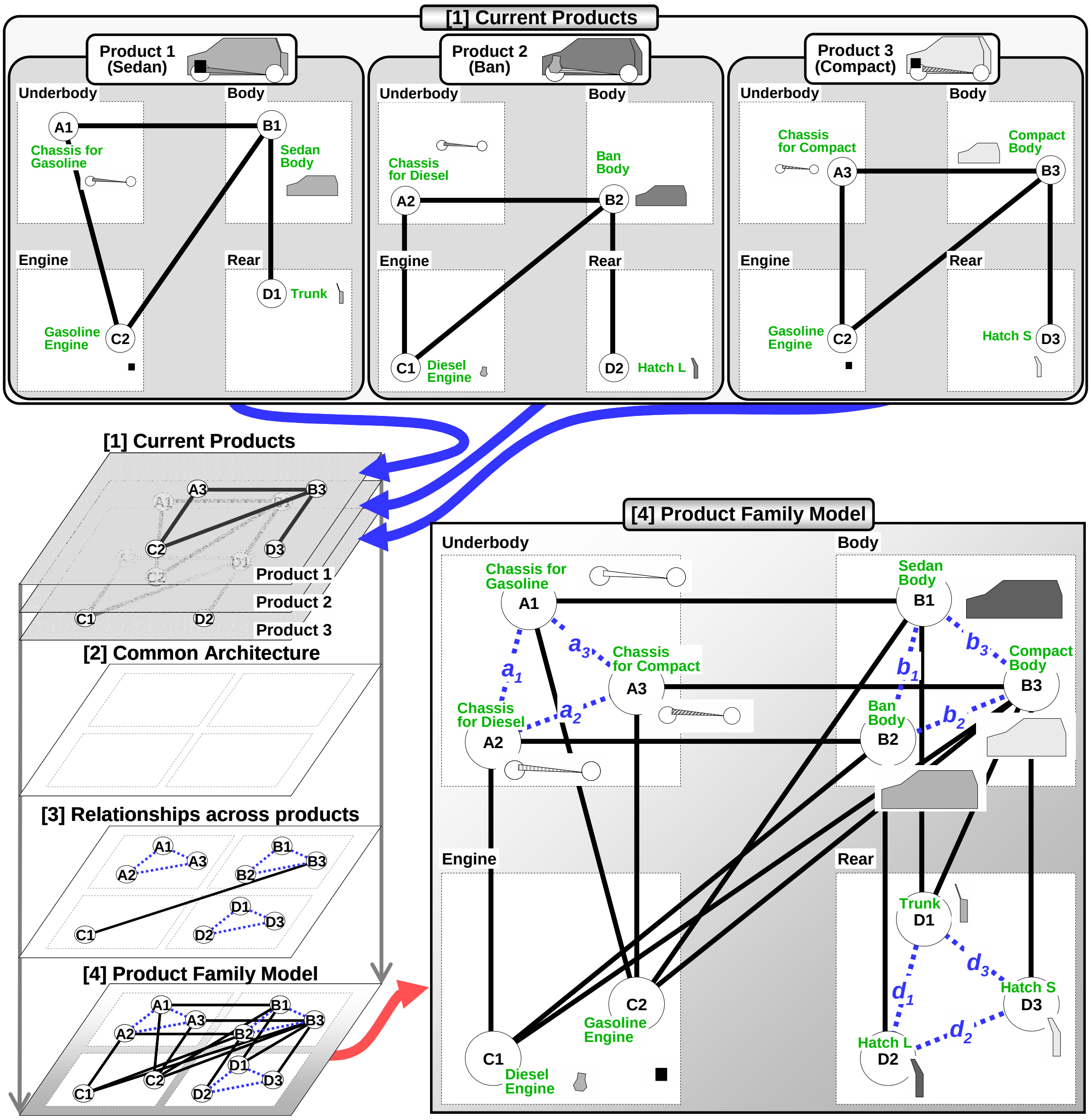

Figure 3. Generation of Product Family Model

STEP [1] Input current product

Based on three types of current products (Product 1, Product 2, and Product 3), the designer inputs three product graphs in the design system (Product1: A1-B1–C2–D1, Product2: A2-B2-C1-D2, and Product3: A3-B3-C2-D3).

STEP [2] Set common architecture

The designer introduces the common architecture that compromises Body, Underbody, Engine and Rear. The design system calculates one graph by integrating the product graphs.

STEP [3] Add relationships across products

The designer adds the relationships across products. The designer evaluates the possibility of the unification of entities in same common architecture element and defines the unification links, e.g. the unification links between A1-A3, A3-A2, and A2-A1. The additional connection links are defined between entities that can connect without design changes, e.g., the connection link between B3 and C1.

STEP [4] Generate product family

The product family model is generated as the integrations of three product graphs, nine unification links and one connection link.

3.2 Unification of Parts and Interfaces

The unification operation is performed by applying formula (7) to the product family model shown in Figure 3. Figure 4 shows an example of the unification operation. Two bodies B1 and B2 are integrated by the unification operation. The new integrated body inherits the connection and unification links. The information processing procedure is described in detail:

STEP [1]: Integration of the entities

The new integrated body B12 is generated by the unification of entities B1 and B2.

STEP [2]: Inheritance of the connection links

B12 inherits three connection links from B1 (A1-B1, C2-B1, and D1-B1) and three connection links from B2 (A1-B1, C2-B1, and D1-B1). Hence, six connection links are defined in B12.

STEP [3]: Inheritance of the unification links

B12 inherits two unification links from B1 (b_1 and b_3) and two unification links from B2 (b_1 and b_3). Hence, one unification link b_{23} is defined in B12.

The inherited connection and unification links indicate the specifications of the new integrated entity.

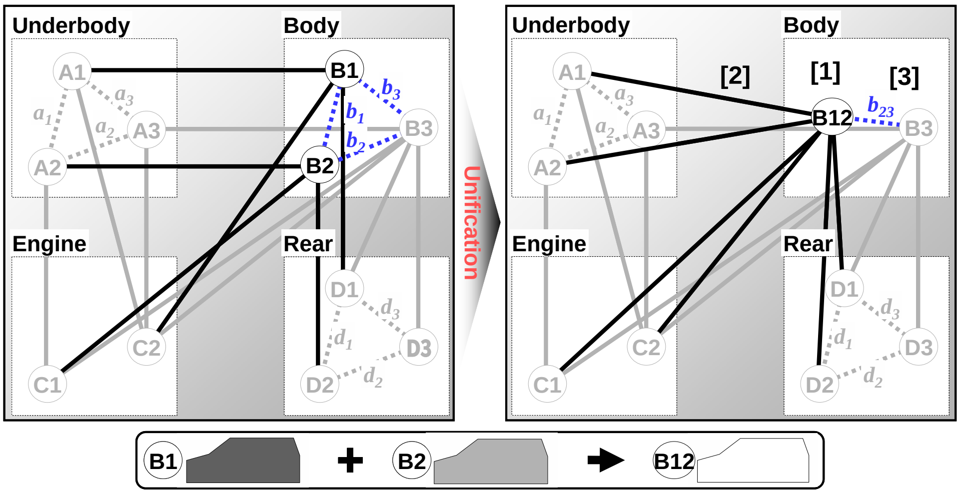

Figure 4. Body Unification

3.3 Unification Tree

The determination of the unification plan is a very difficult design problem since there exists a huge number of candidates. The proposed design method automatically calculates the space of the unification plans by the unification links. Figure 5 shows the unification tree that includes all the unification plans and unification scenarios. The unification tree is defined as a graph whose node is one unification plan (a unification plan node) and its sequence of unification operations.

The unification tree can be generated from the cut-set of the unification links in the product family graph. The unification plan node is defined as the matrix of the unification links. The product family model shown in Figure 3 has nine unification links (d_1, d_2, d_3, b_1, b_2, b_3, a_1, a_2, and a_3), therefore one unification plan node is represented by the nine cells in the matrix. A cell in the matrix represents a unification link. A black cell represents that the unification link is unified. A link between unification plans represents a unification operation. The designer can design a unification plan by selecting a unification plan node and a unification scenario by selecting a path from the root node to the unification plan node. Figure 5 shows the design of the unification plan by selecting two plans -

Family Plan A and Family Plan B - and the unification scenario by selecting the paths: root-F1a-F1b-F1c and root-F2a-F2b-F2c.

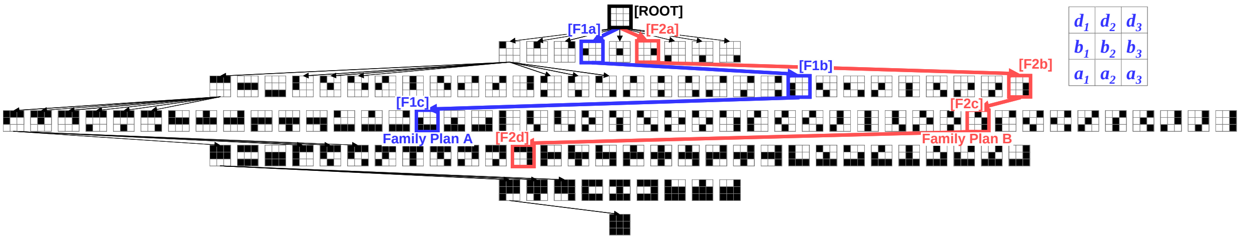

Figure 5. Unification Tree

3.4 Unification Process of Family Plan A

Based on the selected unification scenario, the parts and interfaces are integrated. The unification process changes the product family variety. Figure 6 shows the unification scenario of Family Plan A shown in Figure 5. The information processing procedure is described in detail as follows:

STEP [1]: Integration based on the unification link b_1

The unified entity B12 is designed by the integration of entities B1 and B2, based on the unification link b_1. This integration increases the number of products (six products).

STEP [2]: Integration based on the unification link a_1

The unified entity A12 is designed by the integration of entities A1 and A2, based on the unification link a_1. This integration does not increase the number of products.

STEP [3]: Integration based on the unification link a_2a_3

The unified entity A123 is designed by the integration of entities A12 and A3, based on the unification link a_2a_3. This integration increases the number of products (eight products).

3.5 Unification Process of Family Plan B

As in the case of Family Plan A, the result of the unification scenario of the Product Family B is shown in Figure 7. The number of products increases and decreases (five, three, three, and two products), by the integration based on the unification links (b_3, a_3, d_2, and d_1d_3).

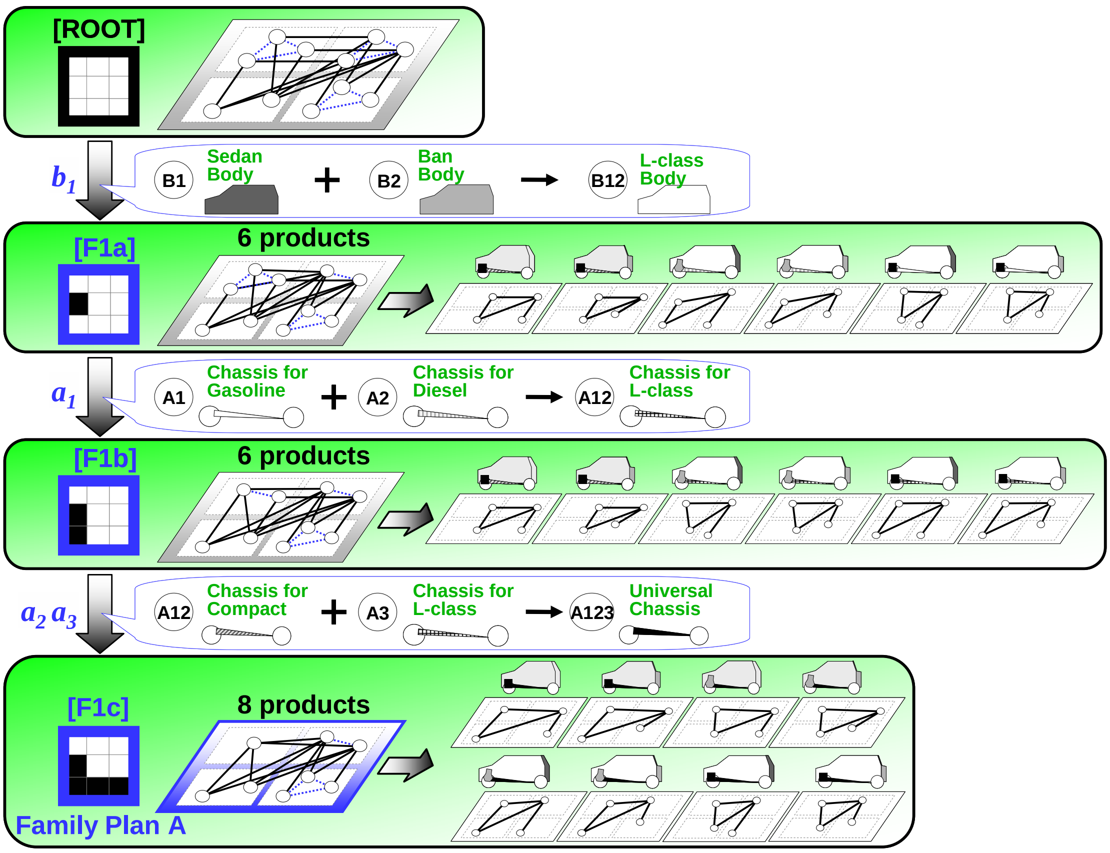

Figure 6. Unification Process (Family Plan A)

3.6 Platformization

The designer designs the platform module by the modularization of the entities that are commonly used in the product family. The module comprises entities that belong to different element in common architecture. Figure 8 shows the platformization result of Family Plans A and B (Figure 8 [3]) by selecting a platformization plan in the automatically calculated result of the platformization tree (Figure 8 [2]) based on the structure of the common architecture (Figure 8 [1]). The information processing procedures is described in detail as follows:

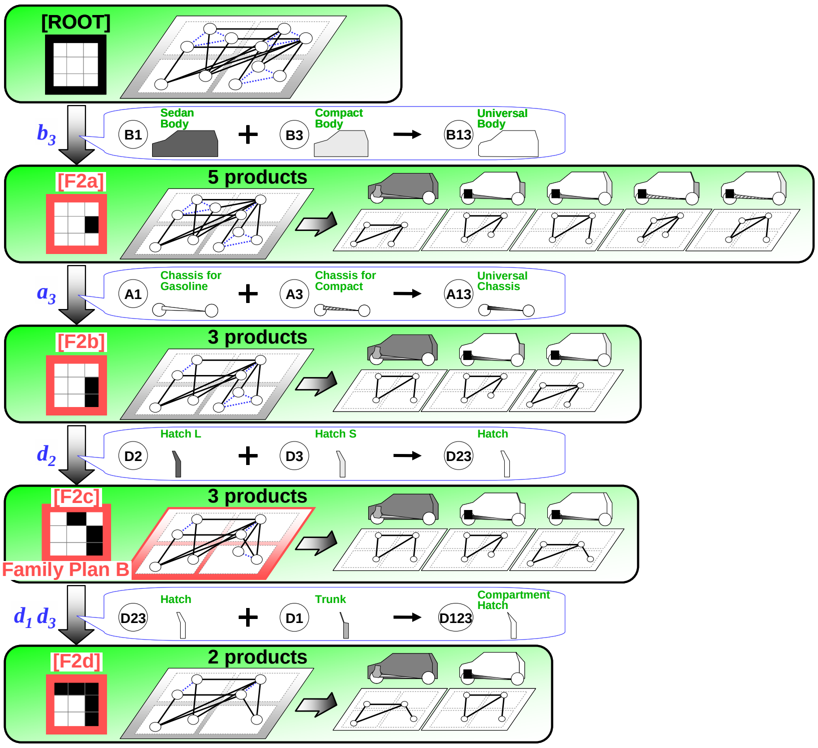

Figure 7. Unification Process (Family Plan B)

STEP [1]: Input structure of the common architecture

The design system calculates the structure of the common architecture (Figure 8 [1]). The architecture interfaces (f_1, f_2, f_3, and f_4) that connect the common architecture elements are defined based on the connection links between the entities.

STEP [2]: Calculation of the platformization graph

The design system automatically generates a platformization tree (Figure 8 [2]). this platformization tree represents the space of the platformization plans. It can be generated based on the cut-set of the common architecture graph (Figure 8 [1]). This tree consists of the platformization plan nodes and platformization links. A platformization plan node is defined as a matrix that comprising architecture interfaces. A black painted cell in the platformization matrix represents an architecture interface that has been eliminated by the platformization operation.

STEP [3]: Platformization based on the platformization plan node

The platform module design is realized by selecting one platformization plan node in the platformization tree. Figure 8 [3] shows the platformization result of Family Plans A and B. In Family Plan A, the platform that integrates the body and underbody is obtained. Platform A-1 for the compact car is designed by integrating the body for compact car and the universal chassis. Similarly, platform A-2 for the L-class car is designed by the integrating the body for the L-class car and the universal chassis.

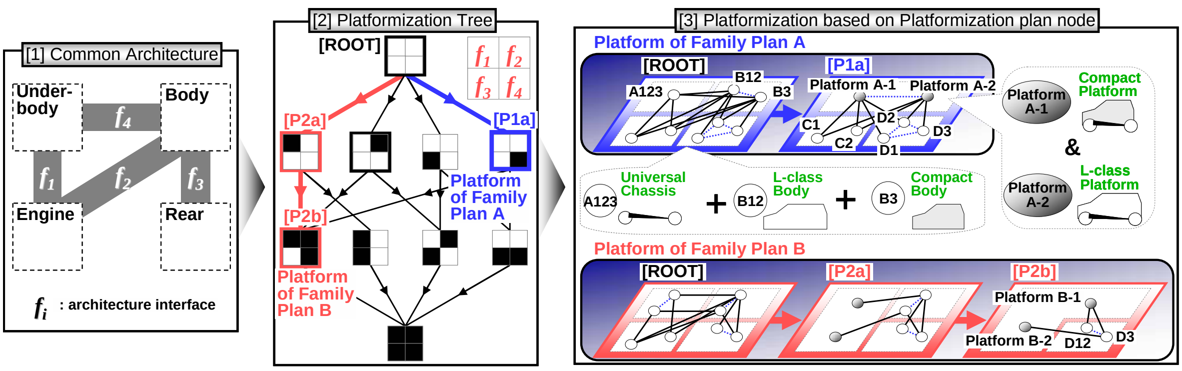

Figure 8. Platformization

A guideline for platformization is shown in Figure 9. The design system calculates the modularization (Figure 9 [3]) and platformization (Figure 9 [4]) points as the platformization guideline. High modularization and platformization points defined in the connection links represent the recommended link for the modularization and platformization by the design system. The wide lines in Figure 9 [3] and [4] represent the connection links with high points. The modularization point is calculated by adding the modularity of each module (Figure 9 [2]). The modularity indicates the coincidence of life-spans and functional independency. All modules can be automatically calculated by the design system from the cut-set of the product graph. The platformization points are defined as the summation of all modularization points in the product family model.

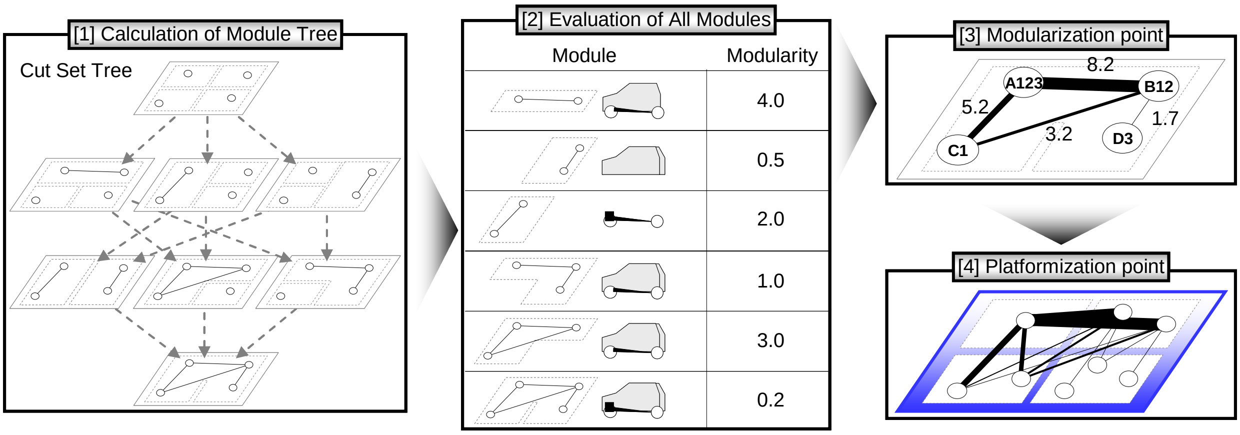

Figure 9. Platformization and Modularization Guideline

3.7 Evaluation

The design system evaluates the product family plans. Each product family plan is evaluated based on the following three aspects: (1) variety of product family, (2) design change cost, and (3) production cost. The platformization value depends on impact of the change in the product family value. The change in the product family value can be expressed by the following equation:

$$\textit{Platformization value} = \textit{Variety of product family} / (\textit{Design change cost} + \textit{Production cost}) \qquad (10)$$

Evaluation [1]: Variety of product family

The variety of the product family is calculated by estimating the positioning of each product in the market. For example, Family Plan A has eight products (Figure 6). Each product has the cover area in the market as one of its attributes. The variety of Family Plan A is evaluated by the positioning of these eight products.

Evaluation 2: Design change cost

The cost of design changes is estimated based on the unification operations. New product components and parts are required for the integration of the entities. Hence, new costs for the development of the integrated entities are estimated. For example, Family Plan A has four integrated unification links (a_1, a_2, a_3, and b_1) and two integrated entities (B12 and A123). The cost of design change is estimated to be the summation of the integration cost of the four links and the design cost of the two entities.

Evaluation 3: Production cost

The production cost is estimated based on the number of modules. Unification and platformization reduces the number of modules in the product family. The number of modules is estimated to be the number of entities in the platformized product family model. The entities include parts, integrated parts and platforms. For example, the platformization plan [P1a] in Figure 8 [3] has seven nodes, therefore the number of the modules is seven. Similarly, the platformization plan [P2b] in Figure 8 [3] has four modules. The production cost can be reduced by the mass effects based on the reduction in the number of modules. Hence, the proposed method estimates the production costs based on the size of modules.

4 PROTOTYPE SYSTEM AND DESIGN EXAMPLE

4.1 Prototype System Implementation

Based on the explanations in the previous sections, we developed a prototype design system using the object-oriented language VisualWorks Release 7.3 (CINCOM Smalltalk). Figure 10 shows the overview of the prototype design system. The prototype system consists of four design tools: [1] product family modelling tool, [2] parts and interfaces unification tool, [3] platform module design tool, and [4] product family evaluation tool. Each window displays the design results of the platformization of the car family. The detailed functions of each tool are as follows:

[1] Product family modelling tool: This tool supports the design of the product family model by adding product graphs, defining a common architecture and connecting unification links.

[2] Parts and interfaces unification tool: This tool supports the unification of parts and interfaces. The unification tree is calculated automatically generated from the product family model. The product family model and the line-up on each unification plan are provided by this window.

[3] Platform module design tool: This tool supports the modularization and platformization. The platformization tree is generated and the list of all modules is provided by this window.

[4] Product family evaluation tool: This tool supports the visualization of the evaluation result of the product family plans. Each product family plan is evaluated based on its variety and cost. The designer can select an appropriate product family plan that matches the product family strategy on this window.

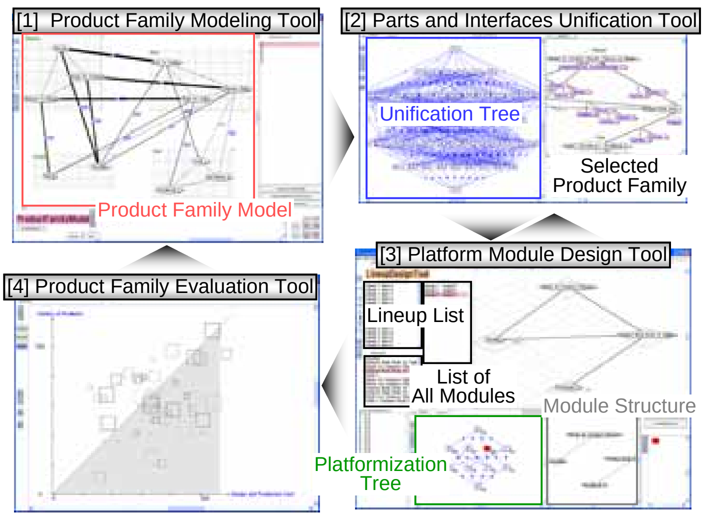

Figure 10. Prototype System and Example

4.2 Result and Discussion

Both of the Family Plans A and B in Figures 5,6,7, and 8 have higher family values than current family. However, they have significantly different strategic directions. The biggest advantage of Family Plan A is the variety of its products. The cover area of the market is very large because of the number of the products in the line-up (eight products, Figure 6 [F1c]). However, the reduction in the number of the modules (seven) is not as much as that in Family Plan B (Figure 6 [3] [P1a]). Hence, Family Plan A has a high variety of products, but its production cost is also high. To the contrary, the biggest advantage of Family Plan B is the reduction in the production cost. The number of the modules in Family Plan B is only four as shown in Figure 8 [P2b]. However, the Family Plan B has only three products in its line-up (Figure 7 [F2c]). Hence, Family Plan B reduces the production cost significantly, but its variety of products is not very high. The comparison between Family Plan A and B is shown in Table 1.

Table 1 indicates that the designer can design an appropriate product family plan that reflects the product family strategy, by comparing with the alternative candidates from the viewpoints of the variety of products and the total cost of the product family. Hence, the platform module design method proposed in this research can definitely improve the product family value by balancing of the variety and cost.

Table 1. Comparison between Family Plan A and B

	Product Family		
	Current	Family Plan A	Family Plan B
Lineup products	3	< 8 >	3
Modules	12 A1 B1 C2 D1 A2 B2 C1 D2 A3 B3 C2 D3	> 7 C1 C2 D1 D2 D3 Platform A-1 Platform A-2 >	4 D12 D3 Platform B-1 Platform B-2

5 CLOSING REMARKS

Conclusions

In this paper, we propose a design method of the platform module taking into consideration the variety and cost of the product family. The design system can assist the designer in comparing the platformization plans by automatic calculation of the solution space of the product family, and by estimating the variety of products and the design and production costs. The to-be design method of platform module design is proposed by comparing the platformization plans while considering various constraints and the growth of the entire product family. The integration of interfaces and the introduction of the use of common architectures are effective in the automobile and computer industries. In the near future, the proposed platform design method will be effectively used to design a to-be model of a common architecture and a direction of the interface integration in the software development industries, service industries, and lifecycle systems industries.

Future Studies

As future studies, an introduction of a multi-phased platform design method and an expansion for a product life cycle design method are required to investigate.

REFERENCES

[1] Neison S., Parkinson M. and Papalambros P. Multicriteria Optimization in Product Platform Design. *ASME Journal of Mechanical Design*, 2001, Vol. 123, 199-204.

[2] Simpson T. et al. Development of a Framework for Web-based Product Platform Customization. *ASME Journal of Computing and Information Science in Engineering*, 2003, Vol. 3, 119-124.

[3] Raghothama S. and Shapriro V. Topological Framework for Part Families. *ASME Journal of Computing and Information Science in Engineering*, 2003, Vol. 2, 246-255.

[4] Fujita K. and Yoshida Y. PRODUCT VARIETY OPTIMIZATION: SIMULTANEOUS OPTIMIZATION OF MODULE COMBINATION AND MODULE ATTRIBUTES. In *Proceedings of the ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference*, DETC2001-21058 (CD-ROM).

[5] Akundi S., Simpson T. and Reed P. MULTI-OBJECTIVE DESIGN OPTIMIZATION FOR PRODUCT PLATFORM AND PRODUCT FAMILY DESIGN USING GENETIC ALGORITHMS. Proceedings of *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, DETC2005-84905.

[6] Siddique Z. and Adupala R. PRODUCT FAMILY ARCHITECTURE REASONING. Proceedings of *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Long Beach, CA, USA, DETC2005-85340 (CD-ROM).

[7] Maier J. and Fadel G. STRATEGIC DECISIONS IN THE EARLY STAGES OF PRODUCT FAMILY DESIGN. Proceedings of the *ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference*, DETC2001-21200 (CD-ROM).

[8] Mantripragada, R., "Assembly Oriented Design: Concepts, Algorithms and Computational Tools", Ph.D. thesis, Massachusetts Institute of Technology, USA.

[9] Aoyama K. and Koga T. Step-by-Step Modular Design and Management of Modular Interface. Proceedings of the *Design Engineering Workshop* 2005, 29-37

Contact: T.Koga
The University of Tokyo
Department of Environmental and Ocean Engineering
7-3-1 Hongo, Bunkyo-ku
113-8656, Tokyo
Japan
Phone: 03-5841-6504
Fax: 03-3815-8364
e-mail: tsuyoshi-koga@msel.t.u-tokyo.ac.jp