ELSEVIER

Advanced Engineering Informatics 16 (2002) 107–125

ADVANCED ENGINEERING INFORMATICS

www.elsevier.com/locate/aei

Estimate design intent: a multiple genetic programming and multivariate analysis based approach

Yoko Ishino*, Yan Jin

IMPACT Laboratory, Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089-1111, USA

Received 12 April 2001; accepted 26 November 2001

Abstract

Understanding design intent of designers is important for managing design quality, achieving coherent integration of design solutions, and transferring design knowledge. This paper focuses on automatically estimating design intent, represented as a summation of weighted functions, based on the operational and product-specific information monitored through design processes. This estimated design intent provides a basis for us to identify the evaluation tendency of designers' ways of doing design. To represent and estimate the design intent, we introduced a *staged design evaluation model* as a general yet powerful model of design decision-making process, and developed a methodology for estimation of design intent (MEDI) as a reasoning method. MEDI is composed of two basic algorithms. One is our newly introduced multiple genetic programming (MGP) and the other is statistical multivariate analysis including principal component analysis and multivariate regression. The characteristics of MEDI are; (1) principal component analysis provides approximate evaluation of how much preferable a specific product model is, assuming the final product model (or design) is the most preferable one; (2) MGP enables us to simultaneously estimate both structure of target performance functions and the approximate values of their weights for a domain of design problems; and (3) multivariate regression readjusts the approximate weights obtained by MGP into more accurate ones for specific design problems within the domain. Our framework and methods have been successfully tested in a case study of designing a double-reduction gear system. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Design process; Design intent; Genetic programming; Multivariate analysis

1. Introduction

In recent years, engineering design projects have grown larger in scale and more complex in contents. In practice, a design task is usually divided into a number of highly coupled sub-tasks that require multiple designers to work together collaboratively. It is common for a group of designers to iteratively design different versions of the same artifacts such as car and electrical appliances. Under these conditions, understanding other designers' intent is important for managing the effectiveness and efficiency of collaboration and for achieving high quality of the overall design. Furthermore, knowing expert designers' intent can help knowledge transfer. For instance, knowing what guided an expert designer during his/her design process may provide insights for new designers to improve their design, especially when the new designers deal with the same or similar design tasks as the expert designer did. Design intent signifies why an object is designed the way it is. It is related to a designer's decision mechanism and reflected through his/her design process. In other words, design intent of a specific artifact designed by a specific designer can be viewed as the decision criteria employed by the designer for making design decisions during the process of designing the artifact.

While viewing designers' decision criteria as their design intent opens new ways for acquiring design intent knowledge, explicitly representing and capturing the intent is still a challenge. It is usually hard for designers to be willing to express their decision criteria since it takes time and distracts their design work. Even if a designer is willing to take time to do so, expressing *quantitative*, *subjective*, and *practical* criteria can be very difficult. For example, a designer may express that "I focused on reducing the total cost first and then tried to increase safety." We cannot exactly understand from this statement how much quantitatively the designer cares about cost and safety since the statement is a qualitative one. Moreover, some components of design criteria such as safety are very vague. This kind of components does not exist explicitly as itself in the design process, and designers usually judge them from several

* Corresponding author. Tel.: +1-213-740-5328; fax: +1-213-740-6668.
E-mail address: okinaka@usc.edu (Y. Ishino).

1474-0346/02/$ - see front matter
PII: S1474-0346(01)00005-2

lower level items. Take safety as an example, lower level items such as material selection and load balance might be synthetically considered into safety. This synthetic thinking process often makes it difficult to analyze what is the design criterion that the designer employed.

Generally speaking, a designer should take a normative decision-making approach and clearly define his/her design objective and value function for making design decisions [8]. In case of complex design problems, multi-objective decision approach [11] can be applied. Practically, however, the normative approach is hardly working partly due to the complexity of design problems. In our research, we believe that designers' intent (i.e. evaluation criteria) is embedded in their design processes. Our goal is to automatically estimate or extract design intent based on the data recorded from the design process, without interrupting designer's normal design activities. To achieve this goal, a new approach is needed to support identifying design intent that captures quantitative, subjective, and practical criteria used by designers to evaluate design alternatives. This estimated design intent provides a basis for us to understand the evaluation tendency of a designer in his/her design.

One major problem we face in achieving our goal is that the decision mechanism in design processes is often ill-structured and complex. Furthermore, the quantity of usable data is small, as contrasted with other fields such as marketing [2]. To address the first problem, a general yet powerful model is needed to represent the decision mechanism in design. We have introduced a *staged design evaluation model* for this purpose. This model provides a way to bridge the gap between the evaluation criteria and actual design parameters. In addition to this model, introduce the following assumptions:

1. The design evaluation criteria for a specific engineering domain or a type of design tasks have certain additive functional structures such as *utility function* found in Utility theory [11]; i.e.

   $$\text{criteria} = \sum_i w_i F_i,$$

   where F_i is a constituent function of the criteria, i identifies a specific constituent, and w_i is the weight coefficient for F_i.
2. The form of functions F_i can be identified from a pool of process information generated by a group of designers working on the same domain or same type of tasks.
3. The differences between the evaluation criteria of individual designers are in the values of weighting coefficients, i.e. w_i in the equation cited earlier, associated with the functional structure.

By introducing the earlier three assumptions, the problem of estimating design intent is formulized as a problem of finding 'what the factors F_i are' and 'how much the coefficients w_i weigh.' This is an inverse problem whose objective is to search the best approximation of the function from observed data sets.

We propose a methodology for estimation of design intent (MEDI) to solve this problem. MEDI is composed of two basic algorithms. One is our newly introduced multiple genetic programming (MGP) and the other is statistical multivariate analysis including principal component analysis and multivariate regression. When MEDI is applied, the values of weighting coefficients (w_i) can be uniquely determined by minimizing total sum square error between estimated values and observed ones, based on the condition that the values of each factors (F_i) are normalized at every data set, e.g. the average of those should be 0.0 and the standard deviation of those should be 1.0.

This paper is organized as follows: Section 2 clarifies design evaluation process and introduces a generic model of evaluation called staged design evaluation model. Section 3 provides a step-by-step presentation of MEDI. In Section 4, we describe a case study where the proposed model and methods are applied to a practical double-reduction gear system design problem. Sections 5 and 6 discuss the related work and conclusions, respectively.

2. Staged design evaluation model

2.1. Design process

A designer's design process usually consists of following steps. First, after clarifying a given design task, the designer adopts an *evaluation policy* based on the design requirements and his/her design experience. Here we define evaluation policy as a qualitative strategy to carrying out a given design task. For example, a designer may decide "to target on reducing the total cost first and then attempt to increase safety, without violating the given constraints." After the evaluation policy is either explicitly or implicitly set, the designer starts to create design alternatives. In this paper, we call a completed design alternative a *product model.* After a product model is created, the designer evaluates the product model based on how its performance matches the evaluation policy. If the product model does not satisfy the designer, he/she adjusts and refines certain design parameters to meet his/her design target. It should be mentioned that a designer's evaluation policy could be revised during the design process depending on what the designer finds through the design process. This cycle of creation and evaluation is repeated until the designer obtains the final design solution that completely meets his/her design preference.

The earlier description suggests that (1) design processes can be generally viewed as trial-and-error, and the evaluation follows every creation of new design alternatives; (2) evaluations on design alternatives are made based on

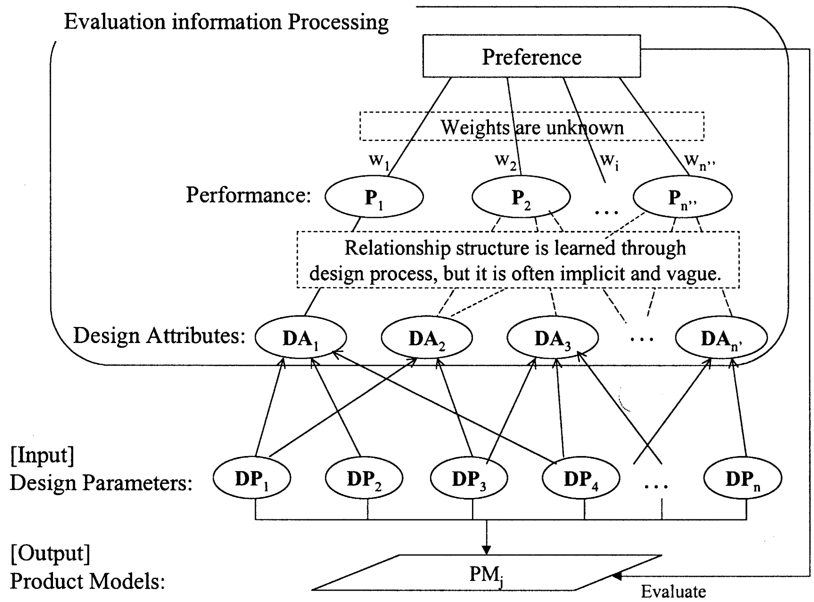

Fig. 1. Staged design evaluation model.

designers' own preference which is formed by synthesizing several, and often competing, factors; and (3) the finally selected design alternative can be regarded as the ideal one for the designer. These ideas have led us to introducing a staged design evaluation model to represent general decision mechanism in design process.

2.2. Staged design evaluation model

We propose a general model, called staged design evaluation model, to represent evaluation process in design, as shown in Fig. 1. This model indicates that although there is an information gap between the evaluation level preference and the actual operation level decisions, a staged information-processing system can be developed to bring together the missing links. The staged design evaluation model consists of four layers of information, namely, design parameter, design attribute, performance, and preference. We define terminology as follows:

Design parameter. Any basic factor of an object being designed whose value can be manipulated by a designer directly, e.g. geometric characteristics, such as position and length, and material of components.

Design attribute. Any property of an object being designed whose value is determined by a numerical value of design parameters, e.g. cost, weight, volume, and stiffness.

Performance. Any target function with regard to effectiveness, e.g. cost- performance, size-performance, stability, and durability.

Preference. An inclusive index that represents how well for a designer a product model goes.

Product model. A complete solution for a design task. This is also called a design alternative.

Based on our proposed model, a design process can be described as follows.

After designers start designing, they generate and configure design attributes and design parameters based on given design requirements, e.g. 'Total weight $\leq$ 1000 kg.' The designers set their evaluation policy along with their design goals using performance concepts, e.g. "target on reducing the total cost first and then try to increase safety, without violating the given constraints." Here, reducing total cost means to pursue cost performance. Both cost performance and safety are performance concepts. At this stage, designers usually cannot directly link the performance concepts to the values of design parameters, since a design parameter has relationships with more than one design attribute. Moreover, although designers qualitatively know which design attribute affects which performance, they cannot exactly know how much impact each design attribute has upon the performance. During design, designers directly operate the values of design parameters, manage the values of design attributes, produce a product model, and then, evaluate the product model. If the product model is not good enough to satisfy the designers, they repeat the process of creation-and-evaluation. However, even after finishing designing, it is often difficult for designers to concretely explain their evaluation process, i.e. how all design attributes were quantitatively related to performances and how much did they weigh each performance. Designers often obtain some knowledge about relationships between preference, performance and design attributes through design process, but it is fragmentary and not well-organized. For instance, a designer may know that both body stiffness and coating material are related to the durability, but he/she is often not consciously aware of how these design attributes are synthesized into the durability in his/her mind.

While it is often difficult to predict the performance of a design without using sophisticated analytical methods and tools, we believe that designers are making rough predictions through certain synthetic information-processing in their mind based on their experiences. They create product models by defining and manipulating design parameters and predict the performance of the product models by mentally linking design parameters with design attributes, and design attributes with performance. The research challenge we face is that "can we extract designers' implicit evaluation process and criteria from their design operational information and make it explicit?"

In our research, we assume the following.

1. Designers' evaluation criteria can be represented as a preference function of summation of weighted performances.
2. The functional form of each performance in the preference function depends more on design domain and task than on designers, since the relationships between performances and design attributes are mostly objective and physics based.
3. The weight of each performance in the preference function depends on individual designers, since it is the designer who determines how important each performance is in his/her design based on his subjective judgment.

Based on the earlier assumptions, we introduce the following relationships among preference, performances, and design attributes.

$$R_j = \sum_i w_i P_i \tag{1}$$

$$P_i = f_i(\mathrm{DA}_1, \mathrm{DA}_2, \dots, \mathrm{DA}_n) \tag{2}$$

where R_j is the preference, a value which represents the relative desirability of a product model, $j = 1\dots u$, where the number of u are the number of total product models, P_i, the individual performance, $i = 1\dots m$, where the number of m depends on a design problem, w_i, the weighting factor assigned to each performance P_i, DA_k, the individual design attribute, $k = 1\dots n$, where the number of n depends on a design problem, and f_i is the individual function which expresses the relationships between each performance P_i and all design attributes which relate to P_i.

We call Eq. (1) a *preference equation*, and the right side of the preference equation signifies an *evaluation function* of designers. Each function f_i in Eq. (2) is called a *performance function*, and the weighting coefficient w_i in Eq. (1) is called a *performance weight* of the performance function. Since we usually deal with situations where multiple designers perform the same or similar design task, there can be two types of evaluation functions. One is generated based on a pooled data set generated by multiple designers a domain, and the other is obtained based on individual designers' data set which is a subset of the pooled data set. We call the former a *domain specific evaluation function*, and the latter an *individual specific evaluation function*. Although only one domain specific evaluation function is obtained from a pooled data set, the number of individual specific evaluation functions corresponds to the number of participating designers. Both types of evaluation functions on the same design task have the same performance functions i.e. f_i's, and the differences in design intent are displayed in performance weights i.e. w_i's. In our research, we regard the magnitude of the normalized performance weight (w_i) as the index of how much designers considered the corresponding performance in their design. Eventually, according to Eqs. (1) and (2), our problem of estimating design intent becomes acquiring a set of performance functions f_i and sets of performance weights w_i in both the domain specific evaluation function and a number of individual specific evaluation functions. We propose a MEDI as a solution.

Before moving to the explanation of MEDI, it is worth mentioning that the form of preference Eqs. (1) and (2) are consistent with the evaluation function found in multi-objective decision theory or utility theory. Utility theory was originally developed by von Neumann and Morgenstern [20], with later development by many other researchers. Keeney and Raiffa published the standard reference text of multi-objective decision approach [11]. In general, the *utility* of a given alternative is the benefit or overall satisfaction for a decision-maker with a specific objective, e.g. to minimize cost, to pursue. When a decision-maker has multiple objectives, e.g. to minimize cost and maximize quality, then the overall utility of a specific alternative can be derived by a summation of weighted utilities of individual objectives given certain conditions satisfied [11]. The concept of performance in our model corresponds to the concept of objective in decision theory. As will be described later, the normalized performance measures can be viewed as utilities of individual objectives. The preference of Eq. (1) corresponds to the weighted multi-objective utility function for evaluating specific alternatives, i.e. product models.

In case of engineering design, designers have their own design objectives. Therefore they know which performance should be considered. As will be described later, in our model, we allow designers to choose their performances from the candidates that are generated by MEDI. Although designers have knowledge about what their performances are, the knowledge is incomplete and MEDI helps them formulate it. The thing that is not obvious for designers is the structure of performance functions in Eq. (2). To estimate design intent using Eq. (1), we need to develop ways to find performance functions in Eq. (2).

I) Calculate each quantitative Preference value of each Product Model
Method: Principal Component Analysis

II) Determine Performance Functions and the approximate Performance Weights
Method: Multiple Genetic Programming (MGP)

1) Search candidate solutions for Performances P_i and their weights w_i
1-1) Preliminary (by human)
• Itemize Performance P_i
• Classify Design Attributes (DA_i) into a specific Performance
1-2) Execution of MGP (by machine)
• Produce candidate solutions of P_i structures and w_i values
2) Determine the P_i structure (by human)
• Select an appropriate set of P_i structures out of candidates

III) Determine the accurate values for Performances
Method: Multivariate Regression

1) Recalculate a set of accurate weights w_i from the normalized data of all designers
$\longrightarrow$ *domain specific evaluation function*

2) Calculate sets of weights w_i from the normalized data of every designer
$\longrightarrow$ *individual specific evaluation functions*

Fig. 2. Framework of MEDI.

3. Methodology for the estimation of design intent

3.1. Framework of MEDI

As mentioned earlier, estimating design intent is to determine how much a designer takes each performance into account while designing. To approach this problem, the following issues need to be addressed:

1. calculate each quantitative preference value of each product model,
2. determine the structures of all performance functions, and
3. determine the sets of performance weights, i.e. ones in both domain specific evaluation function and individual specific evaluation functions.

These are not easy tasks. Especially, issue (2) has many difficulties because many design attributes exist in a usual design task and the number of the combination of them may run into astronomical figures. Utilizing designers' knowledge about performance functions can help dealing with this problem. In the following, we propose a MEDI as a reasoning method to address the earlier three issues. The framework of MEDI is indicated in Fig. 2.

In order to calculate preference value for each product model, one of statistical multivariate analysis methods called principal component analysis is employed. To address issue (2), a novel method named MGP is introduced. When MGP is applied, the designer who did the design task first enumerates what target concepts, i.e. categories of performances, exist in the design task, and assigns design attributes into their mostly related performance categories. We assume that designers do know which design attribute mostly relates to which performance, although they may not know how they are related. Next, the novel evolutionary computing, MGP, is employed. MGP produces several candidate solutions of a set of performance structures and their approximate weights. Then, designers collectively select the best set of structure of performances. In this process, designers can clarify what their exact performances are while seeing candidates created by MGP and discussing them with other designers. MGP effectively helps the designers by providing candidate performance structures and performance weights. These candidates serve as the basis for eliciting the right performance structures.

Lastly, to address issue (3), multivariate regression is used to calculate the precise values of the weights from the data that is normalized where each performance determined through MGP is regarded as a new variable. The set of performance weights in a domain specific evaluation function is acquired from the normalized data of all designers, and the set of them in each individual specific evaluation function is from the data normalized on each designer's basis.

To summarize, the main points of our methodology are; (1) principal component analysis enables us to approximate the quantitative value of preference of each product model being explored; (2) MGP enables us to estimate the structures of performance functions together with the approximate values of their weights; and (3) multivariate regression readjusts the approximate weights obtained by MGP into the accurate ones. Although MGP is not guaranteed to find the optimum solution, it produces a good approximation through evolutional generate-and-tests. The following subsections describe these methods.

3.2. Multivariate analyses in MEDI

Multivariate analysis consists of a collection of methods that can be used when several measurements are made on each individual or object in one or more samples [19]. Historically, a large number of applications of multivariate techniques have been in the behavioral and biological sciences. However, interest in multivariate methods has now spread to other fields of investigation including education, physics, engineering, and psychology. The availability of multivariate techniques and inexpensive computing power also broadened its application. Both principal component analysis and multivariate regression belong to multivariate analyses. Principal component analysis is usually employed to summarize the given information. On the other hand, multivariate regression is used to predict the trends of one dependant variable from multiple independent variables.

3.2.1. Principal component analysis in MEDI

In principal component analysis, maximizing the variance of a linear combination of the variables is executed to summarize the given information [19]. Geologically, this method makes the axes rotated in the multiple-dimensional space formed by the multiple variables so that a smaller number of variables can be found through linearly combining the original set of multiple variables. The new smaller set of variables, which are conventionally called components, has the equivalent power to represent the given information. Moreover, the components are completely independent to one another. Algebraically, this method calculates eigenvalues and eigenvectors of a covariance matrix formed by given multiple variables. The eigenvalues represent variance of components and the eigenvectors represent the synthesizing weights of the components. This method usually cannot provide meaningful interpretations of the components. However, it has the merit of reducing the number of dimensions in the multiple-dimensional space if there are high correlation and redundancy between given dependent variables. We utilize this merit of the method to measure the distance between product models.

In our model, to estimate w_i and f_i in the Eqs. (1) and (2), the data, i.e. pairs of preference R and design attributes DA_i, should be given as input. Although all design attributes can be monitored through design processes by using a customized CAD system for a design task, the value of preference is difficult to obtain from a design history. We introduce the concept of *virtual distance*. If the virtual distance between a given product model and the final or the preferred product model can be calculated, then the distance can be used to approximate the values of the designer's preferences of that product model. To make this distance calculation possible, we need to introduce a Euclidean space with orthogonal dimensions and coordinates in which the 'position' of a given product model can be identified. Because the original multiple variables describing the product model are usually not independent to one another. For finding orthogonal coordinates to calculate the distance, principal component analysis is employed.

We view all design parameters and design attributes of each product model as an original set of multiple variables that are often highly correlated and even redundant to each other. After principal component analysis is applied, the new components are synthesized that are independent to each other. These components form orthogonal coordinates that define the position of a product model. Settling a cutoff point for eigenvalue of the components is effective to reduce the number of dimensions and in practice, it is usual to set the value on 1.0. The distance (D) of a Product model is defined as follows:

$$D_i = \sqrt{\sum_j (C_{i\cdot j} - C_{\mathrm{ideal}\cdot j})^2} \tag{3}$$

where D_i is the distance between the i-product model and the ideal product model that is selected as the best through the design process, $i = 1\ldots u$, where the number of u is the number of product models, $D_{\mathrm{ideal}} = 0$, and C_{ij} is the score of the i-product model on the j component which is made by principal component analysis, $j = 1\ldots v$, where the number of v is the number of synthesized components.

After the distance D is calculated the conversion of the distance into the index of preference is conducted under the assumption that the product model which is created as the final solution by a designer would be the best or the ideal one for the designer.

$$R_i = \mathrm{Normalize}(\mathrm{Const} - D_i) \tag{4}$$

$$R_i = \frac{\dfrac{\sum D_i}{u} - D_i}{\sqrt{\dfrac{u\sum D_i^2 - \left(\sum D_i\right)^2}{u^2}}} \tag{5}$$

where R_i is the preference value on the i product model, $i = 1\ldots u$, where the number of u is the number of product models, Const, a constant to convert a distance into preference value, yet this Const is canceled in the normalizing process, and Normalize is a function normalizing a variable whose average should be 0.0 and whose standard deviation should be 1.0.

3.2.2. Multivariate regression in MEDI

Multivariate regression is a useful way in statistics to predict one Y, called the dependent or response variable, from several X's, called independent variables. In the multivariate regression model, we express Y as a linear function of the X's:

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_pX_p \tag{6}$$

where p is the number of independent variables, and b_j is the regression coefficients, $j = 1\ldots p$.

Here, a random error, e, which is a probabilistic variable that means the difference between a predicted value and an actual measured value of the dependent variable, is introduced to illustrate Eq. (6) by each observation.

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_p x_{ip} + e_i \tag{7}$$

where y_i is a sample of q observations of Y, $i = 1 \ldots q$, where the number of q represents the number of total observations, and x_{ij} is a sample of q observations of X, $i = 1 \ldots q$, where the number of q represents the number of total observations, and $j = 1 \ldots p$, where the number of p represents the number of independent variables.

Multivariate regression usually obtains the values of regression coefficients using least squares estimation, which is equivalent to minimizing total sum square error between estimated values and observed ones [19]. Therefore, we have the following equations.

$$\sum_i e_i = 0 \tag{8}$$

$$\sum_i x_{ij} e_i = 0, \qquad (j = 1, \ldots, p) \tag{9}$$

The regression coefficients, $b_0 \ldots b_p$, can be obtained by solving Eqs. (7)–(9). It is important to point out that to compare the influence of individual independent variables X's upon Y, all X's should be normalized to make the average value and standard deviation of each X the same (usually average is 0.0 and standard deviation is 1.0).

We use this least squares estimation technique for the fitness calculation in MGP. To estimate how much each designer weighted individual performances in their design, we apply the multivariate regression technique again after fixing the structure of performances through MGP to compare their regression coefficients.

3.3. Multiple genetic programming

3.3.1. Genetic programming

Genetic programming (GP) is one of the evolutionary computation methods and was introduced by John Koza as an extension of the concept of genetic algorithms (GA) [12]. Motivated by an analogy to biological evolution, GA provides a learning method based on biased subsequent search using genetic operations, such as mutation and crossover [6]. In common GA, bit strings or symbolic descriptions are used for representing hypotheses. GP succeeds the core idea of GA, but uses tree-structure or graph-structure to represent hypotheses so that GP can deal with structural problems, such as program language, function, and concept-tree. GA and GP can be viewed as general optimization methods that search a large solution space. Although not guaranteed to find an optimal object, GA and GP often succeed in finding an object with high fitness.

Because of its representation power, GP has been applied to produce interesting and successful results in many different applications [9,10,13]. One active area of GP application is to find solutions of inverse problems and effective incorporation of subroutines. The performance of GP largely depends on the choice of representation and on the choice of fitness function. The weak point of GP is the high calculation cost due to the huge size of the hypothesis space it must search.

Our problem, estimating design intent as the summation of weighted functions, is regarded as one of the inverse problems. More specifically, our problem is a system identification problem, i.e. to estimate the behavior of a system based on the pair of input and output data. There are a variety of system identification problems, such as pattern recognition problem and time-series estimation [1]. GP is one of the most effective methods for solving such system identification problems. In order to efficiently apply GP to our system identification problem whose partial structure is given, we propose a method called multiple GP, or MGP for short.

3.3.2. Algorithm of MGP

The MEDI partially includes a human–machine interactive process. Designers know well about their design processes. They know what performances need to be considered and which design attributes are related to which performance, although they may not know the exact relationships between them. In MEDI, designers first create a list of performances and link design attributes to relevant performances. After that, MGP is executed based on the given performances and the links between the design attributes and the performances.

Fig. 3 shows the core idea of MGP. MGP is a series circuit of GP modules, and each GP module undertakes the estimation of each f_i, respectively. We call a solution candidate a *chromosome*, which contains several genes that correspond to GP modules, respectively. The example of f_i ($i > 0$) is shown in Fig. 3. The f_i ($i > 0$) has tree representation whose nodes are divided into two types, one is a terminal, and the other is a non-terminal. The terminal nodes are design attributes linked to the performance function concerned, and constants from 0.000 to 1.000, e.g. 0.578. The non-terminal nodes are primitive functions such as *add*, *subtract*, *multiply*, and *divide*. A non-terminal node always has its arguments as its descendant nodes in a tree-structure. Each GP module uses an evolutionary search to explore the vast solution space. If f_i ($i > 0$) has only one design attribute as a terminal candidate, we call it a fixed gene and it is removed from the chromosome to save calculation cost. We assume that when designers classify only one design attribute into a performance, the performance function could be simply replaced by the design attribute with the condition that larger value of

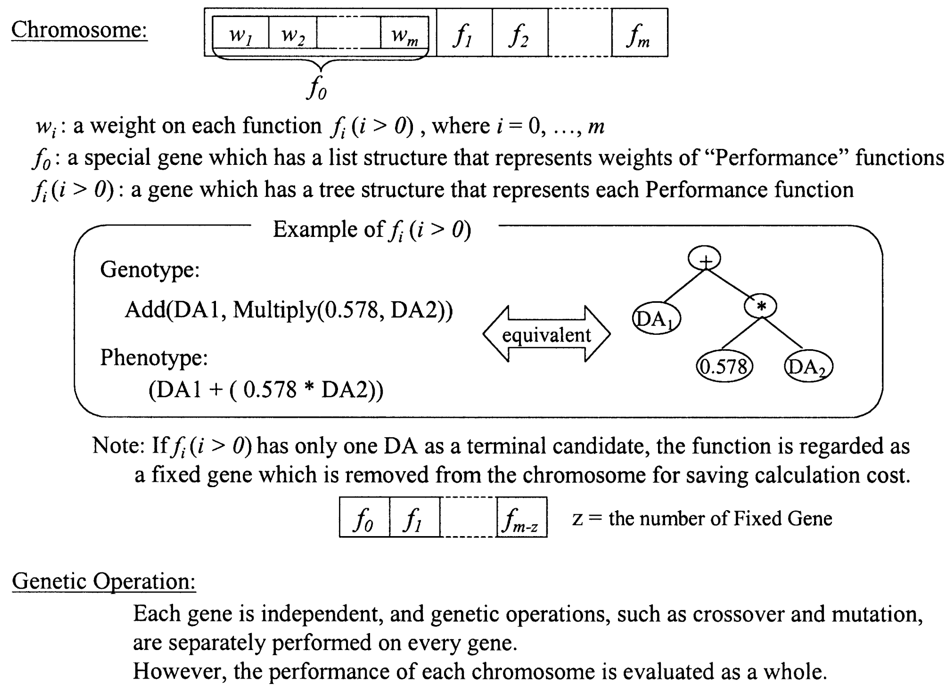

Fig. 3. Mechanism of MGP.

the design attribute means better performance for the design. If a design attribute has its value the smaller the better for the design, then the reciprocal of the design attribute is used as the performance function. On the other hand, f_0 has a specific feature that differs from f_i ($i > 0$). The f_0 has a list structure that represents weight values of Performance function f_i ($i > 0$). So, the genetic operator for f_0 behaves as simple GA.

Each f_i is treated as an independent gene, and genetic operations are independently performed between the same kinds of genes. However, the evaluation using the fitness function, which is described in the next sub-subsection, is executed as a whole individual chromosome. The rewarded evaluation value is divided equally among genes that constitute the chromosome. This type of rewarding action is a kind of simple experiential reinforcement learning with profit sharing [7].

The algorithm of MGP is shown in Fig. 4.

The process of MGP is as follows.

- Step1. Initialize a population
- Step2. Evaluate each chromosome in the population
- Step3. Distribute the evaluated value to each gene, f_i
- Step4. Create the next population based on evaluated value
- Step5. Repeat Step2 to Step4 until fulfill the closing condition

As in a common GP, the MGP maintains a population of individuals. In each stage of iteration, it produces a new generation whose population is as many as the first generation's by using selection, crossover, and mutation. It is calculated by the fitness function how good an individual chromosome is.

The main characteristics of MGP are (i) connecting more than two independent components of GP in series, (ii) synchronizing every GP's genetic operations, and (iii) rewarding each GP's component by using reinforcement learning.

3.3.3. Fitness function of MGP

Since each chromosome produces solution candidate for estimation of w_i and f_i, chromosomes are considered to be hypotheses. We have to evaluate how good each hypothesis is. We use *fitness value* to indicate how close the value predicted by hypothesis fits to an actual measured value. In MGP, the fitness value indicates how far away the value predicted by hypothesis is from the value obtained by principal component analysis. Therefore, smaller fitness value means better fit. The function that is used to determine fitness values is called a fitness function. The earlier mentioned least squares estimation technique is used in our fitness function, as is indicated in Eq. (10).

$$F = \sqrt{\sum_j (\hat{R}_j - R_j)^2} \tag{10}$$

$$\hat{R}_j = \sum_i w_i f_i \tag{11}$$

where F is a fitness value of a chromosome, where the closer

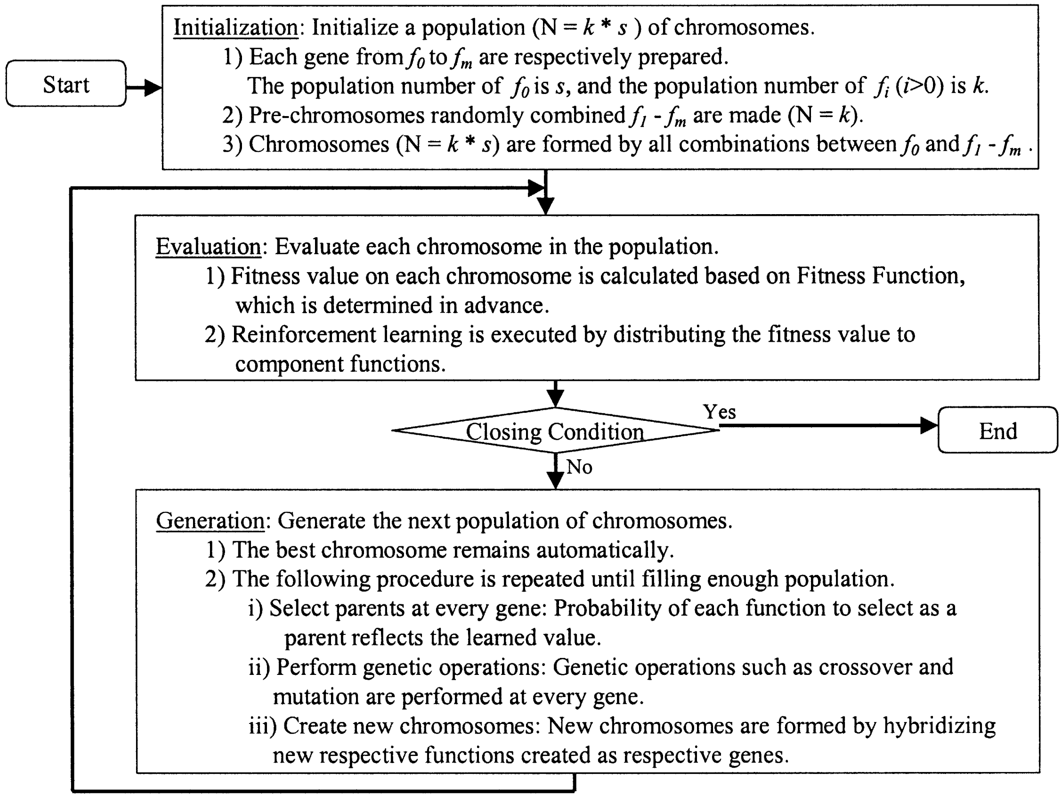

Fig. 4. Algorithm of MGP.

to zero is the better, R_j, a value measured by principal component analysis, $j = 1...r$, where r represents the number of product models, and $\hat{R}_j$ is a value predicted by a chromosome, $j = 1...r$, where r represents the number of product models.

3.3.4. Genetic operations of MGP

In MGP, the probability by which an individual chromosome will be selected to the next generation is given in inverse proportion to its fitness value towards the fitness of other chromosomes in the same population. That is because the fitness value in MGP has a reverse direction, as compared to an ordinary fitness value in usual GP. This method is called roulette wheel selection or fitness proportionate selection. Two individuals selected by this roulette wheel selection are viewed as parents, and then they are, respectively, divided into each gene (f_i). And such common genetic operations as crossover and mutation are executed in every gene that corresponds to every performance function. Although all genetic ways that are commonly used in GP can be used in MGP, a single-point crossover and a point mutation, which are the most common genetic operations in GA and GP, are adopted in our MGP. The probability of these genetic operations should be experimentally determined, e.g. single-point crossover is conducted in 80%, and point mutation is executed for inherited parent's gene in 10%. In each gene section, a cycle, in which basically two selected individuals produce two offspring candidates through genetic operations, is repeated until the number of offsprings reaches a regular population. After finishing all genetic operations in each GP module, new chromosomes are created by connecting respective produced genes in order of performance function. The chromosomes as offsprings in the next generation are put in the evaluation stage again. Thus, genetic operations are conducted in respective GP module, but generational changes are synchronized through all the process of MGP.

4. Case study

4.1. Double-reduction gear system

Our proposed methods were evaluated in a case study, design of 'double-reduction gear system.' The double-reduction gear system is composed of four gears, three shafts, bearings, and a case. Basically, the teeth number of the gears determines the speed reduction rate. Since the power of the revolution makes the torque and the bending moment, gears and shafts should be designed to stand up to the force. We developed the CAD system called 'Gear-CAD.' Gear-CAD is an integrated design environment that allows designers to access needed information through the CAD software and executes the basic technical computing

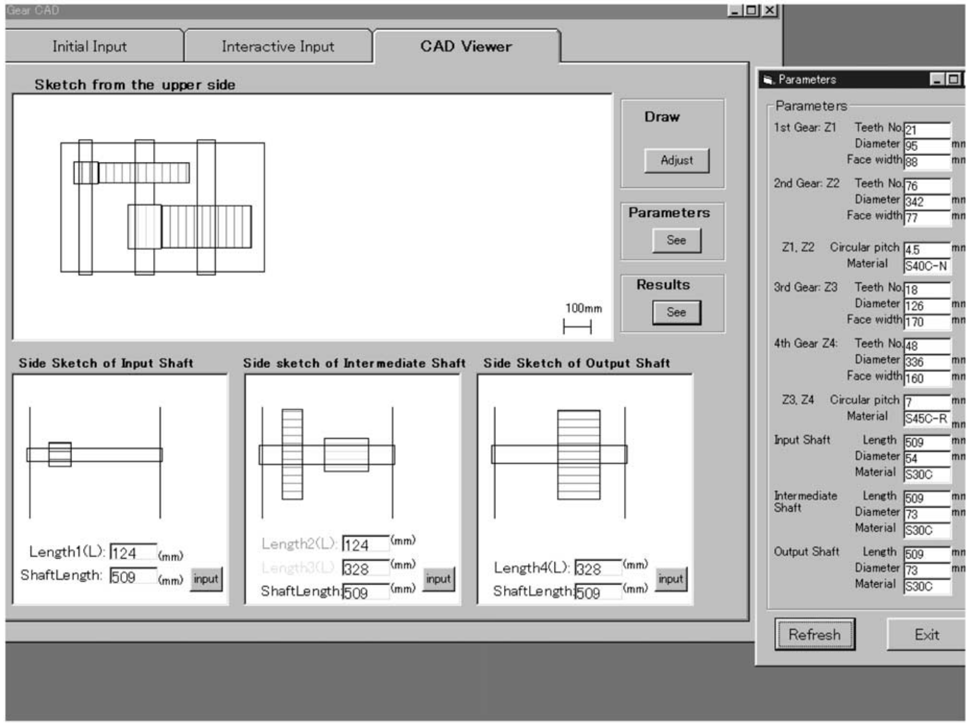

Fig. 5. Example screen of gear-CAD.

for the gear system. Gear-CAD supports design, and simultaneously monitors and records what designers do, i.e. all events designers generated during the design process. This double-reduction gear system has 29 design parameters, such as gear teeth number and a shaft length, etc., and 11 design attributes, such as total weight and total cost, etc. Gear-CAD also records these values in detail through the design process.

Our monitoring system including Gear-CAD and knowledge capturing system were developed on Windows98 OS. The demo system was written in Visual Basic 6.0. Fig. 5 shows an example of Gear-CAD user interfaces.

4.2. Design experiments

Design experiments were planed as follows. To investigate if our proposed methodology could accurately acquire domain specific evaluation function and individual specific evaluation functions, three designers, who are subsequently called designer-A, designer-B, and designer-C, were participating in the design experiments using Gear-CAD. However, they did not know the existence of others. They were separately assigned the following design task.

- Common requirements and conditions to all designers
  All design components in double-reduction gear system should be determined in detail, i.e. size, material and position.
  Required reduction ratio is 10:1.
  The gear system will be used in outer space. It must be light, small, and cheap to build as well.
  Total weight of the system must be equal to or less than 155.0 kg.
  Total cost of the system must be equal to or less than \$6000.
  Total volume of the system must be equal to or less than 6.0×10^4 cm^3.
  Spur gears that have teeth with a 20° pressure angle should be utilized in this system.
  The input power and speed of rotation should be 10.0 kW and 500 rpm, respectively.
- Individual suggestion to the designers
  To designer-A. Although lighter weight, smaller size and cheaper cost should all be achieved, the most important thing should be the size.
  To designer-B. Although lighter weight, smaller size and cheaper cost should all be achieved, the most important thing should be the cost.
  To designer-C. Although lighter weight, smaller size and cheaper cost should all be achieved, the most important thing should be that the system is stable and well balanced.

4.3. Outcome of design experiments

Three designers, respectively, executed the assigned design task. According to the data monitored by Gear-CAD,

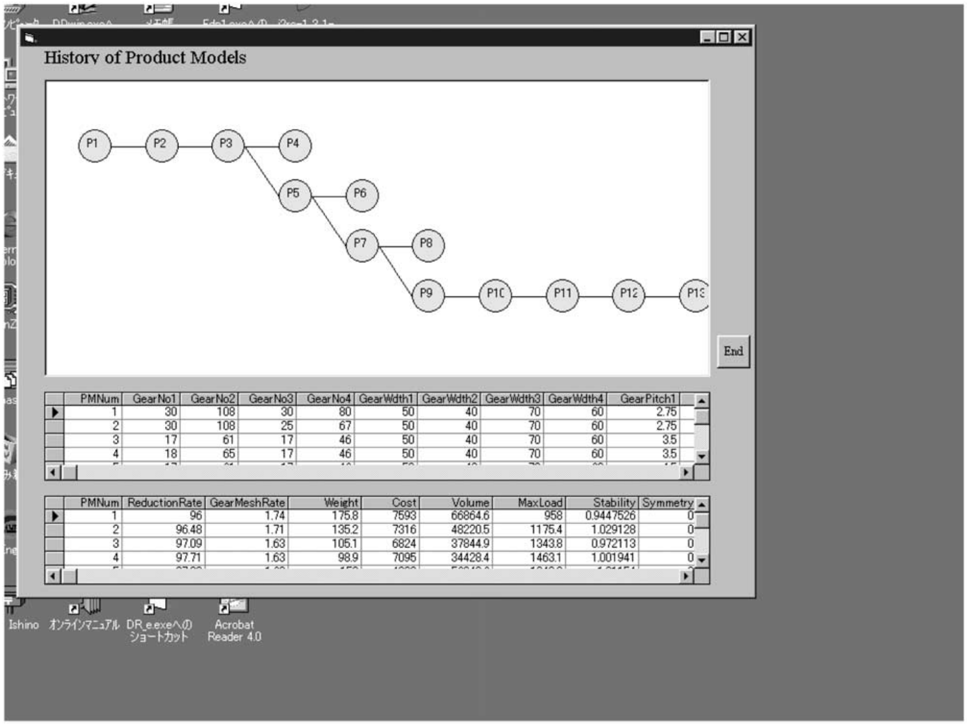

Fig. 6. Design history of designer-A.

the designer-A made 15 product models and finally selected the last one as the solution through design process. In the same manner, the designer-B and the designer-C made 13 product models and eight product models, respectively, and both of them chose the product model they made lastly as the final solution. Figs. 6–8 show their design process recorded Gear-CAD, respectively.

4.4. Results of MEDI

4.4.1. Application of principal component analysis

For all 36 product model data (15 from designer-A, 13 from designer-B and eight from designer-C), principal component analysis was executed to calculate their distance based on the independent components made of design parameters and design attributes. Although this design problem has 40 variables (29 design parameters and 11 design attributes), principal component analysis produced seven synthetic components whose eigenvalue was more than 1.0. To understand the characteristics of product models, we plotted them on a two dimensional space constituted by the first and the second components from principal component analysis. The plotting is shown in Fig. 9.

From Fig. 9, we can see the differences between the models of three designers in the two dimensional, although the meaning of the synthetic components is unknown. Using the coordinates of all product models on seven synthetic components, the distance values between them were calculated, and preference values were obtained from the distance values.

4.4.2. Application of MGP

After designs are all finished, the three designers gathered and decided on their collectively selected performances. They set six performances, namely, main function performance (P_1), size performance (P_2), stability performance (P_3), durability performance (P_4), cost performance (P_5), and maintenance performance (P_6). And then, all 11 design attributes were linked to the performances. P_1 included two design attributes; reduction ratio and gear mesh ratio. P_2 contained two design attributes; reciprocal weight and reciprocal volume. In the same manner, P_3 and P_4 included five design attributes, respectively. P_5 and P_6 contained only one design attribute, respectively.

Then, MGP was conducted based on the classification. MGP was applied for the data of all three designers that contained 36 record sets as to product models. MGP parameters were set as follows:

- Population number per generation: 10,000
- Generation number: more than 50
- Terminal symbols: each design attribute and constant (a condition with all constants, Const; $0 \leq \text{Const} < 1.0$)
- Non-terminal symbols (function symbols): ‘ $+$ ’,‘ $-$ ’, ‘ $\times$ ’,‘ $\div$ ’
- Crossover method: gene f_0 employed uniform crossover, and other genes employed single-point crossover. For all

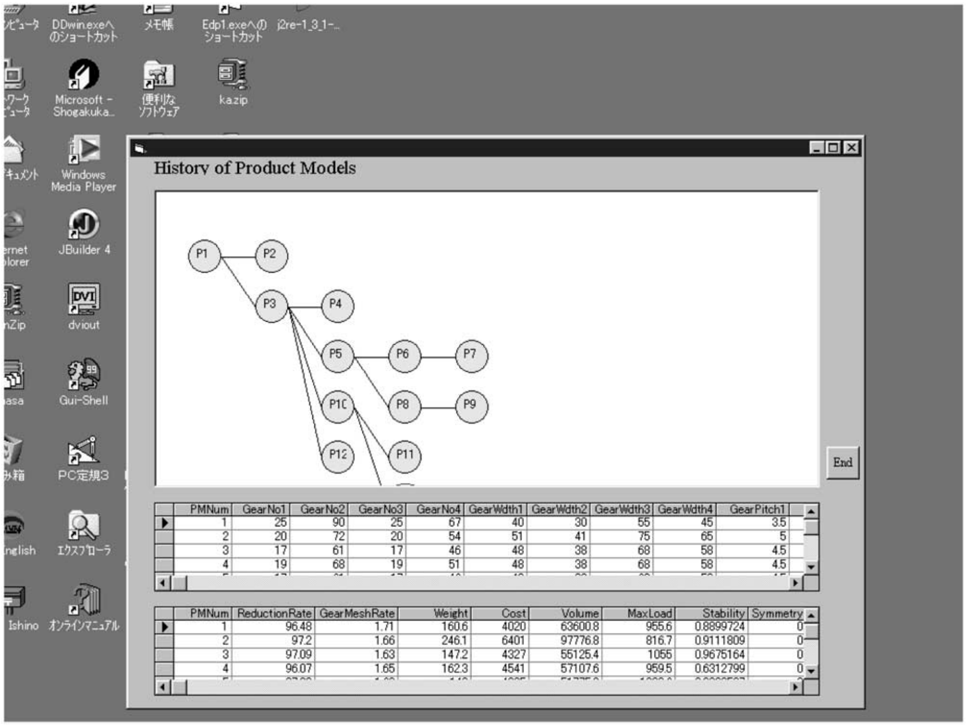

Fig. 7. Design history of designer-B.

genes, roulette wheel selection was applied to choose parents based on the fitness value.

- Mutation ratio: 0.15 for gene f_0, 0.10 for other genes.

4.4.3. Results of MGP and multivariate regression

Three experiments with MGP were conducted in total. The vicissitudes of the best fitness value and the average fitness value in each generation are shown in Figs. 10–12, with every experiment.

In each experiment, the best chromosome was obtained in the 12-generation, the 27-generation, and the 45-generation, respectively. In all experiments, the average fitness value was gradually improving along with the generation, which means the evolutions were well accomplished.

Obtained performance functions f_i and their weights w_i from the best individual (chromosome) are described in Tables 1–3, with every experiment.

In all tables, the raw function indicates the genotype in MGP and the rearranged function is the phenotype translated with a usual notation. Also coefficients were adjusted corresponding to the functions. From the results, three designers discussed and decided that the performance functions in the Experiment 3 were the best suitable for their thoughts about the design task. They seemed to attach greater importance to the understandability and the reasonability of the structure of the performance functions. They stated that it was an understandable finding that they had judged the stability mainly based on only gear balance and symmetry. While doing designing, they had not known how to evaluate the stability performance, although they had checked the performance in a vague manner. And they expressed the reasonability of the size performance. They had learned through their design processes that there had been high correlation between the two design attributes, weight and volume. By viewing the result of MGP, their conviction that one of two design attributes is enough to represent the size performance was confirmed. However, they told that the durability performance was a kind of second-guessing. They thought that although material of gear 1 (MatG1) and material of shaft 3 (MatS3) are good representatives of all gears' and shafts' properties, respectively, they had never subtracted MatS3 from MatG1 in their mind. However, overall, the structure of functions from Experiment 3 satisfied them. The performance functions were determined as follows.

$f_1 =$ Reduction Ratio
$f_2 =$ Reciprocal Volume (= VolumeP in Table 3)
$f_3 =$ Gear Balance + Symmetry
$f_4 =$ Material_of_Gear1 − Material_of_Shaft3
$f_5 =$ Reciprocal Cost (= CostP in Table 3)
$f_6 =$ Material Kinds (= MaintenanceP in Table 3)

Next, multivariate regression was applied for the data that had been renormalized based on the fixed performance

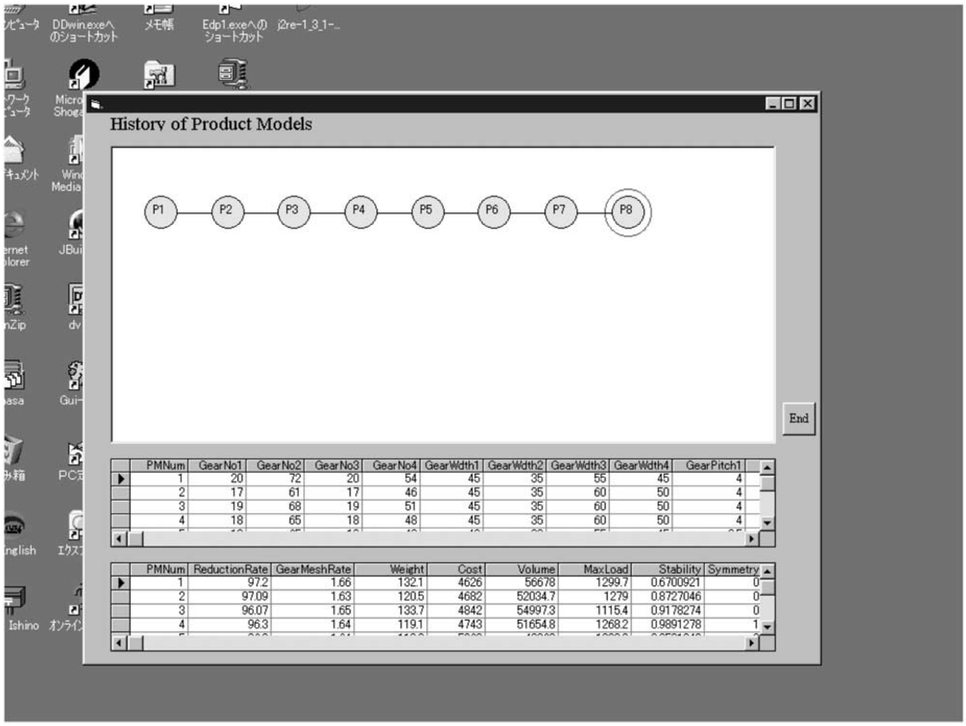

Fig. 8. Design history of designer-C.

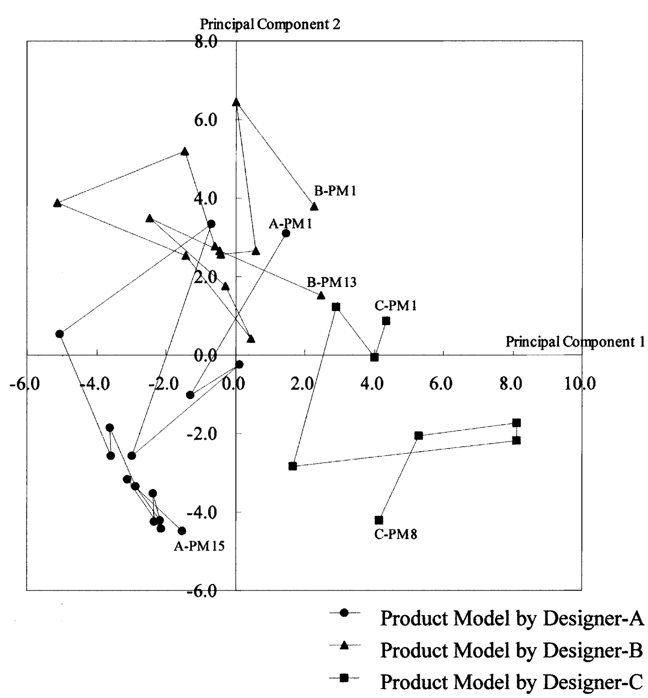

Fig. 9. Product models on the first- and the second-component by principal component analysis.

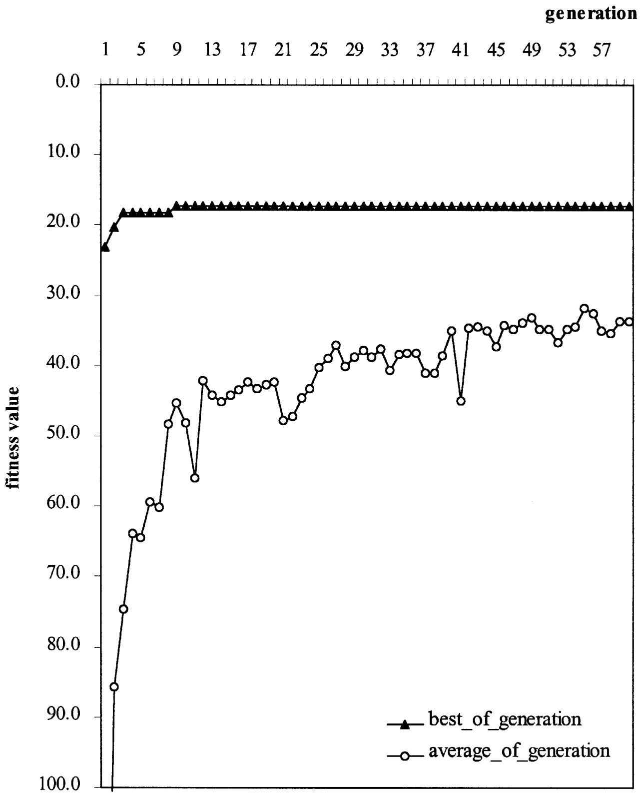

Fig. 10. Vicissitude of fitness value in Experiment 1.

functions, so that actual weights were obtained shown in Table 4.

In Table 4, we use the b for representing the actual weights. The $b_{i\mathrm{all}}$ indicates performance weight in a domain specific evaluation function, and $b_{i\mathrm{A}}$, $b_{i\mathrm{B}}$, and $b_{i\mathrm{C}}$, respectively, indicates the weight in an individual specific evaluation function of each designer. The domain specific evaluation function and individual specific evaluation functions are as follows:

The domain specific evaluation function

$$\text{Preference} = 0.282\times f_1 + 0.177\times f_2 + 0.637\times f_3 + 0.418 \times f_4 + 0.531\times f_5 + 0.101\times f_6$$

The individual specific evaluation function of designer-A

$$\text{Preference} = -0.384\times f_1 + 0.917\times f_2 + 0.528\times f_3 + 0.000\times f_4 + 0.316\times f_5 + 0.000\times f_6$$

The individual specific evaluation function of designer-B

$$\text{Preference} = 0.277\times f_1 - 0.489\times f_2 + 0.590\times f_3 + 1.062 \times f_4 + 0.527\times f_5 + 0.000\times f_6$$

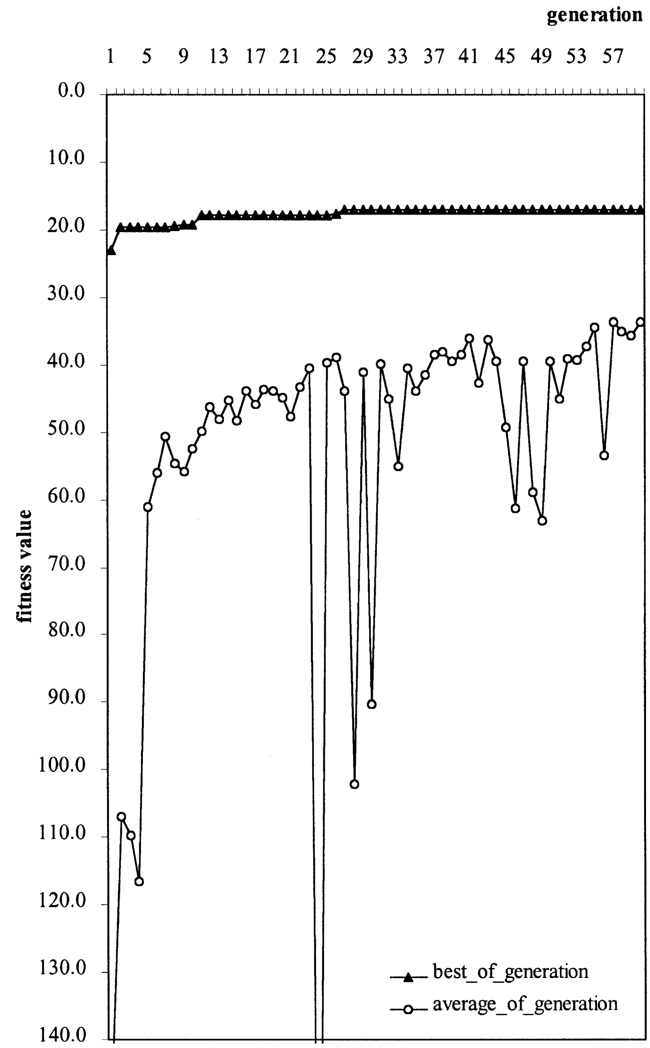

Fig. 11. Vicissitude of fitness value in Experiment 2.

The individual specific evaluation function of designer-C

$$\text{Preference} = 0.314\times f_1 - 0.475\times f_2 + 0.960\times f_3 - 0.478 \times f_4 - 0.985\times f_5 + 0.000\times f_6$$

Compared with performance weights in individual specific evaluation functions, the followings were found. In the case of designer-A, the highest value was put on size performance with the value 0.917. The second highest weight of the designer-A was on the stability performance with 0.528. This indicated that designer-A had paid attention to performances in the order, first size performance (P_2), secondly stability performance (P_3). This outcome matched the suggestion for the designer-A, which had been given to only designer-A. As for the designer-C, the obtained weights

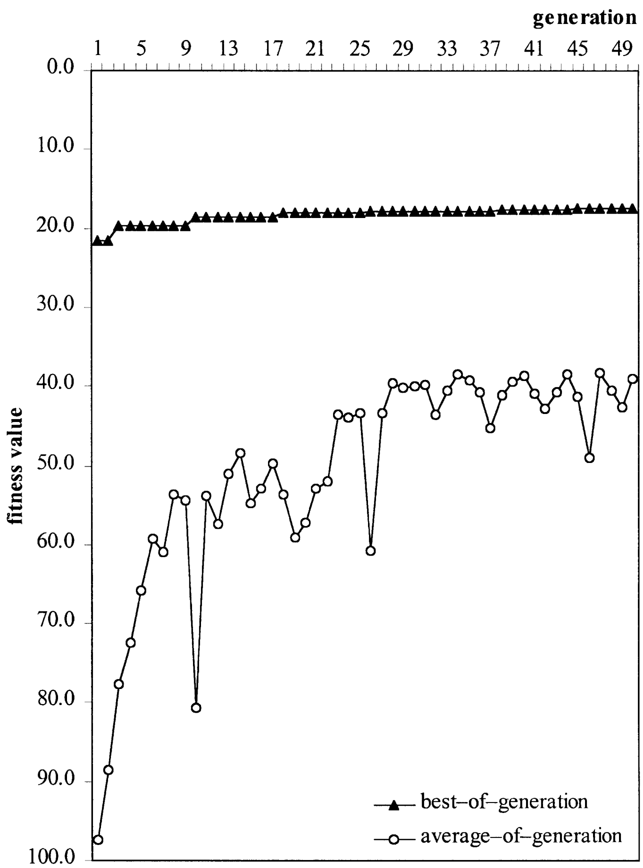

Fig. 12. Vicissitude of fitness value in Experiment 3.

also agreed with the beforehand suggestion; the weight for stability performance (P_3) was the highest with 0.960. In designer-C's case, several weights were negative value, which means that he had not cared these performances as far as the values of design attributes that were included in these performances met the given requirements. In designer-B's case, however, the weight for cost performance (P_5) was the third in spite of the beforehand suggestion. We suppose that it was because he had made some changes on materials during design process. While the kinds of materials were related to cost performance, they also had relationship with durability performance (P_4). And, from the results of domain specific evaluation function, stability (P_3) and cost performance (P_5) had been given attention by the group as a whole.

The earlier results suggest the effectiveness of our methodology in estimating design intent of designers. The differences in beforehand instructions to individual designers could be detected in the magnitude of performance weights in both a domain specific evaluation function and individual specific evaluation functions. Designers can know their individual tendencies in their design by being informed of the results of MEDI. This suggests that MEDI promote knowing expert designers' intent and it should help knowledge transfer in organizations.

Table 1
The best individuals in Experiment 1

Best of fitness value		17.2171
Raw function	Main function performance	f_1 = MUL(0.06243539, ADD(GearMR, GearMR))
	Size performance	f_2 = ADD(0.04034072, WeightP)
	Stability performance	f_3 = MUL(0.8821044, ADD(Gear Balance, ADD(ADD(ADD(Gear Balance, Load Balance), 0.07436836), Symmetry)))
	Durability performance	f_4 = MUL(MatG2,0.5483157)
	Cost performance	f_5 = CostP
	Maintenance performance	f_6 = MaintenanceP
Raw coefficient	Coef1	$w_1 = 0.210$
	Coef2	$w_2 = 0.257$
	Coef3	$w_3 = 0.360$
	Coef4	$w_4 = 0.614$
	Coef5	$w_5 = 0.337$
	Coef6	$w_6 = 0.506$
Rearranged function	Main function performance	f'_1 = GearMR
	Size performance	f'_2 = WeightP
	Stability performance	$f'_3 = 2 \times$ Gear Balance + Load Balance + Symmetry
	Durability performance	f'_4 = MatG2
	Cost performance	f'_5 = CostP
	Maintenance performance	f'_6 = MaintenanceP
Rearranged coefficient	Coef1	$w'_1 = 0.026$
	Coef2	$w'_2 = 0.257$
	Coef3	$w'_3 = 0.032$
	Coef4	$w'_4 = 0.336$
	Coef5	$w'_5 = 0.337$
	Coef6	$w'_6 = 0.506$
Rearranged constant		Const = 0.013

5. Related work

Recently knowledge acquisition using AI techniques in general has attracted attentions from many researchers in the engineering design field. For example, machine-learning techniques are used to predict the replacement of aircraft components [16], and some classification learning techniques are utilized by astronomers to automatically identify stars and galaxies in a large-scale sky survey [3]. It has been recognized, however, that capturing design intent is rather difficult.

Table 2
The best individuals in Experiment 2

Best of fitness value		16.9322
Raw function	Main function performance	$f_1 = \text{MUL(ADD(ADD(ReductR, ReductR), GearMR), 0.01242131)}$
	Size performance	$f_2 = \text{DIV(WeightP, 0.9694437)}$
	Stability performance	$f_3 = \text{SUB(ADD(Gear Balance, Gear Balance), 0.2145615)}$
	Durability performance	$f_4 = \text{SUB(0.5478688, ADD(MatS1, MatS1))}$
	Cost performance	$f_5 = \text{CostP}$
	Maintenance performance	$f_6 = \text{MaintenanceP}$
Raw coefficient	Coef1	$w_1 = 0.160$
	Coef2	$w_2 = 0.268$
	Coef3	$w_3 = 0.216$
	Coef4	$w_4 = 0.209$
	Coef5	$w_5 = 0.166$
	Coef6	$w_6 = 0.469$
Rearranged function	Main function performance	$f'_1 = 2 \times \text{ReductR} + \text{GearMR}$
	Size performance	$f'_2 = \text{WeightP}$
	Stability performance	$f'_3 = \text{Gear Balance}$
	Durability performance	$f'_4 = \text{MatS1}$
	Cost performance	$f'_5 = \text{CostP}$
	Maintenance performance	$f'_6 = \text{MaintenanceP}$
Rearranged coefficient	Coef1	$w'_1 = 0.002$
	Coef2	$w'_2 = 0.277$
	Coef3	$w'_3 = 0.432$
	Coef4	$w'_4 = -0.418$
	Coef5	$w'_5 = 0.166$
	Coef6	$w'_6 = 0.469$
Rearranged constant		Const = 0.068

The reason behind is that estimating design intent seems more domain specific and it is difficult to generalize design intent capturing techniques. Moreover, gathering enough data is difficult in many real engineering design situations. In this paper, we presented our initial attempts to challenge these difficulties.

Research on *design rationale* in the past decade has developed various approaches from different point ofviews [17]. There are three major models: argumentation-based design rationale, action-based design rationale, and model-based design rationale. In the first approach, rationale is represented as a set of arguments (pros and cons) attached to issues, and the issues are interconnected. The issue-based information system (IBIS), developed by Rittel [14], is an example of such methodologies. However, the argumentation-based approach increases designers' burden and interrupts their normal design process. Next, the action-based rationale was developed. The claims are that actions can be explained by themselves [15]. However, they could not well manage the large volume of information recorded. Last, model-based design rationale was proposed. The active design document (ADD) system [5] is based on a certain computational model of design rationale, which is developed for parametric design tasks. Although this system works effective, its model and method are limited to a certain subject. Ganeshan et al. [4] proposed a framework to capture *how* and *why*, in which the core idea is to model design as selection from predefined transformation rules. When a rule is selected, the choice is recorded along with rationale associated with that rule. In their approach, designers' activities are constrained and they are translated into the pre-defined rules beforehand. Myers et al. [18] proposed the framework to capture design rationale from a general CAD data. They use *design metaphor* in order to meaningful activity of a designer, and design rationale is inferred by using *qualitative reasoning*. It is valuable that they aim to develop framework to apply to a general design problem.

Although design rationale models described earlier can yield more specific information behind design, the excessive requirement of interactions with designers limits their effectiveness. On the other hand, our proposed method, MEDI based on a staged design evaluation model, can automatically reason the design intent based on the data automatically gathered through a design process. This method does not interrupt designers' normal design activities. Our original point of view is that design intent can be interpreted as a summation of weighted

Table 3
The Best individuals in Experiment 3

Best of fitness value		17.4536
Raw function	Main function performance	$f_1 = \mathrm{ADD(ADD(0.6219068, ADD(0.2196131, 0.02071297)), ReductR)}$
	Size performance	$f_2 = \mathrm{MUL(0.2333337, ADD(0.2333337, ADD(ADD(VolumeP, ADD(0.2480602, VolumeP)), 0.6926309)))}$
	Stability performance	$f_3 = \mathrm{ADD(Gear\ Balance, Symmetry)}$
	Durability performance	$f_4 = \mathrm{SUB(SUB(MatG1, 0.1749092), MatS3)}$
	Cost performance	$f_5 = \mathrm{CostP}$
	Maintenance performance	$f_6 = \mathrm{MaintenanceP}$
Raw coefficient	Coef1	$w_1 = 0.042$
	Coef2	$w_2 = 0.280$
	Coef3	$w_3 = 0.576$
	Coef4	$w_4 = 0.213$
	Coef5	$w_5 = 0.354$
	Coef6	$w_6 = 0.054$
Rearranged function	Main function performance	$f'_1 = \mathrm{ReductR}$
	Size performance	$f'_2 = \mathrm{VolumeP}$
	Stability performance	$f'_3 = \mathrm{Gear\ Balance} + \mathrm{Symmetry}$
	Durability performance	$f'_4 = \mathrm{MatG1} - \mathrm{MatS3}$
	Cost performance	$f'_5 = \mathrm{CostP}$
	Maintenance performance	$f'_6 = \mathrm{MaintenanceP}$
Rearranged coefficient	Coef1	$w'_1 = 0.042$
	Coef2	$w'_2 = 0.131$
	Coef3	$w'_3 = 0.576$
	Coef4	$w'_4 = 0.213$
	Coef5	$w'_5 = 0.354$
	Coef6	$w'_6 = 0.054$
Rearranged constant		Const = 0.075

performance function. Our proposed framework provides us a starting point toward a general understanding about designers' intent.

6. Conclusions

This paper focuses on estimating design intent, represented as a summation of weighted functions, based on the data monitored through design processes. This estimated design intent provides a basis for us to identify the evaluation tendency of a designer's way to do design. It can be applied to manage design quality by adjusting group members' design intent, and to achieve better design integration and knowledge transfer. In order to achieve our research goal, we (1) introduced a staged design evaluation model as a general yet powerful model to represent decision mechanism in design process, and (2) developed MEDI as a reasoning method. MEDI contains MGP and some multivariate analysis techniques. The characteristics of MEDI are; (i) principal component analysis provides approximate evaluation of how much preferable a specific product model is, assuming the final product model (or design) is the most preferable one; (ii) MGP enables us to simultaneously estimate both structure of target performance functions and the approximate values of their weights for a domain of design problems; and (iii) multivariate regression re-adjusts the approximate weights obtained by MGP into more accurate ones for specific design problems within the domain. Although MGP is not guaranteed to find the optimum solution, it is a novel approach to produce good approximations through a large number of evolutional generate-and-tests. Our framework and method have been tested in a case study of designing a double-reduction gear system. In our experiments, we successfully acquired the target design intent as a domain specific evaluation function and individual specific evaluation functions. A set of performance weights in the domain specific evaluation function means design characteristics as a whole domain, and a set of them in the individual specific evaluation function well reflects individual designers' tendency in their design. Comparing the weights allows us to know the difference among designers. MEDI enables us to know expert or superior designers' design intent, and it can be applied for both design integration and knowledge transfer in organizations.

One limitation of our methodology is that it needs enough amount of design alternatives obtained through design process. Otherwise, it cannot produce stable results. The needed number of design alternatives depends on the

Table 4
Performance functions and weights

All designers	Standard regression coefficient	Coef1 for main function performance	$b_{1\mathrm{all}} = 0.282$
		Coef2 for size performance	$b_{2\mathrm{all}} = 0.177$
		Coef3 for stability performance	$b_{3\mathrm{all}} = 0.637$
		Coef4 for durability performance	$b_{4\mathrm{all}} = 0.418$
		Coef5 for cost performance	$b_{5\mathrm{all}} = 0.531$
		Coef6 for maintenance performance	$b_{6\mathrm{all}} = 0.101$
	Constant		$b_{0\mathrm{all}} = 0.000$
	Coefficient of determination		$R^2 = 0.573$
Designer-A (required being size-conscious)	Standard regression coefficient	Coef1 for main function performance	$b_{1\mathrm{A}} = -0.384$
		Coef2 for size performance	$b_{2\mathrm{A}} = 0.917$
		Coef3 for stability performance	$b_{3\mathrm{A}} = 0.528$
		Coef4 for durability performance	$b_{4\mathrm{A}} = 0.000$
		Coef5 for cost performance	$b_{5\mathrm{A}} = 0.316$
		Coef6 for maintenance performance	$b_{6\mathrm{A}} = 0.000$
	Constant		$b_{0\mathrm{A}} = 0.000$
	Coefficient of determination		$R^2 = 0.852$
Designer-B (required being cost-conscious)	Standard regression coefficient	Coef1 for main function performance	$b_{1\mathrm{B}} = 0.277$
		Coef2 for size performance	$b_{2\mathrm{B}} = -0.489$
		Coef3 for stability performance	$b_{3\mathrm{B}} = 0.590$
		Coef4 for durability performance	$b_{4\mathrm{B}} = 1.062$
		Coef5 for cost performance	$b_{5\mathrm{B}} = 0.527$
		Coef6 for maintenance performance	$b_{6\mathrm{B}} = 0.000$
	Constant		$b_{0\mathrm{B}} = 0.000$
	Coefficient of determination		$R^2 = 0.690$
Designer-C (required seeking a stable and well-balanced system)	Standard regression coefficient	Coef1 for main function performance	$b_{1\mathrm{C}} = 0.314$
		Coef2 for size performance	$b_{2\mathrm{C}} = -0.475$
		Coef3 for stability performance	$b_{3\mathrm{C}} = 0.960$
		Coef4 for durability performance	$b_{4\mathrm{C}} = -0.478$
		Coef5 for cost performance	$b_{5\mathrm{C}} = -0.985$
		Coef6 for maintenance performance	$b_{6\mathrm{C}} = 0.000$
	Constant		$b_{0\mathrm{C}} = 0.000$
	Coefficient of determination		$R^2 = 0.951$

complexity of design problem, and is related to the number of design parameters and design attributes.

As a next step, we plan to investigate, through more case studies, the issues related to computational cost and sensitivity of our proposed model and method.

References

[1] Astrom KJ, Eykhoff P. System identification, a survey. Automatica 1971;7.

[2] Berry MJA, Linoff G. Data mining techniques: for marketing, sales, and customer support. New York: Wiley, 1997.

[3] Fayyad UM, Djorgovski SG, Weir N. Automating the analysis and cataloging of sky surveys, in advances in knowledge discovery and data mining. California: AAAI Press/The MIT Press, 1996. p. 471–93.

[4] Ganeshan R, Garrett J, Finger S. A framework for representing design intent. Des Studies 1994;15(1):59–84.

[5] Garcia ACB, Howard HC. Acquiring design knowledge through design decision justification. Artificial Intell Engng Des, Anal Manufact 1992;6(1):59–71.

[6] Goldberg DE. Genetic algorithms in search, optimization, and machine learning, building-block hypothesis. Reading, MA: Addison Wesley, 1989.

[7] Grefenstette JJ. Credit assignment in rule discovery systems based on genetic algorithms. Machine Learning 1988;3:225–45.

[8] Hazelrigg GA. Systems engineering: an approach to information-based design. Upper Saddle River, NJ: Prentice-Hall, 1996.

[9] Hondo N, Iba H, Kakazu Y. COAST: an approach to robustness and reusability in genetic programming. Proceedings of the Third International Conference on Evolutionary Computation, IEEE Press, 1996.

[10] Iba H, deGaris H, Sato T. Genetic programming using a minimum description length principle. In: Kinnear Jr KE, editor. Advances in genetic programming. Cambridge, MA: MIT Press, 1994. p. 265–84.

[11] Keeney RL, Raiffa H. Decisions with multiple objectives: preferences and value tradeoffs. New York: Wiley, 1976.

[12] Koza J. Genetic programming, on the programming of computers by means of natural selection. Cambridge, MA: MIT Press, 1992.

[13] Koza J. Genetic programming II: automatic discovery of reusable programs. Cambridge, MA: MIT Press, 1994.

[14] Kunz W, Rittel W. Issues as elements of information systems. Working paper 131, Center for Planning and Development Research, University of California, Berkeley, 1970.

[15] Lakin F, Wambaugh J, Leifer L, Cannon D, Sivarel C. The electronic design notebook: performing medium and processing medium. Visual Comput: Int J Comput Graphics 1989;5:214–26.

[16] Letourneau S, Famili F, Matwin S. Data mining to predict aircraft component replacement, in data mining: a long-term dream. IEEE Intell Syst 1999;November/December:59–66.

[17] Moran TP, Carroll JM. Design rationale: concepts, techniques, and use. Mahwah, NJ: Lawrence Erlbaum Associates, 1996.

[18] Myers KL, Zumel NB, Garcia P. Automated capture of rationale for the detailed design process. Proceedings of the Eleventh Conference on Innovative Applications of Artificial Intelligence. (IAAI' 99), 1999.

[19] Rencher AC. Methods of multivariate analysis. New York: Wiley-Interscience, 1995.

[20] von Neumann J, Morgenstern O. Theory of games and economic behavior. 2nd ed. Princeton, NJ: Princeton University Press, 1947.