COMPUTER INTELLIGENCE IN INTEGRATED SATELLITE DESIGN SUPPORT INFRASTRUCTURE

Shinichi Nakasuka and Koichiro Maeda
Department of Aeronautics and Astronautics, University of Tokyo,
Hongo 7, Bunkyo-ku, Tokyo, 113-8656 , JAPAN
TEL:+81-3-5841-6590, FAX:+81-3-5841-8560, nakasuka@space.t.u-tokyo.ac.jp

Masao Sato, Kotaro Kiritani, Kenichiro Sato and Toshio Koda
Kamakura Works, Mitsubishi Electric Corporation
Kamimachiya 325, Kamakura, 247-8520, JAPAN
TEL: +81-467-41-6155, FAX: +81-467-41-6951

ABSTRACT

An intelligent computer system to support satellite design is proposed. To support human designer in time consuming trial-and-error design process, the system has the capability 1) to integrate various local design support tools to obtain one feasible solution as quickly as possible, 2) to give intelligent advise to the designers on how to modify the current design in order to improve the design solution in a certain direction, and 3) to enable human designers to customize and implement a certain design sequence so that the computer can perform parts of design process autonomously. Gradient search technique and machine learning-based production system is employed for the second function, and a concept of design process editor is proposed for the third function. The way to implement this computer intelligence into the system is discussed. Basic concept of the system architecture as well as a prototype model is described and discussed.

1. INTRODUCTION

Satellite design is a highly complicated and time consuming task because many different areas must be considered concurrently to reach a consistent as well as satisfactory design solution. In AI terms, this process can be interpreted as searching for a set of many design parameters which yield a design solution satisfying various given requirements under various constraints. In satellite design case, the number of design parameters, and consequently, the search space is huge. Moreover, the search process becomes even more difficult because of the complicated interactions between design parameters. For example, a modification of a certain parameter affects, in most cases, not only the quality of the design solution but also the characteristics of the design problem, such as the sensitivity of the design quality to the change of other parameters. As a result, the search process should become highly trial-and-error fashion, including many backtracking and iterations.

In the current satellite design, this search is made in most cases by human designers. Computer support is partly incorporated, but is limited to the local analysis of the effects of design parameters within each design area, such as within thermal design, communication link design or control system design. The most difficult task of trading-off between the requirements given from the different areas or obtaining a consistent design solution is dependent on human designers, which results in the long time required even to obtain one feasible solution. From the needs' side, however, it is essential to obtain one feasible design solution quickly especially in the conceptual design phase, and it is highly desirable if the computer system could advise how to change certain parameters to improve the design quality. These capabilities have not been provided in the current computer aided satellite design system.

In order to respond to these needs, we have been developing an integrated computer support infrastructure for satellite design. The key objectives of this system are as follows:

1) Integration of the various local design support tools in order to obtain one feasible solution as quickly as possible.

2) Intelligent support by the computer to advise the designers how to modify the current design in order to improve the design solution in a certain way.

3) Enabling human designer to customize and implement a certain design sequence so that the computer can perform parts of design process autonomously.

In section 2, the concept as to how the computer supports human in design process is proposed, and section 3 gives the detail of the system architecture. Modification of design parameters plays the key roles in the proposed design framework, and computer intelligence is highly required to support this function, which is discussed in section 4. Current status of actual implementation is briefly given in section 5, and conclusions and future works are summarized in section

Proc. Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1-3 June 1999 (ESA SP-440)

6.

2. BASIC CONCEPT OF COMPUTER SUPPORT IN DESIGN PROCESS

2.1 Difficulties of Design Problem

Design problem can be interpreted as searching for a set of usually many design parameters which yield a design solution satisfying given requirements under given constraints. In case of satellite design, the difficulty comes from not only its large number of design parameters and evaluation items but also from the mutual interactions between them. The whole design problem can seemingly be separated into several design sub-problems such as orbital design, structural design, thermal design, control system design, etc, but actually these sub-problems have strong mutual interactions, and one good solution of a certain design problem often is a bad solution for another problem. At that time, human designers often negotiate with designers of the conflicting design fields, trading-off the requirements given to them, finally to reach the acceptable, not optimal, solution. This process is quite time consuming and even stressful, and therefore we need an intelligent support from computer not only in easy analysis or drawing of the designed subsystems but also in trading-off the various requirements in the different design fields.

This kind of "Divide-and-govern" approach has been studied extensively especially in the field of distributed problem solving or multi-agent systems (MAS). For example, Yoshida et al. proposed a new MAS architecture in which each agent performing different field of design communicates with the other agents by way of evaluating the design results of the other agents [1]. By learning appropriate balance between the desire of each agent to pursue their own benefits and the desire to cooperatively achieve the goal, the total system gradually acquires expertise to reach acceptable solutions more and more efficiently. Many researches have been performed in this line, but the common limitation of this approach is the difficulty to specify the way or expertise to mitigate conflicts between agents in the different design fields.

2.2 Sequential Operator Application Approach

Another approach to this design problem is "sequential operator application" framework such as in Fig.1. Starting from the initial design (top node), "design operators" are applied sequentially until the goal condition (i.e., sufficiently good design) is achieved. Here, "design operators" mean such operators as to design a certain part which has not been designed yet, to change the current design in a certain way, or sometimes to modify the given requirements and constraints. The initial design may be a design of the previous satellite with similar missions, or sometimes nothing. The search strategy may be depth-first type, breadth-first type or best-first type. This formulation is quite straightforward and suited for applying various AI techniques, but its difficulty resides in the combinatorial explosion of the search space because there are usually many applicable operators at each node.

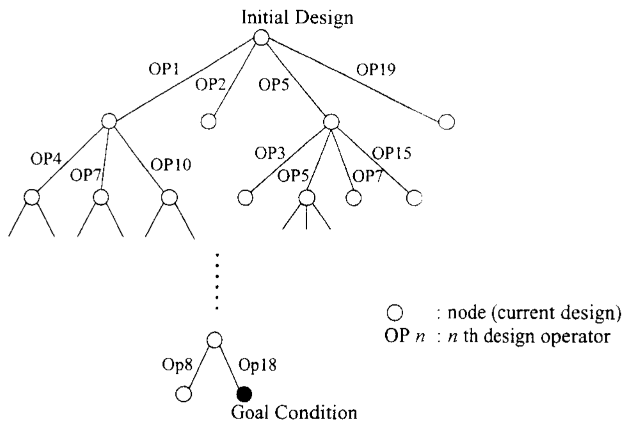

FIG.1 Sequential Operator Application Framework for Design Problem

One way to solve this problem is that the system is implemented with knowledge to specify what type of operator should be applied at what situation. Nakasuka et al. suggest in [2] and [3] that machine learning can play the essential role in obtaining such knowledge, in case that human designer cannot provide enough knowledge for this objective. In [2] and [3], knowledge as to the relationships between the attribute values describing the current design status and the desirable design operators are acquired by machine learning in the course of initially trial-and-error style problem solving process. This methodology has been found quite effective and applicable to various problems, and successful applications have been made to design of control system [2] and scheduling problems [3][4]. The architecture proposed in this paper also employs this machine learning based methodology in principle. The detail will be described in section 4.

2.3 Design Process Editor

Another important knowledge often very useful for design efficiency is "sequence of design." For example, we had better define the equipment list before designing the size of solar panel, and we frequently had better design satellite orbit before designing communication system (Of course, there is some problem where the reverse is true.) This appropriate sequence of design sometimes comes naturally from the causal relationships of parameters (such as the former example above) or sometimes comes from more complicated efficiency reasons (such as the latter example). The desirable design sequence of the former type can easily be deduced from the causal network of design parameters. The latter type, on the other hand, is itself a certain expertise and cannot easily be specified. Computer should support human designers also in these respects. In [5], Obata et al. proposed a satellite design support system which can provide the capability of generating several "design processes," which specify the sequence of designing multiple parameters, from which users can select the most appropriate one according to their intention. Obata uses model based method which does not require any experiential knowledge to generate the candidates of design process.

In this paper, this idea is further enhanced to propose an idea of "design process edit," which enables users to "design the design process" freely, in addition to the above capability of model based generation of the sequence of designing parameters based on the causal relationships of parameters. Figure 2 and 3 give one example of this capability; Fig. 2 is a causal relationships of design parameters and Fig.3 shows a generated design process partly based on Fig.2 and partly on the user edition.

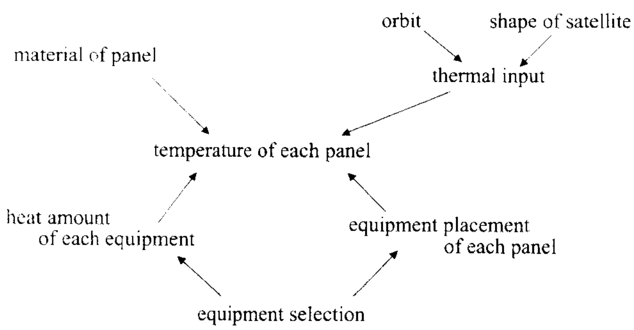

FIG.2 Example of Causal Relationships of Parameters

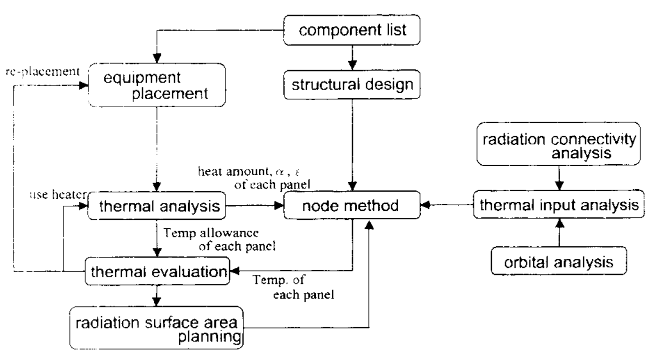

FIG.3 Example of Generated Design Process

It should be noted that not only a sequential parameter setting but also an iteration loop can be incorporated, if needed. Such loops are needed where parameters are dependent on each other. The design process can also reflect the priority between different requirements.

The design process is in a sense a "script" in AI terminology; it describes how the parameters are sequentially got set or tuned. Once this design process is completed, the computer automatically chooses how to set a certain parameter from among the options of simply calculating it from the already set parameters, doing some iterations until converged, or consulting human designers for the value. The system should be made so that once a certain parameter has been modified, then all the parameters affected by this change are re-calculated automatically. With these capabilities, the system can quickly generate one solution while maintaining the consistency of the parameter values, even though of course the solution does not always satisfy all the requirements and constraints.

3. PROTOTYPE SYSTEM ARCHITECTURE

A prototype system is now being developed based on this concept. The system consists of several modules each of which corresponds to a local design tool such as link design tool or solar paddle design tool, and is coded in object oriented fashion (Fig.4). Each module has input parameters and output parameters and knows which modules can possibly change the values of its input parameters. Some modules have a user interface to enable human designer to modify some parameters directly and others have just the function of calculating the output parameter (design results) from the input parameters (assumptions.) In the design phase, when a certain module is triggered to change some parameter values, the downstream modules (modules which are affected by this change) are also triggered automatically to maintain consistency of the parameter values.

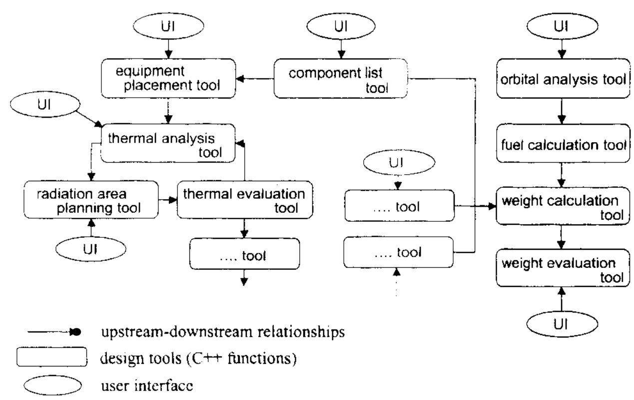

FIG.4 Overall Architecture of Prototype System

Some iterative design modules are also implemented in the same manner, which performs a certain design process requiring an iterative calculation which does not require any expertise such as which parameter to change, etc., until a certain local constraint(s) is satisfied. Examples of such modules already implemented in the prototype system are as follows:

- Tuning of RF power to satisfy link margin requirement and PFD constraint
- Tuning of altitude to satisfy link margin requirement and PFD constraint
- Tuning of antenna gain to satisfy link margin requirement and PFD constraint
- Tuning of area of surface radiation and heater electricity power to satisfy temperature requirement
- Tuning of required fuel to keep the desired altitude
- etc.

As you can note from these examples, there are several ways to modify parameters to satisfy a certain currently not-satisfied requirement (such as the first three in the above example for link margin and PFD requirements.) These modules do not provide support in terms of which modification way is the best in the current situation, but provide just an automatic calculation function which requires, if a human designer does it, lots of time because of iterative nature. A human designer, therefore, must specify which to use to modify the parameters.

The key characteristics of this architecture are that :

(a) each module is coded in declarative form and the sequence of its activation is not hard coded, which enables human designers to pursue their own design process and to modify any design parameters at any time of the design phase,
(b) the consistency of the design parameters is maintained at any moment, which enables designers to change parameters without paying much attention to the effects of this change,
(c) local experiential design rules can easily be implemented in the form of design modules which can also be triggered at any moment, and
(d) the system does not change parameter values by itself unless the changes are necessary to keep consistency or a certain iterative design module is triggered by a human designer.

With this capability, the system can quickly obtain one design solution. However, it is not guaranteed that the given requirements and constraints are satisfied in the obtained solution. The next question is how the system can support the efficient modification of parameters to more satisfy the requirements and constraints, and computer intelligence is, of course, required in this respect. We will discuss it in the next section.

4. INTELLIGENT SUPPORT FOR PARAMETER MODIFICATIONS

4.1 Sum of penalty functions

In order to systemize the satellite design, it would be required to provide some way to evaluate the overall goodness of the design in an objective way. In AI field, "sum of penalty functions each of which represents how each requirement or constraint is satisfied" is frequently used for this objective. We would like to follow this trend. In this method, hard constrains can be represented by very sharp valley of penalty function as in the right figure of Fig.5, while standard requirement will be like the left figure. The merit of using this strategy is that the intention of the human designer can very easily be represented, such as which requirement has higher priority or how strict the constraint is, etc.

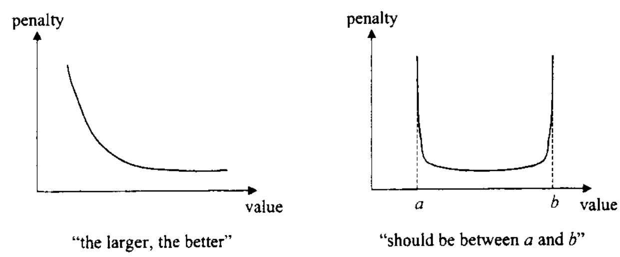

FIG.5 Penalty Functions to Represent Requirements (left) and Constraints (right)

Using the term of "Sum of Penalty Functions" which is referred to as "SPF" hereafter, the ideal design process can be stated as "the search for the global minimum of SPF" This is, of course, the ideal design, and usually we cannot obtain the global minimum, but just local minimums or a certain point whose SPF is below a certain threshold. In our concept of computer intelligence to help design, computer should provide some guidance to lead the human designer to the global optimum or the design point whose SPF is as near the global minimum as possible. We identify two mechanisms needed for this objective.

4.2 Local modification based on gradient search

Locally, we can tell, based on the sensitivity of SPF to each parameter, modifications of which parameter to which direction will reduce the SPF. This type gradient descent algorithm is rather easy to implement, such as in the following algorithm:

(i) find the penalty function which has the worst value in the current design
(ii) find the design parameters which have large effects

on the selected penalty function
(iii) search for the changes of these parameters which result in the largest descent of the SPF
(iv) if any changes will not reduce SPF any more, then quit. Otherwise go back to (i).

The most important expertise in this algorithm is (ii), that is, the knowledge as to which parameters should be modified in order to improve a certain penalty function. In the prototype system, this expertise is obtained from the human expert designers in the satellite design area. We are now studying application of machine learning technology in order to more precisely indicate which parameters to modify.

4.3 Global modification based on design knowledge

When the above algorithm stops, the design solution is at "a local minimum point." If SPF of this point is below the threshold, then the design is completed, i.e., we have obtained one feasible solution which satisfy the requirements and constraints. But, if not, the local search alone cannot lead the design to the good direction any more. Or, if we want more than one design alternatives which all satisfy the requirements and constraints to some extent, this local search will not be of any help. In these cases, we need more global or "jump type" modifications such as in Fig.6.

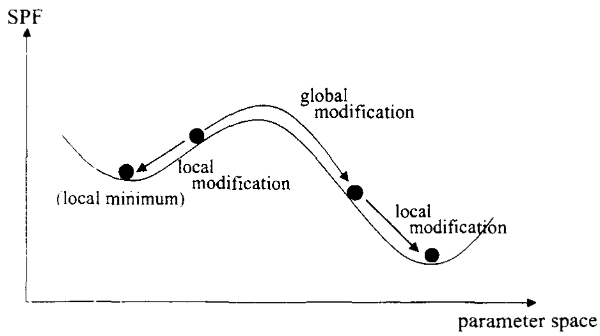

FIG.6 Local and Global Parameter Modifications

The simple and effective way to realize this function is to prepare several modification rules in the form of IF-THEN production rule. IF part represents the current situation of the design, which should include:

- which penalty function(s) has bad value now
- which parameters still have room for increase (or decrease)
- relative hardness of requirements/constraints (such as "weight constraint is very hard in this design problem as compared with electricity constraint", etc.)
- etc.

THEN part shows some modification operation such as:

- modify parameter A to reduce a certain penalty
- modify parameters A and B so that a certain penalty function(s) can be as small as possible
- modify parameter A until a certain penalty function becomes less than a certain value and then modify parameter B until this penalty function become sufficiently small
- increase (or decrease) a certain parameter by a certain amount
- etc.

The attributes in IF part and the modification strategies in THEN part as well as some combinations of IF and THEN parts can be specified beforehand using expertise in satellite design. Examples of such knowledge are as follows.

- IF the total weight is too large, THEN change the equipment to be used.
- IF the total weight is too large, THEN replace the current equipment with one requiring less power.
- IF the total weight is too large, THEN change the initial altitude.
- IF the required heater power is too large, THEN change the position of the equipment requiring largest power.
- etc.

However, as you can easily imagine, this a priori knowledge alone is not sufficient to effectively lead the design process to the global optimum point. In order to deal with this "knowledge bottleneck" problem, we have been applying machine learning technology in the same way as employed in [2] -[4]. In these papers, the relationships between the attributes describing the current status and the effective operators are learned using training data obtained during the exhaustive search phase. Neural network [2], state-space representation [3], or decision tree [4] is utilized as a learning schema. In our prototype system, neural network is utilized because of its flexibility in representing the relationships between IF and THEN parts.

5. CURRENT STATUS OF ACTUAL IMPLEMENTATION

University of Tokyo and MELCO are now cooperatively implementing the actual intelligent design support system based on this concept. This is the first step activity in MELCO towards the ultimate goal of building an integrated satellite design infrastructure including mission design, early conceptual design, detailed design and analysis [7]. The target will include wide variety of satellites from small to large as well as

LEO to GEO. The key requirements for the system are that:

1) it can quickly generate many alternative designs, even if it is not optimum solutions, assuming its usage in the early conceptual design phase,
2) it can analyze the current design and make suggestions as to the inconsistency and parameter modifications, and
3) it can incorporate various in-house design routines already coded or will be coded by the satellite design experts

The system consists of C++ functions of autonomous design routines, design process editors and evaluation modules as well as spread sheet type human interfaces.

The current status of development is that the first version of the system which uses only human expertise for IF-THEN rules of parameter modifications are completed, and the machine learning capability is now being implemented. The "design process edit" function is now implemented in the form of modification of code level, not on the graphical user interface.

6. CONCLUSIONS AND FUTURE WORKS

Integrated design support infrastructure and computer intelligence can be said essential to accelerate and improve the otherwise quite complicated and time consuming satellite design process, especially in the near future when more and more varied satellites should be designed in much shorter time. The operations in the design process for which computer intelligent support is the most indispensable are deciding the sequence of subsystem designs and the parameter modification strategy when the current design should be modified. The proposed concept provides one approach to how these kinds of intelligence is obtained and in what way it is implemented.

Much work should be done towards the really useful design support system: in the research level, the incorporation of machine learning is the main research item. Especially how to represent the attributes to describe the current status and how to obtain sufficient training data for learning will be the main issues. In technical level, graphical user interface for design process editor as well as IF-THEN rule editors should be implemented. We are continuing research in these directions.

REFERENCES

[1] T.Yoshida, K.Hori and S.Nakasuka, A Cooperation Method via Metaphor of Explanation, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E81-A, No.4, (1998)

[2] S.Nakasuka and T.Ninomiya, Applications of Dynamic Scheduling Technique to Space Related Problems - Some Case Studies, Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp.423-426, (1994)

[3] S.Nakasuka, Machine Intelligence in Control System Design - Its Possibility and Methodology, Transactions of the Japan Society for Aeronautical and Space Sciences, Vol.32, No.98, pp.206-216, (1990)

[4] S.Nakasuka and T.Yoshida, Dynamic Scheduling System Utilizing Machine Learning as a Knowledge Acquisition Tool, International Journal of Production Research, Vol.30, No.2, pp.411-431, (1992)

[5] T.Obata, T.Watanabe, R.Tachibana, K.Hori and S.Nakasuka, Concept of Intelligent Design Support System for Satellites, Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp.343-346, (1997)

[6] D.Goldin, S.Venneri and A.Noor, New Frontiers in Design Synthesis, IAF-98-U.2.01, 49th IAF Congress, (1998)

[7] K.Ijichi, T.Fuji, T.Anzai, M.Sato, H.Hasegawa and T.Yamaguchi, The Advanced Small Satellite Bus System with Integrated Spacecraft Controller, IAF-98-U.1.05, 49th IAF Congress, (1998)