

Comparative Assembly Planning During Assembly Design*

Gerard Jounghyun Kim
Rm. A-127, Bldg. 220
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA
e-mail gkim@cme.nist.gov

Sukhan Lee† and George A. Bekey
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0782, USA

Abstract

Recent strides toward concurrent engineering have called for a need for integrating design with assembly planning. That is, a true concurrent engineering platform must be able to, for example, perform preliminary assembly planning during conceptual design stages so that alternative assembly plans can be evaluated and compared for redesign or refining a design with a promising candidate assembly plan. This paper presents such an integrated system of an assembly planner and a (re)designer, in which, results from comparative analysis of preliminary candidate assembly plans are used to (re)design further details of a given assembly. The integration results in a more effective way of doing assembly planning and "Design-for-Assembly". A redesign process of a simple switch box is demonstrated to illustrate the benefits.

1 Introduction

Generally, the problem of assembly planning concerns determining an order (linear or partial) of assembling a product. Most usually, assembly planning is performed upon a final design, and based on the results, a need for redesign may be revealed. This is a rather inefficient and ineffective redesign process which may require repetitive and lengthy design revisions, starting from early stages of design.

The concept of assembly planning as a separate process from design is being challenged as the practice of concurrent engineering is becoming more prevalent in

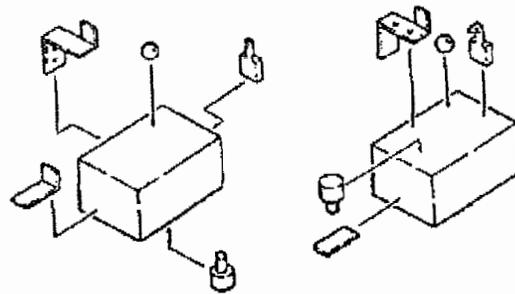


Figure 1: Designing for fewer assembly directions. Reprinted with permission of the Society of Manufacturing Engineers (SME). Copyright 1983 from [1].

the American industry. A true concurrent engineering system should, if possible, allow prediction and resolution of design problems with a preliminary assembly plan before the design is finalized. Conversely, an assembly plan may change due to new constraints given by a revised design.

Closely related to the issue of achieving concurrent engineering in this context are the principles of "Design-for-Assembly (DFA)". DFA is a set of design guidelines for improving product designs for easy and low cost assembly [2]. Principles of DFA have surfaced as one of the important criteria and heuristics for determining a cost-effective assembly plan.

Figure 1 illustrates an example of (re)designing a subassembly for fewer assembly directions. Note that this particular (re)design is dependent on a choice of a particular base part and subassembly grouping (i.e. an assembly plan).

This paper presents an integrated system of an assembly planner and a DFA redesigner, called INSPIRE-2 (INtelligent ASsembly Planning Integrated with REdesign), in a continued effort to the work reported in [13]. It is more than a mere software inte-

*This research was conducted at the National Institute of Standards and Technology under the National Research Council Postdoctoral Research Associateship. A more complete version of this paper with large sized figures are available on WWW at <http://cwis.nsc.edu/dept/robotics/personal/gkim/home.html>.

†Also with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA.

gration, since the activities of assembly planning and (re)design are interleaved during the design process under a single framework.

INSPIRE-2 is based on two previous projects in assembly planning and DFA redesign by the authors: an assembly planner called "Backward Assembly Planner (BAP)" [12], and a DFA redesigner called "REV-ENGE" [10]. BAP operates based on a recursive decomposition of an assembly into feasible subassemblies, and analyzes assembly cost in terms of stability, directionality, and manipulability (which are among the most important DFA criteria) to guide the generation of preferred assembly plans.

The redesigner, called REV-ENGE (from REVERSE ENgineering), interactively produces a redesign strategy by, first, generating a *default* design process of an assembly. A default design process is a probable sequence of design decisions that might have led to the creation of the original design. The default design process is replayed from its beginning as if the product is being designed from scratch. Design decisions are analyzed with respect to their effect to ease of assembly, and if a certain design decision is found problematic, a case-based approach is taken to modify the appropriate portion of the design process. Intermediate and final outputs are redesign suggestions in a form of symbolic description of a newly revised design process (to be geometrically realized manually).

The main objective of INSPIRE-2 is to allow simultaneous modeling of "assembly-conscious" design and associated processes. The advantages of the overall framework can be summarized as: (1) reduction in cycle time between design and assembly planning, (2) change of constraints (both design or operational) in a direction of selecting a more cost effective assembly plan, (3) a more complete and global DFA analysis by identifying design problems dependent on a particular assembly sequence, and (4) a more realistic cost analysis through incorporation of operational rationale and explicit design stages for assembly operation selection. In order to integrate assembly planning (BAP) into the (re)design process (REV-ENGE), new design stages and design decision vocabularies have been added to REV-ENGE framework to explicitly represent assembly planning, assembly design, and assembly process selection. To evaluate and compare alternative assembly sequences and alternative redesigns, a comparative analysis module has been added to the system, which is the focus of this paper.

2 Related Work: Concurrent Assembly Design

Much work has sprung up in the field of concurrent engineering [15] [3] [4] and assembly modeling [16] [11] [18] [6]. However, very little focus has been paid to the issue of concurrent assembly design. Most concurrent engineering systems focus on individual part design and are configured as a loose coupling (e.g. by networking) of engineering softwares (e.g. FEM, product database, CAD, process planner), where it is up to the user to access and gather relevant information for his or her particular engineering purpose.

Many assembly analysis methods have been devised in connection with assembly planning systems with design feedback capabilities [5] [1] [7] [17], although it is unclear how the analysis results could be integrated with assembly design systems. Hsu, et. al. developed a feedback system that used three objective criteria for evaluating an assembly sequence with regards to DFA to identify parts that need redesign, and to propose a redesign [8]. In [20], a method for incrementally analyzing an assembly during design with respect to tolerance is presented in the context of a larger concurrent engineering system called Next-Cut [4].

3 Overview of INSPIRE-2

Figure 2 shows the overall architecture of INSPIRE-2. Given input descriptions of an existing design and its assembly operations, a *default* design process is generated stage by stage, starting from conceptual design. The default design process is replayed from its beginning as if the product is being designed from scratch. For each design stage, a *Local* analysis is performed to find DFA problems that are independent of an assembly sequence. A few promising assembly plan candidates may be chosen based on a *Global* analysis. The *Global* analysis evaluates each candidate assembly sequences with respect to subassembly stability, directionality, and parallelism. A comparative analysis among candidate plans may be performed in an attempt to optimize the assembly cost with respect to certain analysis parameters. Findings from the analysis are mapped to the appropriate portion of the design process. The design process is replayed and whenever a problematic design decision is encountered, it is modified using a case-based approach.

When all the design problems in a given design stage are addressed, the design process for the next design stage is generated and the same process continues. When the system reaches the final design stage-

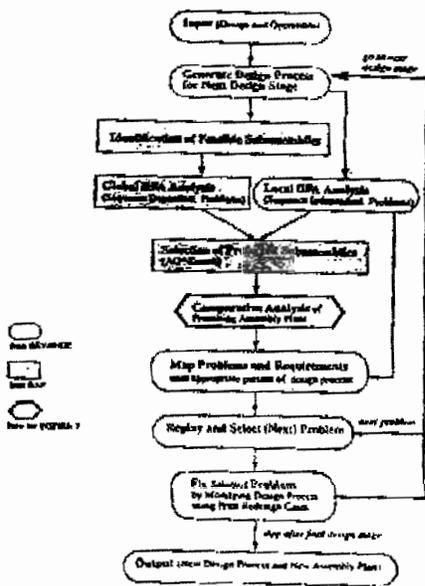


Figure 2: System architecture of INSPIRE-2. Modules from BAP and REV-ENGE are enclosed in rectangles and ovals respectively.

an output in the form of a new design process and a new assembly plan is produced. For a better understanding of the details of BAP and REV-ENGE, please refer to [12] and [10].

4 Interleaving Assembly Planning and (Re)Design

4.1 Design Representation

A *Design Object* refers to any object whose existence or form is to be determined by a designer and includes *Artifact and Operation*. An *Artifact* is a collection of design objects that represent the assembly. Note that an assembly operation is regarded as a design object as well. Each design object is associated with design decisions, which create and set various attribute values of the design object, and define its relations to others. Each design decision may be supported by number of design rationales, and marked with design problems (after analysis). Design problems may map to redesign cases that might be able to solve them. An example assembly of a switch box is shown in Figure 3.

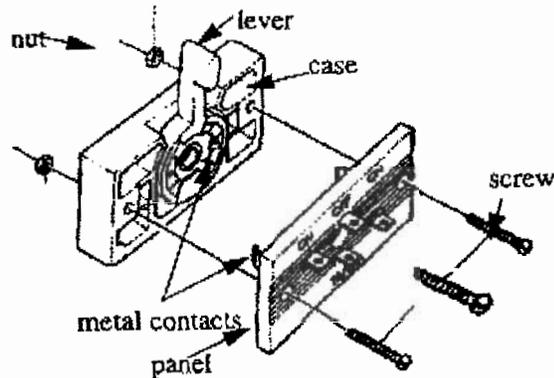


Figure 3: An exploded view of the switch box assembly (Courtesy of Tamiya Plastic Model Co). The switch box assembly consists of a case, panel, lever, six metal contacts, 2 sets of nut/bolt, and a center screw. For simplicity, four metal contacts and the center screw are not used in the succeeding illustrations.

4.2 Generating Default Design Process

Once descriptions of a design and associated assembly operations are entered into the system, the next step is to generate a default design process in order to start the process of (re)design by *replay and modify* [14]. A default design process is defined as a probable sequence of design decisions and their justifications that could have created the given design.

Vocabularies describing different design decisions are predefined in terms of a precondition-and-effect operators. Different types of design decisions are grouped into different design classes, starting from functional design to detailed form design, and further subdivided into 11 design stages as shown in Table 1. Each design stage is composed of series of design decisions of which are responsible for filling in certain details of the design appropriate for its stage. A heuristic algorithm, based on a generic design process model, selects and schedules probable design decisions, and constructs design states associated with them. The reconstruction starts from a default null initial design state, and ends at a final state where all of design objects and design information have been accounted for. The detailed design process model is shown in Table 1.

4.3 Incremental Design Analysis

A redesign first proceeds by incrementally generating each stage of the design process and perform-

Order	Design Stages	Synopsis
1	create-fn	create non-assembly related functions
2	create-fcn	create conceptual part for non-assembly related functions
3	create-assembly-fn	create assembly related functions
4	create-assembly-fcn	create conceptual liaisons (mating relationships)
	assembly planning	preliminary assembly planning and analysis wrt. parallelism
5	create-subassembly-fcn	create conceptual subassemblies after preliminary assembly planning
6	select-fcn	apply values for attributes of design objects and select type of their assembly processes
7	select-liaison-fcn	select types of matings for liaisons created in stage 4, and select attributes for attributes for corresponding assembly processes
8	create-assembly-fcn	create conceptual form for parts or features needed for mating types created in stage 7
9	select-assembly-fcn	approximate values for attributes of parts or features created in stage 8, and select type of assembly processes
	assembly planning	preliminary assembly planning and comparative analysis wrt. subassembly stability and directionality
10	select-subassembly-fcn	apply values for attributes of subassemblies created in stage 6 and select type of assembly processes
11	detail-fcn	finalize design by determining more detailed aspects of design objects
	assembly planning	final assembly planning and analysis

Table 1: Design process in INSPiRE-2.

ing a design analysis (in this case, a DFA analysis) to find any problematic design decisions within a given design stage. Each problematic design decision will be marked so that a case-based redesign can be attempted. A redesign may result in regenerating a new design process. All problematic design decisions are addressed before moving to the next design stage. At each design stage a local or global design analysis may be performed.

4.3.1 Local DFA Analysis

A local design analysis refers to a process of identifying any DFA problems that are assembly sequence independent. A local DFA analysis looks for design decisions that are responsible for, for example, selecting expensive type of mating, creating non-necessary parts, selecting design attributes that contribute to difficult handling and special processes. A rule based system has been implemented which examines design decisions and marks (See Figure 6) them if problematic.

4.3.2 Global DFA Analysis

On the other hand, a global analysis refers to a process of identifying any DFA problems that are assembly sequence dependent. Therefore, a global analysis involves performing assembly planning and selecting

preferred subassemblies. When subassemblies are selected, design decisions involving them are generated and analyzed, as if they were individual parts/features in the manner described in the preceding subsection. Problematic design decisions for subassemblies may include those responsible for missing guidance features, unstable subassemblies, and high number of assembly directions.

The following conditions (subset of Lee's conditions [12]) are used to identify feasible subassemblies for producing all possible subassemblies. For a cluster of parts, S , that belongs to an assembly A (denoted $S|A$) to be a subassembly of A if, (1) $S|A$ can be brought to $A - S|A$ from free space for mating, (2) interconnection defined between $S|A$ and $A - S|A$ is feasible, (3) all matings between $S|A$ and $A - S|A$ have at least one common axis of separation, and (4) $S|A$ and $A - S|A$ are stable either by itself or by external devices. Conditions 1 and 2 are computed automatically using a non-guaranteed algorithm based on simple interference check along a projected path in six principal directions. Condition 3 is checked by querying a knowledge base of various interconnections, while condition 4 is manually checked.

Then, an AO^* search is performed to find the preferred subassemblies. The global DFA analysis, therefore, corresponds to the heuristic function used in the AO^* search described in [12]. In this paper, we introduce a simpler version of Lee's heuristic function [12] that does not include local cost (c_s and c_t), since local analysis is provided separately as described in Section 4.3.1. The search space to which the AO^* algorithm is applied can be represented by an AND/OR tree. The decomposition of an assembly A implies the expansion of an AND node (representing an assembly A) into its OR children representing the alternative decompositions of A . Figure 4 shows a AND/OR tree for feasible subassemblies of the switch box example. A potential solution tree is an AND tree¹ having the minimum value for the evaluation function at the current stage of search, whereas a solution tree is an AND tree with leaves consisting of only single parts.

To formulate the evaluation function, e_f , for the AO^* algorithm, let us introduce the following definitions:

Definition: The Heuristic Estimate, $h_e(n)$, associated with an OR node, n , represents an estimate of the optimal relative assembly cost to assemble $S|A$ and $A - S|A$, and can be computed by the weighted sum of the following components.

¹An AND tree is an AND/OR tree where every AND node has no more than one OR child

- 1) The relative assembly cost, $R_o(S|A)$ and $R_o(A - S|A)$, due to the number of reorientations involved in the assembly of $S|A$ and $A - S|A$.
- 2) The relative assembly cost, $R_s(S|A)$ and $R_s(A - S|A)$, due to the number of stacking and non-stacking operations involved in the assembly of $S|A$ and $A - S|A$.
- 3) The relative assembly cost, $I(S|A)$ and $I(A - S|A)$, associated with the stability of $S|A$ and $A - S|A$, where²,

$$I(S|A) = \frac{\text{no. of floating liaisons in } S|A}{\text{ave. deg. of a node in } S|A}$$

- 4) The effect of parallelism, $w_p(S|A)$, where,

$$w_p(S|A) = |N_1 - N_2| / (N_1 + N_2),$$

where N_1 and N_2 represent the number of parts in $S|A$ and $A - S|A$, respectively.

Definition: The evaluation function, $e_f(T)$, associated with an AND tree T , is equal to sum of $h_e(n)$ for all n , $n \in T$.

The content of the input representation for both design and assembly operation is assumed to be obtainable except for attribute values that are not usually available from standard design documents, including DFA characteristics of parts and subassemblies such as handling and orienting costs, stacking, and part symmetry. The values of these attributes are obtained from the user during the analysis process using a question/answer form (similar to those of Boothroyd DFA software and Sapphire's Assembly View[2] [19]) in order to compute the heuristic function.

4.3.3 Comparative Analysis

Instead of just selecting the current best assembly plan and continue to (re)design for it, several candidates for assembly plans may be kept and analyzed with respect to an evolving design. For instance, the second best assembly plan may be better than the best in terms of number of fixtures required, but only worse in terms of number of stacking operations. Each assembly plan candidate requires different redesign strategies and, therefore, after redesign, a previously "worse" assembly plan may become "better". Therefore, keeping a multiple number of assembly plans and (re)designing them involves keeping multiple threads of (re)design processes.

² An assembly is represented by a liaison graph, where nodes correspond to parts, and arcs correspond to liaisons or mating relationships among parts. A non-floating liaison requires external connectors, forces or mechanisms for interconnecting the associated parts, while a floating liaison does not.

Figure 4 shows an AND/OR graph generated right after design stage 9. It contains two feasible subassemblies for the switch box example (other possibilities are omitted). The first possibility (upper half of the AND/OR graph), hence called *plan 1*, is to assemble one part at a time, and the other possibility (lower half of the AND/OR graph), hence called *plan 2*, splits the assembly into two groups of subassemblies. At design stage 9, *plan 2* is the better than *plan 1* as indicated by the lower e_f value shown on the top OR node of the corresponding AND/OR subtree.

For a comparative analysis, one analysis with respect only to directionality (stacking) and another with respect only to subassembly stability are performed. Figure 5(a) and Figure 5(b) shows the resulting AND/OR graphs respectively. Figure 5(a) shows that *plan 1* is as good as *plan 2* as far as directionality is concerned, although it contains two unstable subassemblies, namely due to two floating liaisons between the case and two contacts, *contact1* and *contact2*. This is automatically found by examining the corresponding AND/OR subtree and looking for a significant jump (arbitrarily defined) in the accumulated heuristic measure, e_f . Appropriate problem descriptions are generated and attached to the corresponding responsible design decision. Figure 6 shows problems mapped on one of the corresponding design decisions *select-form-case-contact1-liaison*. Note that these problems are only relevant and meaningful in the context of using *plan 1*.

4.3.4 Redesign

As for the switch box example, there will be two separate processes for redesign, since two distinct assembly plans are being considered in this particular example (*plan 1* and *plan 2*).

After the global comparative analysis, it is found, for *plan 1*, that two loose mating relationships (floating liaisons) between the case and metal contacts are causing instability in the respective subassemblies.

A simple redesign is proposed and executed interactively by the case-based redesigner to change the type of liaison to a force-fit (instead of insertion) to provide stability. A new assembly plan evaluation is performed³. As shown in Figure 7, *plan 1* is now as good as *plan 2* with regards to a global analysis.

Other problems that are found and solved for each redesign process are summarized in Table 2 and Figure

³ If a redesign involves introduction of new parts or liaisons, then a new assembly planning needs to be performed to identify new subassembly groupings. Otherwise, only a new cost evaluation is needed.

8.

Plan 1		Plan 2	
Problem	Solution	Problem	Solution
Expensive mating btw case and panel	snaps fit	Expensive mating btw case and panel	they fit
Expensive mating btw panel and corresponding subassembly	larger feature size		
No guidance features btw panel and corresponding subassembly	chanfer		

Table 2: Problems and proposed solutions for *plan 1* and *plan 2*.

Note that the second and third redesign solutions in Table IV are only applicable in the context of subassembly groupings of *plan 1*. Widening of slot sizes in the *panel* for *plan 2* would make the subassembly of *panel* and metal contacts unstable.

5 Conclusion

In this paper, we have presented an integrated system of (re)design and assembly planning, called INSPIRE-2. INSPIRE-2 has been implemented in COMMON LISP (about 15,000 lines of code). INSPIRE-2 is more than a mere software integration, since the activities of assembly planning and (re)design are interleaved during the (re)design process under a single framework.

Interleaving preliminary assembly planning with design can result in a reduction of cycle time between artifact design and process design. According to Keshavan [9], manual assembly planning usually goes through many iterations (typical 6-10), and plan maintenance (in response to design changes) is about 5 times the cost of plan generation. Traditionally, process design follows after an artifact design is completed. In a worst case scenario, this results in a vicious cycle of requests for design revisions and degradation in the level of automation.

Although conventional DFA methods have proven to be effective analysis tools, they do not provide a framework for considering alternative assembly plans and subassembly groupings. Consequently, a correct analysis may not be possible since some DFA criteria must be measured with respect to an actual assembly sequence. Combining design and assembly planning is a natural step to both realize a more global DFA analysis and effective redesign strategies.

Certainly, there are still many number of problems that need to be resolved in order to further enhance the system in terms of its usability and effectiveness.

Current research efforts are focused on improving and expanding representations, particularly for describing mating relationships and assembly design operators, automating the process of determining motion constraints, subassembly poses and stability to reduce number of required user queries and providing visual and geometric feedback for newly proposed designs.

References

- [1] D. F. Baldwin, T. Abell, M. Lei, T. De Fazio, and D. Whitney. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. *IEEE Trans. Robotics and Automation*, pages 18-24, February, 1991.
- [2] C. Boothroyd and P. Dewhurst. *Product Design for Assembly Handbook*. Boothroyd and Dewhurst, Inc., 1989.
- [3] N. Chao and S. Lu (Ed.). *Concurrent Product and Process Design*. ASME, 1995.
- [4] M. Cukusky and J. Tannenbaum. A methodology and computational framework for concurrent product and process design. *Mechanism and Machine Theory*, 25, 1990.
- [5] L. S. Homem de Mello and A. C. Sanderson. Representations of assembly sequences. *Proceedings of International Joint Conference on Artificial Intelligence*, 1989.
- [6] J. Gul and M. Mantyla. Functional understanding of assembly modeling. *CAD*, 26:588-591, 1994.
- [7] J. M. Hensford, F. Bonaccelli, and A. Bouzjaneh. Evaluation and selection of assembly plans. *APMS '90*, pages 212-218, Aug., 1990.
- [8] W. Han, C.S.G. Lee, and S. Su. Feedback evaluation of assembly plans. Technical Report, TR-ERC-92-3, Purdue University, 1992.
- [9] K. Keshavan. Current practices of assembly planning in industry. *ISRA Workshop on Assembly and Task Planning*, 1993.
- [10] G. Kim and G. Bekey. Design-for-assembly by reverse engineering. *Artificial Intelligence in Design*, 9, pages 717-728, 1994.
- [11] K. Lee and D. Gossett. A hierarchical data structure for representing assemblies. Part I. *Computer-aided Design*, 17:13-19, 1985.
- [12] S. Lee. Backward assembly planning with dfa analysis. *Computer-Aided Mechanical Assembly Planning*, Chap. 14, 1991.
- [13] S. Lee and G. Bekey G. Kim and. Combining assembly planning with redesign: An approach for more effective DFA. *Proceedings of IEEE International Conference on Robotics and Automation*, invited Paper, 1993.
- [14] J. Mestaw. Design by derivational analogy: issues in the automated reply of design plans. *Artificial Intelligence*, 48:119-124, 1999.
- [15] J. Nevins and D. Whitney. *Concurrent Design of Products and Processes*. McGraw-Hill Publishing Co., 1989.
- [16] R. Popplestone, A. Ambler, and I. Belle. An interpreter for a language for describing assemblies. *Artificial Intelligence*, 14:79-107, 1980.
- [17] B. Romney, C. Godard, M. Goldwasser, and G. Ramkumar. An efficient system for analyzing assembly complexity. Manuscript, Stanford University, 1994.
- [18] R. Sodhi and J. Turner. Towards modelling of assemblies for product design. *CAD*, 26:63-77, June, 1994.
- [19] Sapphire Design Systems. *User Manual of Assembly View*. Sapphire Design Systems, Inc., 1989.
- [20] J. Tsai, H. Koukar, and M. Cukusky. Issues in incremental analysis of assemblies for concurrent design. *Proceedings of AI in Design*, 82, pages 817-825, 1992.

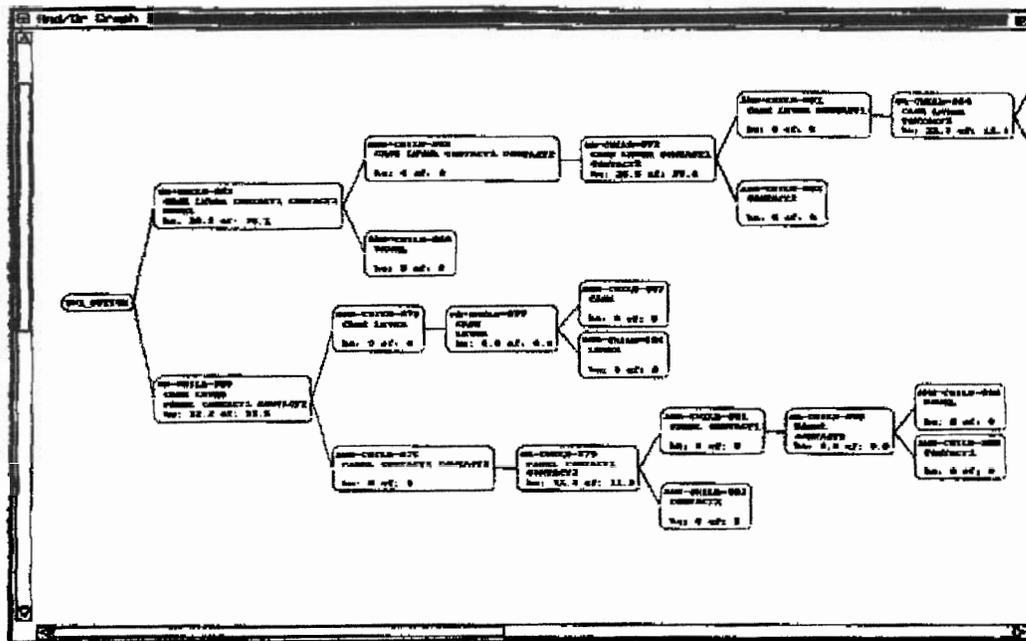
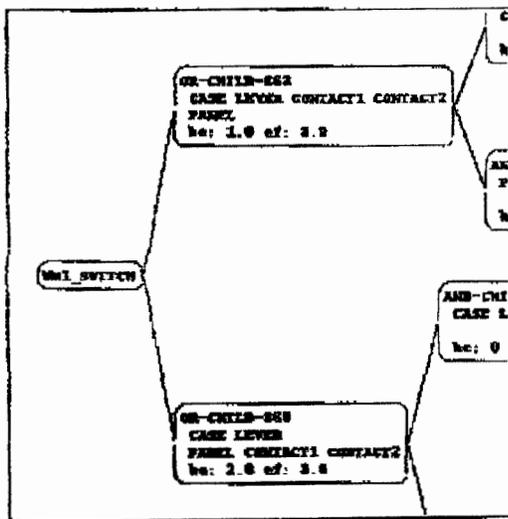
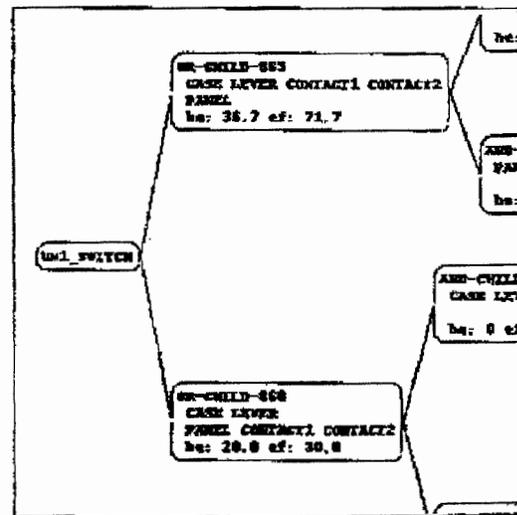


Figure 4: An AND/OR tree for feasible subassemblies of the switch box example. Only two possible subassembly grouping *plan 1* and *plan 2*) is shown. For each OR node, the corresponding subassembly pairs and heuristic measures are printed.



(a)



(b)

Figure 5: (a) An AND/OR tree for feasible subassemblies of the switch box example. Analysis is performed only with respect to directionality, that is, how stacking the subassemblies are. (b) An AND/OR tree for feasible subassemblies of the switch box example. Analysis is performed only with respect to instability of subassemblies.

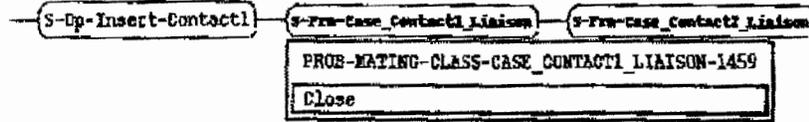


Figure 6: Problematic design decisions in design stage 7 marked by the comparative analysis.

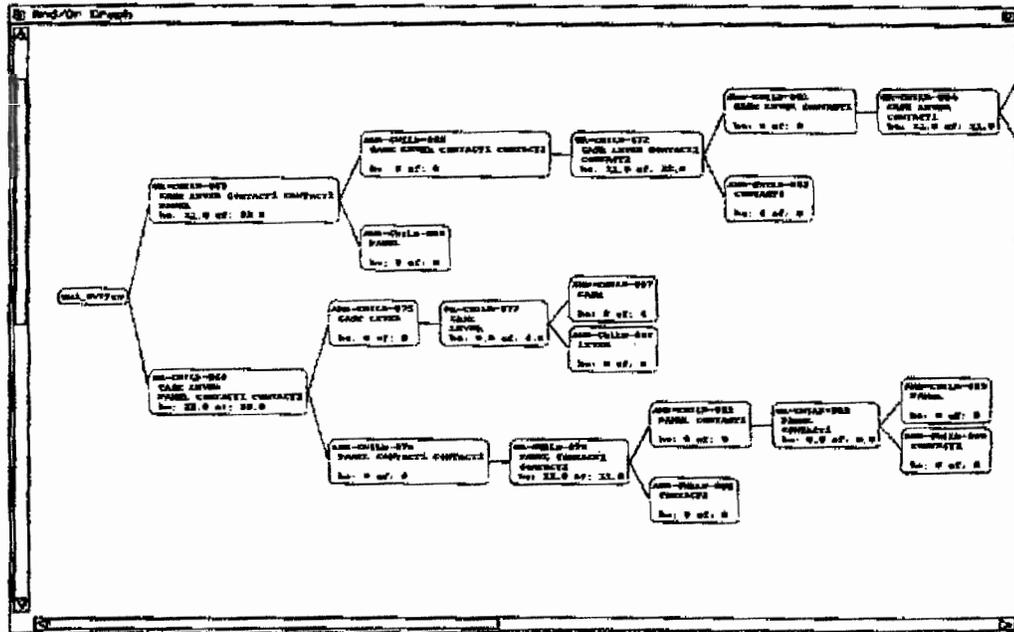


Figure 7: An AND/OR tree for feasible subassemblies of the switch box example after redesign. At this point, plan 1 is as good as plan 2.

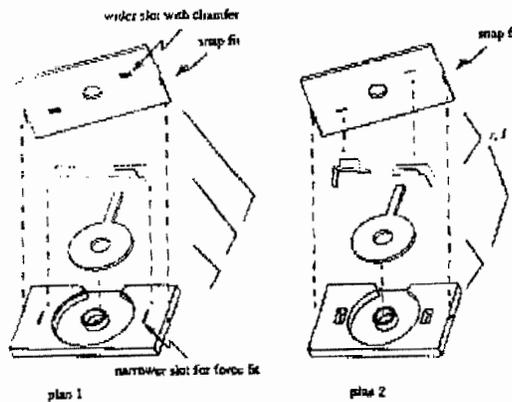


Figure 8: Two possible redesigns for two assembly sequence candidates realized with Pro/Engineer CAD systems following redesign advice from INSPIRE-2. In the end, plan 1 results in lower overall cost because it no longer requires fixtures.