Synthesis of Spacecraft Designs

Erik K. Antonsson, Ph.D., P.E.
Executive Officer and Professor of Mechanical Engineering
Division of Engineering and Applied Science
California Institute of Technology
Pasadena, CA 91125
April 30, 1998
Traditional Design Process

Adapted from:
G. Pahl and W. Beitz

*Engineering Design*

The Design Council, Springer-Verlag
New York, 1984, page 41
A New Approach to Synthesis is needed to Achieve One Launch/Month.
syn.the.sis n, pl -the.ses [Gk, fr. syntithenai to put together, fr. syn- + tithenai to put, place–more at do] (1589)

1 a: the composition or combination of parts or elements so as to form a whole

1 b: the production of a substance by the union of chemical elements, groups, or simpler compounds or by the degradation of a complex compound

1 c: the combining of often diverse conceptions into a coherent whole; also: the complex so formed

2 a: deductive reasoning

2 b: the dialectic combination of thesis and antithesis into a higher stage of truth

3: the frequent and systematic use of inflected forms as a characteristic device of a language – syn.the.sist n

Merriam-Webster’s Collegiate Dictionary
Copyright ©1994 Merriam-Webster, Incorporated
**com.pile** *vt* **com.piled** ; **com.pil.ing* [ME, fr. MF *compiler*, fr. L *compilare* to plunder] (14c)

1: to compose out of materials from other documents

2: to collect and edit into a volume

3: to run (as a program) through a compiler

4: to build up gradually <compiled a record of four wins and two losses>

**com.pil.er** *n* (14c)

1: one that compiles

2: a computer program that translates an entire set of instructions written in a higher-level symbolic language (as Pascal) into machine language before the instructions can be executed

**com.pi.la.tion** *n* (15c)

1: the act or process of compiling

2: something compiled

Merriam-Webster’s Collegiate Dictionary
Copyright ©1994 Merriam-Webster, Incorporated
Software Compilation

```c
#include <stdio.h>
#include <math.h>

main()
{
    float x, y;
    x = 30.0 * 3.14159 / 180.0;
    y = sin(x);
    printf("%f \n", y);
}
```

Assembly code

```
.globl main
main:
    .type main,.function
    .proc 04
    .align 8
    .file "jpl.c"
gcc2_compiled:
    .section ".rodata"
    .align 8
    .asciz "%f \n"
    .align 4
    .section ".text"
    .align 4
    .global main
    .type main,.function
    .proc 04
    .file   "jpl.c"
gcc2_compiled:
    .section        ".rodata"        
    .align 8
    .LLCI:
        .asciz  "%f \n"
        .align 4
    .LLCO:
        .long   0x3f060a8a
        .section   ".text"
        .align 4
        .global main
        .type main,.function
        .proc 04
    main:
        .PROLOGUE # 0
        save %esp, -120, %sp
        .PROLOGUE # 1
        .sethi %hi(LLCO), %0
        ld { %0, %lo(LLCO), %1
        st { %1, %fp(20)
        ld { %fp(20), %2
        fstd %2,%6
        std %6,[ %fp(16)]
        ldd [%fp(16), %2
        mov %0, %0
        mov %b3, %1
        call sin, 0
        nop
        fmovs %0,%2
        fmovs %1,%3
        fdtos %2,%4
        st %4,[ %fp(24)]
        ld [%fp(24), %2
        fstd %2,%6
        std %6,[ %fp(16)]
        ldd [%fp(16), %2
        .sethi %hi(LLCI), %0
        or %0, %lo(LLCI), %0
        mov %0, %1
        mov %b3, %2
        call printf, 0
        nop
    .LL1:
        ret
    restore.
    .LLfe1:
    .size main, LLfe1-main
    .ident "GCC: (GNU) 2.7.2.3"
```

Machine code
Digital VLSI Compilation

- Specify Desired (Boolean Logic) Function
- Automated Procedures to produce Mask-Layouts from standard components (logic gates, shift registers, memory cells, etc.)

(Mead and Conway, *Introduction to VLSI Systems*, 1980)
Chemical Process Plant Compilation

- Specify Desired Chemical Process Function
- Automated Procedures to produce Plant Flow-Chart from standard components (reactors, pumps, distillation columns, etc.)
Hand-held Electric Drill Compilation

Figure 5. FFREADA's optimal design configurations

HVAC Layout Compilation

Figure 1. Simulated annealing placed components routed by a) classification-based tube routing algorithm, b) simulated annealing-based tube routing algorithm, c) the most desirable tubes from both approaches.

The optimum solution, with cost 441.97, is:
For POWER-SUPPLY, US-3Ph-220 with cost 0
For MOTOR, 3N593 with cost 192.72
For GEAR-PUMP, TYPE-103 with cost 133.0
For VALVE, TYPE-1 with cost 50.0
For CYLINDER, 1.25 with cost 6.25
For VALVE-2, TYPE-1 with cost 50.0
For CYLINDER-2, diameter 2.0 with cost 10.0

(Ward and Seering, “The Performance of a Mechanical Design Compiler”, MIT Artificial Intelligence Laboratory memo 1084, 1989)
Current Related Work at Caltech

- Refinement of Design Imprecision (Fuzzy Sets)
- Set-Based Design
- Rapid Assessment of Early Designs (RAED)
- Genetic Algorithms in MEMS Synthesis
- Compilers for Mechanical Systems
Current Related Work at Caltech

Refinement of Design Imprecision (Fuzzy Sets)
Current Related Work at Caltech
(Fuzzy) Set-Based Design

DVS  
(Design Variable Space)  
PVS  
(Performance Variable Space)

Design Exploration  
Point by Point

Design Exploration  
Sets at a Time
Aggregation Operator Axioms

At each point $\vec{x}$ the following hold:

1. **Monotonicity:**
   \[ P(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) \leq P(\mu_1, \mu'_2; \omega_1, \omega_2)(\vec{x}) \quad \forall \mu_2(\vec{x}) \leq \mu'_2(\vec{x}) \]
   \[ P(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) \leq P(\mu_1, \mu'_2; \omega_1, \omega_2')(\vec{x}) \quad \forall \omega_2 \leq \omega'_2; \; \mu_1(\vec{x}) < \mu_2(\vec{x}) \]

2. **Symmetry:**
   \[ P(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) = P(\mu_2, \mu_1; \omega_2, \omega_1)(\vec{x}) \]

3. **Continuity:**
   \[ P(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) = \lim_{\mu'_2(\vec{x}) \to \mu_2(\vec{x})} P(\mu_1, \mu'_2; \omega_1, \omega_2)(\vec{x}) \]
   \[ P(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) = \lim_{\omega'_2 \to \omega_2} P(\mu_1, \mu_2; \omega_1, \omega'_2)(\vec{x}) \]

4. **Idempotency:**
   \[ P(\mu, \mu; \omega_1, \omega_2)(\vec{x}) = \mu(\vec{x}) \quad \forall \omega_1 + \omega_2 > 0 \]

5. **Annihilation:**
   \[ P(\mu, 0; \omega_1, \omega_2)(\vec{x}) = 0 \quad \forall \omega_2 \neq 0 \]

6. **Self-scaling weights:**
   \[ P(\mu_1, \mu_2; \omega_1 t, \omega_2 t)(\vec{x}) = P(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) \quad \forall \omega_1 + \omega_2, t > 0 \]

7. **Zero weights:**
   \[ P(\mu_1, \mu_2; \omega_1, 0)(\vec{x}) = \mu_1(\vec{x}) \quad \forall \omega_1 \neq 0 \]
Current Related Work at Caltech
Rapid Exploration of a Large Design Space
A pillar thickness

\[ \mu_{\text{max} \, \text{achievable}} = 0.440 \]

B pillar thickness

\[ \mu_{\text{o \, selected}} = 0.306 \]

Floor sill thickness

Floor pan thickness

B pillar location

\[ \mu_{\text{O}} \]

A Pillar

0.9

B Pillar

1.1

Floor Sill

1.0

Floor pan

1.2

B Pillar Location

50

Close
A pillar thickness

$\mu_{o, \text{max}} \text{ achievable} = 0.440$

$\mu_o \text{ selected} = 0.440$

B pillar thickness

Floor sill thickness

Floor pan thickness

$\mu_o$

B pillar location

$\mu_o$
A pillar thickness

\[ \mu_{\text{max}} \text{ achievable} = 0.440 \]

\[ \mu_{\text{o}} \text{ selected} = 0.384 \]

B pillar thickness

Floor sill thickness

Floor pan thickness

B pillar location

design preferences

A Pillar

B Pillar

Floor Sill

Floor pan

B Pillar Location

Reload

Info

Prefs

Close
A pillar thickness

$\mu_{\text{max}}$ achievable = 0.440

B pillar thickness

$\mu_{\text{selected}}$ selected = 0.394

Floor sill thickness

design preferences

Floor pan thickness

B pillar location

Design preferences

A Pillar

0.8

B Pillar

0.9

Floor Sill

0.9

Floor pan

1.2

B Pillar Location

50
A pillar thickness:

$\mu_{\text{max}}$ achievable = 0.440

$\mu_{\text{selected}}$ selected = 0.440

B pillar thickness:

Floor sill thickness:

Floor pan thickness:

B pillar location:

design preferences
Current Related Work at Caltech
Rapid Exploration of a Large Design Space

\[ \mu_0^* = 0.40159 \]

Floor Thickness
B pillar location

MiniCommand Window:
global DATA
\[ dv(1) = 0.7; i1=1; \]
Current Related Work at Caltech
Rapid Exploration of a Large Design Space

Empirical data: B=0.00, F=0.75, theta=15 fixed
Aggregated with BR=0, FR=0.75, ConeAngle=15 fixed, mu(free molec.)=0
Current Work Elsewhere

- “Constrained Three Dimensional Component Layout Using Simulated Annealing”, Jon Cagan, CMU
- “Agent-Based Design”, Jon Cagan, CMU
- “Using Pareto optimality to coordinate distributed agents”, Mark Cutkosky, Stanford
- “Set-Based Concurrent Engineering”, Al Ward, U.Michigan
- “Chain Models of Physical Behavior for Engineering Analysis and Design”, Vadim Shapiro, U.Wisconsin, Madison
- “Synthesis of Schematic Descriptions in Mechanical Engineering”, Warren Seering, MIT
- “Genetic Algorithm Attributes for Component Selection”, Susan Carlson
- “Quantitative Inference in a Mechanical Design Compiler”, Al Ward, U.Michigan
- “Structured Synthesis for MEMS”, Gary Fedder, CMU
Current Work Elsewhere

9th Int’l Conf. on Solid State Sensors and Actuators (Transducers ’97)
Chicago, IL, June 1997, pp. 1109-1112.

Automated Optimal Synthesis Of Microresonators

G. K. Fedder, S. Iyer and T. Mukherjee
Department of Electrical and Computer Engineering and The Robotics Institute
Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890

ABSTRACT

The rapid layout synthesis of a microresonator from high-level functional specifications and design constraints is demonstrated. Functional parameters such as resonant frequency, quality factor, and displacement amplitude at resonance are satisfied while simultaneously minimizing an objective function. The optimal synthesis tool allows exploration of micromechanical design issues and objectives, as illustrated with a polysilicon lateral resonator example modeled in three mechanical degrees-of-freedom. Layouts for three sets of five different resonators from 3 kHz to 300 kHz are generated, with each set globally optimized to minimize either active device area, electrostatic drive voltage, or a weighted combination of area and drive volt-age.
Current Work Elsewhere

OPTIMIZED PORTABLE ALGORITHMS AND APPLICATION LIBRARIES (OPAAL) INITIATIVE FOR COMPLEX PHYSICAL SIMULATION

Algorithms And Libraries For Virtual Prototyping And Simulation

DEADLINES:

May 22, 1998: e-mail letter of intent
July 1, 1998: Proposal receipt

NSF 98-64

INTRODUCTION

The Division of Mathematical Sciences (in collaboration with the Office of Multidisciplinary Activities of the Directorate for Mathematical and Physical Sciences, the Directorate for Computer and Information Science and Engineering (CISE) and the Directorate for Engineering (ENG)) of the National Science Foundation (NSF) and the Defense Sciences Office of the Defense Advanced Projects Research Agency (DARPA) plan jointly to support research and development of new approaches to the design and creation of efficient algorithms and optimized libraries for large-scale numerical modeling and simulation of physical phenomena arising in industrial applications.

The Optimized Portable Algorithm and Application Libraries (OPAAL) initiative is an opportunity for researchers in the mathematical sciences to join with other scientists and engineers in the development of innovative mathematical techniques applicable to simulation of complex physical processes. Fresh ideas, approaches, techniques, the formation of new teams, and the participation of researchers new to such problem areas are particularly encouraged. Areas of mathematics which are not currently considered “applied” are a likely source of such fresh ideas, approaches and techniques.
Current Work Elsewhere

NSF/DARPA OPAAL Continued . . .

The use of modeling and simulation by industry to speed up and reduce the cost of design of new products and systems has become a well-accepted paradigm in many areas, with explosive growth expected in the years to come. The ability to reduce the amount of “cut and try” in design will depend on the accuracy, speed, robustness, and affordability of computational alternatives.

Major bottlenecks in numerical simulation include geometry representation, grid generation, the accuracy of numerical solutions of partial differential equations, and the efficiency and speed of numerical methods when implemented on specific computer architectures. Areas of “core” mathematics playing potential roles here would be geometry, analysis and algebra. Historically, it has been difficult to realize algorithms that simultaneously guarantee low computational complexity, low memory usage, accuracy, high efficiency, and portability across diverse computer architectures. Algorithmic design is further complicated by the disparity in mathematical, computational, and application considerations that often work at cross purposes when addressed in a sequential or compartmentalized fashion. In addition, the labor intensive nature of implementing numerical methods to run efficiently on modern platforms having deep memory hierarchies and multiprocessor architectures results in significant costs and delays for the applications community.

Major technical themes of the initiative include:

- Mathematical representations of complex surfaces and volumes to facilitate Computer Aided Design (CAD) and discretization;
- Well-conditioned, scalable, high order accurate numerical methods;
- Mathematical abstractions and computational models for automatic generation of OPAAL; and
- Demonstrations and technology transfer.
Proposed Work: NASA ISE Program

- Automatic Synthesis/Compilation of Spacecraft Configuration/Design
  - Parameterized Modules
  - Choose Modules
  - Choose Parameters

- Automatic Synthesis/Compilation of Mechanical Transmissions
Sobieski Paradox (1988)

100%

$ Committed

Improved Modeling

Knowledge of the Problem

Improved Design Processes

Time
ISE Improvement Measure

Faster

Better

Cheaper