# NSF Workshop on Design Methodologies for Solid Freeform Fabrication

**June 5-6, 1995**

**Engineering Design Research Center**
**Carnegie Mellon University**
**Pittsburgh, PA**

**Sponsored by**

**The National Science Foundation Computer and Information Science and Engineering Directorate Microelectronics Information Processing System Division**

# NSF Workshop on Design Methodologies for Solid Freeform Fabrication

**June 5-6, 1995**

**Engineering Design Research Center**
**Carnegie Mellon University**
**Pittsburgh, PA**

**Sponsored by**
**The National Science Foundation Computer and Information Science and Engineering Directorate**
**Microelectronics Information Processing System Division**

## Telephone Device for the Deaf

The National Science Foundation has a Telephonic Device for the Deaf (TDD) capability which enables individuals with hearing impairment to communicate with the Division of Personnel and Management for information relating to NSF programs, employment, or general information. This number is (703) 306-1636.

## Electronic Dissemination

You can get information fast through STIS (Science and Technology Information System), NSF's online publishing system, described in the "STIS Flyer" at the end of this document.

## Ordering by Electronic Dissemination

If you have access to either Internet or Bitnet, you may order publications electronically. Internet users should send requests to pubs@NSF.gov. Bitnet users should address requests to pubs@NSF. In your request, include the NSF publication number and title, number of copies, and your name and complete mailing address. Printed publications also may be ordered by FAX (703) 644-4278. Publications should be received within 3 weeks after receipt of request.

## Solid Freeform Fabrication Workshop Report URL

You can access the NSF Workshop on Design Methodologies for Solid Freeform Fabrication Report directly at **"http://www.edrc.cmu.edu/proc/DMSFF95/"**.

**NSF participants were involved in the technical discussions in the workshop but did not participate in the recommendations. The opinions expressed in these proceedings are those of the individual participants and do not necessarily represent NSF policy. Their recommendations are currently under review by NSF.**

# Table of Contents

# Workshop on Design Methodologies for Solid Freeform Fabrication June 5-6, 1995

## 1. Background and Workshop Operation

The Workshop on Design Methodologies for Solid Freeform Fabrication (SFF) was convened to examine existing design methodologies being used in SFF. The primary goal was to determine whether substantial benefit would arise from research into applying key elements from the VLSI experience to SFF rapid prototyping. The term "Solid Freeform Fabrication" means the production of freeform solid objects directly from a computer model without part-specific tooling or human intervention.

An NSF-sponsored workshop [1] held in May, 1994 found significant similarities between the layering processes used in VLSI and SFF, suggesting potential benefits from adapting some of the VLSI design methodologies. A position paper on "Design Methodology for Rapid Prototyping," by Drs. Bernard Chern and Jack Hilibrand, was prepared and distributed prior to the SFF Workshop. (See Section 4.3.) Their paper identified some of the key elements applicable to SFF design and raised a number of issues and questions.

The workshop was hosted by Dr. Daniel P. Siewiorek, Director of the Engineering Design Research Center at Carnegie Mellon University. Prior to the workshop, copies of the Chern and Hilibrand position paper were circulated to a group of interested researchers who were invited to attend the two day workshop. Attendees were asked to submit position papers defining their views of the present status of design methodology for SFF and discussing ways to apply the VLSI experience to advance the field of SFF. Some of these position papers were circulated to the attendees for their consideration in advance of the workshop.

The workshop opened with a charge from Dr. Bernard Chern (See Section 1.1). Five overview presentations were presented to provide a common terminology and understanding to the diverse group of researchers. Each workshop participant then joined a breakout group dealing with one of the following classes of SFF technology:

- Stereolithography
- Laser-based SFF
- Shape Deposition Manufacturing
- 3D Printing
- Lamination

Each of the five groups met during the afternoon of June 5 and the morning of June 6. Each group considered the key questions stated in the Workshop Charge and prepared a response discussing the design methodology in use in their technologies and the needs of the technology community. The breakout groups met in plenary session at the end of the first day and at lunch time on the second day. In the plenary sessions, each group presented the materials they were working on and provided feedback to the entire workshop. The chairmen and the breakout group members have worked on formalizing their responses, which are presented in Section 3.0.

Section 1.1 of this Proceedings reproduces the Workshop Charge with the six key questions. The remainder of the Proceedings consists of an Executive Summary and the breakout group reports. The workshop presentations, the original position papers, the list of attendees and the workshop agenda are in the Appendices.

It is hoped that this workshop will open the field for extended discussion of new and improved ways to design mechanical systems for fabrication using SFF technology.

## 1.1 Charge to Workshop - Bernard Chern

The goal of research in rapid prototyping is to develop and integrate methodologies, tools, environments and technologies needed to be able to automate the design and construction of processes, artifacts and systems of artifacts rapidly and efficiently. A key long term objective is to develop a design methodology that can be applied generally to mechanical and electro-mechanical systems.

This workshop is intended to identify and encourage research efforts on implementing SFF design methodologies. The scope of these efforts, however, includes not just the design methodology itself, but also the design tools, design environments and design technologies that will be available for rapid prototyping using SFF.

At this workshop we will grapple with research needed to create a VLSI-like design methodology for the SFF technologies in which there is a clean separation between design and fabrication; that is, between the design community with its concerns about CAD tools, design environments, *etc.* and the fabrication community with its concerns about equipment and processing capabilities as well as such customer servicing criteria as cost and responsiveness. To achieve such an outcome we will need to answer some key questions:

1. Can we use the same generic layering model for all SFF fabrication processes?

2. Can we identify a common digital specification language that can be used generally to describe the desired prototype in terms of the resulting geometry on each layer?

3. Can we use this digital interface to achieve a clean separation between the design and processing activities?

4. What is the nature of a set of design rules that might be generally applicable in determining which SFF processes can provide assurance of manufacturability in building a given design?

5. What levels of abstraction are appropriate to mechanical design and how can we improve the ease of moving among these levels?

6. Can we make the design tool hierarchy independent of process change without a negative impact on the process development activities? Are the fabricators driven by the design community through interaction across the 'clean separation' interface?

A successful workshop for developing a common design methodology for the SFF technologies will result in:

- A strong argument for an NSF research program to focus on making the SFF technologies increasingly effective for rapid prototyping.

- Definition of a common low-level layer-based digital interface descriptive language for SFF technology implementations. This can be a great advantage, making a variety of SFF technologies available to the designer (without learning a new interface language) and making many customers available to the SFF fabricator without the need to invest time and energy bringing the designer up to speed to address this technology. Further, modifications and advances in the SFF process area could be accommodated within the same design framework, merely involving changes in the design rules. Also the digital interface language would permit design submission over a network so that brokerage services, such as MOSIS become practical.

- Steady accumulation in the industry of feature and object description libraries, in a common language, that can be incorporated in the design heritage of the field and that will encourage the refinement of a hierarchical design methodology that will make them useful to the entire design community.

  Momentum to create paths from the higher level descriptions (in solid geometry and feature-based languages) to this digital interface language.

  Momentum to create paths (algorithms and translation code) to go from this digital interface language to the languages that drive the SFF fabrication systems.

  Refocusing by those active in SFF-based prototyping from the nuts and bolts of the design process to the optimization of system implementation along the dimensions of time to market, multiple sourcing, reuse of design building blocks, de-skilling of low level design activities, etc.

  Lower cost, reduced delay and fewer errors for system implementation using SFF-based rapid prototyping technology. (Lower cost because the design time is reduced and more designs go down this design path to many competitive vendors; reduced delay and errors because the design path is significantly automated and takes advantage of design heritage.)

  Increased exploration of sophisticated product design alternatives because the time and cost for experimental implementation are brought within reach so that there will be more cycles of learning about market requirements incorporated in each product generation.

These techniques for using a low level digital descriptive interface in SFF can (potentially) and will (very likely) be translated into enhancements for more general mechanical system design where they will provide an impetus for extending design automation to ever-widening regions of mechanical design space.

# 2. Executive Summary

## 2.1 Background

Over the last three decades, the microelectronics industry has undergone unprecedented growth and has had an enormous impact on the nation, often called the "VLSI revolution." Research activities in both industry and academia have led to the rapid introduction of advanced semiconductor process technology, hierarchically structured design methodologies, automated design tools, simulation models and rapid prototyping techniques. One key to the rapid success of the VLSI development effort was the early definition (about 1970) of a digital interface that separated design efforts from the growing complexities of the fabrication processes. This allowed the designer to use process-independent design tools and methodologies.

An NSF-sponsored workshop was held in May, 1994 to examine the successful VLSI experience, the lessons learned, and their applicability to other design and fabrication activities, particularly electromechanical design and manufacturing. The major findings and recommendations of this workshop are contained in the Workshop Proceedings [1]. One of the findings of the Workshop reads as follows: "While there are no general paradigm or technology transfer opportunities across all of mechanical manufacturing, there are potentially important opportunities for synergy in restricted device and process domains. Two of the emerging fabrication processes - SFF (Solid Freeform Fabrication) and MEMS (MicroElectroMechanical Systems) - have significant similarities to VLSI manufacturing process and layering technology" [1]. To follow up on this finding and the associated recommendations, a workshop was held on June 5 and 6, 1995 on Design Methodologies for Solid Freeform Fabrication at the Engineering Design Center of the Carnegie Mellon University chaired by Professor Daniel Siewiorek.

Participants were invited from academia, industry and government. Attendees came from existing SFF fabrication facilities, from the mechanical design community, from the solid geometry community and from the VLSI community. The workshop opened with an initial charge from Dr. Chern of NSF, who emphasized the potential benefits from the development of well-structured design methodology and design tools for SFF processes. Dr. Chern posed several key questions regarding abstraction levels, design rules, digital interface, design languages and simulation models for SFF. To provide a broad background in SFF state of the art to the participants, a series of technical presentations were given on VLSI design methodology, specific SFF technologies, modeling for SFF, user needs and designer orientation. The workshop was then organized into five breakout groups, each group concentrating on one or more specific SFF technologies. The breakout groups initially sought some common ground for their activities and then focused on responding to the specific issues posed in the workshop charge. At the end of each of the two breakout sessions, each group reported on their results at a plenary session.

### Breakout Groups

SLA - Stereolithography
E. Antonsson (Chair), B. Chern, P. Fussell, N. Mankovich, H. Voelcker, S. Yencho

Sintering - Laser-based SFF (SLS and LADRP)
J. Beaman, R. Crawford (Chair), J. Glimm, P. Kulkarni, A. Kar, A. Mukherjee, J. Stivoric

SDM - Shape Deposition Manufacturing
S. Finger, E. Lutz, P. Padmanabhan, T. Peters (Chair), R. Sproull, G. Sussman, L. Weiss

3D Printing - Three Dimensional Printing
C. Kasabach, F. Prinz, R. Riesenfeld, E. Sachs (Chair), C. Sequin, J. Staudhammer

Lamination - Laminated Object Manufacturing (LOM) - D. Dutta, M. Burns, J. Hilibrand, V. Kumar, S. McMains, W. Newman, C. Pina, D. Siewiorek (Chair)

## 2.2 SFF Taxonomy

The workshop provided a unique forum to discuss and compare the various SFF technologies. All share the common feature of being layer-based; however, they vary in the degree to which they are amenable to the critical VLSI-like separation of design and processing activities, that is, the extent to which the manufacturing process is insensitive to the geometry of the object to be prototyped. The following categories were identified and are listed in order of decreasing ease of separating design and process:

- Freeform Accretion or Additive Processes
  - Stereolithography (SLA)
  - Three-dimensional Printing (3DP)
  - Selective Laser Sintering (SLS)
  - Sheet Based Processes (LOM- Laminated Object Manufacturing)
  - Extrusion (FDM- Fused Deposition Manufacturing)
  - Ballistic Particle Manufacturing (BPM)

- Formative Processes (combining accretion and subtraction)
  - Shape Deposition Manufacturing (SDM)
  - Laser Aided Freeform Accretion or Direct Rapid Prototyping (LADRP)

- Subtractive Processes
  - The traditional NC (Numerically Controlled) and CNC (Computer-based Numerically Controlled) processes do not belong to the class of SFF processes.

Although all of the SFF processes build one layer at a time, important differences exist among them in the nature of modeling and process planning. One key feature of the VLSI layering paradigm is that the processing steps are independent of the 2D layout geometry. This is also true for many SFF processes that have a uniform layer thickness and homogeneous material so the object geometry is decomposable into uniform sliced layers. On the other hand, if the object needs support structures for its layered growth or there are composition variations, a process planning step is necessary to prescribe the orientation of the object and support structure before it is sliced into layers.

Certain geometric shapes are unrealizable by some SFF processes. For example, objects with enclosed inner regions cannot be produced by some because unused material or powder will be left at the end of the process. Another issue is the layer thickness, which depends on the particular fabrication process. Both the minimum layer thickness and the maximum layer thickness depend on the particular SFF process. For some of the formative processes, each layered deposition must be followed by material removal and/or stress removal operations.

The question of how to handle, in the design stage, composite materials, mechanical property anisotropy and embedded components remains open for those SFF processes incorporating these complexities. The design rules at the layered description level are, of course, process-dependent and the digital interface separating design and fabrication will have to be at a level of abstraction, dealing with geometry, shape and functionality, that is appropriate in order to keep design and process considerations separated.

#### Finding

For many SFF processes the principle of independence of processing steps and the desired object geometry holds true. The independence of processing and geometry holds, to be specific, when the layer thickness is chosen to be uniform, the material is homogeneous and the object geometry is decomposable into sliced layers. However, for SFF processes with full 3D geometric and structural freedom, the digital interface separating design and fabrication has to be at a level higher than the layer level. By full 3D geometric and structural freedom we mean incorporating anisotropy, spatially distributed microstructure and non-uniform layered thickness.

#### Recommendation

Support basic research to understand the taxonomy and range of SFF processes and the applicability of the layering paradigm to each of them in order to identify a hierarchy of design abstractions that could lead to a structured design methodology for many, if not all, SFF processes.

## 2.3 Design Hierarchy

As in VLSI, an effective design methodology can best be supported by a multi-level design hierarchy. At least three levels of design hierarchy can be identified for the SFF processes.

1. Design Level, including functions, features, properties and inhomogeneities.

2. Three-dimensional Geometry Level, including solid geometry description in mathematically useful form. Call it SIF (Solid Interchange Format).

3. Two-and-half-dimensional Layered Level, adapted to the specific SFF technology being used. Call it L-SIF (Layered SIF).

At a higher level of abstraction, a formalism to capture and incorporate functionality from the shape and form of the parts will be desirable. Relevant mechanical and physical properties of the material must also be considered in the context of the intended application and performance of the part. The latter can be specified in the form of a set of design constraints derived by mathematical modeling and experimental studies including, but not limited to, simulation. A feature-based system can capture some aspects of functionality at a sufficiently high level; however, there is no agreement on which features should be included to cover present and future applications or how they should be described.

For mechanical design, formal systems that lead to effective synthesis algorithms do not yet exist. Indeed such formal systems may never exist since the VLSI synthesis algorithms depend on the Boolean properties of circuit elements that do not exist in the mechanical world. Computer-based synthesis in the mechanical world may be based on parameterizable building blocks in extensive libraries.

Once the design has been specified, the abstraction level that provides a completely process-independent representation is the level of geometry abstraction. The 3D geometry, or the shape of the object, can be expressed by a solid modeling system such as CSG (Constructive Solid Geometry) or B-rep (Boundary Representation) augmented by NURBS (Non-Uniform Rational B-Spline) for curved surfaces. To incorporate composite material structures, anisotropic material properties, color, surface roughness, support structures, tolerances and other physical properties, the model can have associated annotations. A set of high level generic design rules delimiting structural feature capabilities applicable to specific SFF processes such as limits of minimum and maximum sizes, surface finish, abruptness of transitions, *etc.* could be defined at this level. This is the level at which the digital interface separating the design from fabrication will reside (to be called SIF).

At the next level of abstraction, a slicer can take this description and convert it into a layered description (to be called L-SIF) suitable to be transmitted to a specific SFF process equipment. The level of abstraction at this stage is a 2.5D description of the object corresponding to a particular process. Although the representation language used at this stage is technology-dependent, the description must satisfy the fabrication rules of the process. The process planner for the SFF fabrication facility also specifies the orientation, support structures if necessary, as well as material composition information in the digital data used to drive the fabrication facility.

#### Finding

At least three levels of design hierarchy can be identified for SFF processes. There is a functional design level, where the shape, size and material structure of the object is derived based on application and performance considerations. There is a geometry description level in three dimensional space, for which a process-independent representation based on solid modeling techniques can be used (SIF). Finally there is a 2.5D process-dependent layered description level (L-SIF), which provides a process plan for the fabrication of the device using a specific SFF technology. The digital interface between design and fabrication resides between the three-dimensional object description (with appropriate annotations) and the layered structure.

#### Recommendations

Support research to develop formal design methodology at the functional level of mechanical design and develop software simulation tools to verify the design, taking into account the design constraints and performance specifications. Support both experimental and theoretical study to understand the design constraints and performance limits.

Support research to derive design rules for classes of SFF processes that share common characteristics. Support the evolution of standard SFF technologies and environments for rapid prototyping.

Support research to define a digital interface for mechanical design at the geometry level using a solid modeling language as the design interchange language.

## 2.4 Design Languages

What kind of language should be used to describe the objects to be produced by SFF processes? As noted earlier, for mechanical design, formal systems that lead to effective synthesis do not yet exist. A feature-based language can capture some aspects of the functionality, but there is no agreement on which features should form the basis of definition of such a language. The language definition must be based on an abstraction level that provides a completely process-independent representation. The level of geometry abstraction expressed by a solid modeling system such as CSG or B-rep augmented by NURBS for curved surfaces seems to be a reasonable level of abstraction. A 3D language, called SIF or Solid Interchange Format language, has been proposed by Sequin (See Section 4.4.1). To incorporate composite material structures, anisotropic material properties, surface roughness, tolerances and other physical properties common to all SFF processes, the language can have an associated content annotation facility that may be process-dependent. Such a language has greater capability than the current industry de facto standard of STL that describes the geometry by tessellating the surface into a mesh of non-overlapping triangular facets. STL has several drawbacks in handling large curvature surfaces and surfaces with fine features, which may lead to errors of unwanted holes or gaps or give rise to non-manifold models of parts. Similarly, IGES representations give rise to ambiguities, lack of robustness and translational problems.

We also need a language to describe the 2.5D layers to exchange process dependent information and software that will compile the normally process-dependent layered description of the object from the SIF representation. It is an interesting research issue to address the need to incorporate capabilities in the 2.5D language to support embedded structures.

Another geometric representation that has been presented at the workshop but not discussed in any great detail is the idea of "voxel" (volume element) based object data. Typically such data form the initial description of objects in biomedical applications such as craniofacial surgery or orthopedic surgery or building of artificial bone or tissue material. The structure of the object is represented by CT (Computer-based Tomography) or MRI (MagnetoResonance Imaging) data that describes the three-dimensional volume based on opacity or color values. For this kind of application, surface or volume generation techniques using ray-casting algorithms have to be used to create a model of the object. This description is then transformed into slice information for the SFF process for building the part.

### Finding

SFF designers need to use both the 3D representation of artifacts and the slice description. They often want to use feature descriptions, and they need to annotate the 3D representation with material composition and material property information. Languages at the functional design level or feature-based languages are not universally accepted for 3D modeling. A language based on solid model representation to describe the geometry and shape of objects might be used to define a process-independent digital interface between design and fabrication.

There is also the need to develop a 2.5D language that can be used to describe the object and to exchange process-dependent layered information to the process planner for specific SFF processes

### Recommendation

Support research aimed at the creation of two descriptive languages for SFF design:

1. A 3D language based on a solid modeling system such as CSG or B-rep, possibly augmented by NURBS to handle curved surfaces. Such a language, called SIF (Solid Interchange Format) should be developed as a standard for geometric data exchange. The language should make provision for material composition, property anisotropy, surface roughness, tolerances and other physical properties as annotated features. Initially SIF should not deal with feature-based descriptions until more is known about the set of features desired and the limitations on the use of SIF without such capability.

2. A 2.5D layer description language (L-SIF) based on slicing algorithms from the 3D description. Provision must be made for carrying the materials annotations through these transformations. Techniques are essential for ensuring that the two descriptions are identical. The slice description must be computer-testable against the design rules that define SFF process capability.

### Finding

Development of mechanical design systems for SFF is key to the success of the digital interface. On the design side, work must be supported to understand the nature of mechanical design and simulation tools to capture the design constraints and performance of the design. On the fabrication side, research is needed to develop process simulation, derivation of design rules and tools to capture microstructure and material properties. All of these capabilities will need to be exercised extensively through active use working with a MOSIS-like fabrication service that makes rapid prototyping available to mechanical designers through transmitted digital descriptions in a prescribed format.

### Recommendations

Define design tools and rapid prototyping environments for a selected set of SFF technologies and exercise them for rapid prototyping by designing and building artifacts. Develop test structures for benchmarking, testing process limitations, and defining and evaluating design rules for these chosen technologies.

Create a national infrastructure for rapid prototyping and create a scientific community of researchers, educators and industrialists who will share design tools, SFF fabrication capabilities, educational material and technical expertise.

# 3. Breakout Group Reports

## 3.1 SLA Breakout Session Report

**Participants**
Erik Antonsson (California Institute of Technology), Bernard Chern (National Science Foundation), Paul Fussell (Alcoa), Nick Mankovich (UCLA), Herbert Voelcker (Cornell), Steve Yencho (Stanford)

The SLA group began by addressing some basic questions about stereolithography and the independence of design and manufacturing. Among the questions discussed were:

Does Stereolithography lend itself to the separation of design and manufacture?

Can we create a taxonomy of process-specific key characteristics?

What are the intrinsic characteristics of this process?

What does this tell us about design side? How do you get design rules?

Do we need a process independent representation?

Is STL the appropriate representation? What about solid modeling?

Design features will trap you into a certain set of processes. In the VLSI case you have the advantage that the function can be achieved without a certain layout. The physical layout can be abstracted; this is difficult with mechanical products. With SLA, design feature abstraction does not map one-to-one onto process abstraction (*e.g.* sacrificial materials, anisotropy, specific geometries). We might have to consider geometry in new ways.

We could give up independence if it gives some advantages. Formalisms and design automation are impossible without giving up process independence. Are we willing to trade off a reduction in the design space for some level of formalism for design, where the intrinsic properties of the process (rules) are part of the design? Old style parts from 75 years ago have design plus rules ("drill and finish here"). This dependency was eliminated for a reason. Should we rethink this for manual versus automated fabrication? For example, Toyota uses rules for design that arise from known properties of metal (bend radius, weld strength, *etc.*). These process-dependent limitations serve as rules. In prototyping, the feedback of design and process integration is useful. However, many manufacturing processes rely on the separation.

Most CAD systems currently generate STL files; however, there are difficulties in the generation of these files. What are the limitations of the STL file? It is a "bag of triangles." You are going from a rich environment in CAD and throwing away a tremendous amount of information to do that. We should be able to do other things with the representation, *e.g.*, finite element analysis, *etc.* STL represents shape but not function. The designer should know about the fabrication process in order to avoid process-dependent problems. If we are going to take a structured VLSI-like design approach, then the designer should be aware of what can and can't be built.

There exists a proper theory of solid modeling that is adhered to by various systems. Solids are modeled as sets of rigid points. There is an algebraic geometry. Solidity means homogeneous solidity, with no notion of gradients. There is finiteness, finite describability. Boundary determinism makes SLA work. You don't need the inside when you have an exhaustive boundary. The whole idea of solid modeling relies on homogeneous materials.

However, some models, like those derived from medical images are innately heterogeneous. A strategic research topic may be the description of heterogeneous slices. We can handle multiple materials but complexity increases. Increasing the complexity of properties increases the burden of representing the functionality and makes difficult the calculations made upon that complex model.

### Summary

For SFF technologies it is worthwhile to look at the design methodology Description (recipe) to Shape (properties) to Function and visa versa. With SFF independence of design and manufacturing is not necessary. It offers the promise of being able to provide some stages of automation. Currently you must talk to the process engineer to tweak the process to get what the customer wants.

There is an advantage of introducing a new design methodology. It has the advantage of a clean separation (a real interface) without need for independence.

### Recommendations

1. Terminology

- Freeform Fabrication (FF) should be replaced with Freeform Accretion (FA) with the important subclass Laminar Freeform Accretion (LFA).
- The adjectives heterogeneous versus homogeneous should be reserved for use as qualifiers for the topological dimensionality of physical artifacts and their mathematical representations.
- The adjectives isotropic versus anisotropic should be reserved as qualifiers for the material properties of physical artifacts and their mathematical representations.

2. Modeling Systems: Many current CAD systems are capable of representing isotropic and piecewise isotropic c solids of dimensionality 1, 2, and 3, but are ill equipped to handle anisotropic solids.

- Surveys should be conducted to determine the character of the anisotropies that are (1) important or promising in known or projected applications of FA or LFA technologies and (2) currently or potentially attainable in known FA or LFA technologies.
- Schemes for representing anisotropic solids of dimensions 1, 2, and 3 should be investigated theoretically and experimentally. Promising candidates include, but are not limited to, spatial enumerations, cell decompositions, and ray representations.
- Means for building or extending CAD user interfaces to specify anisotropic material properties should be studied both conceptually and experimentally.

3. Structured design methods: All of these are likely to be process dependent.

- Develop design rules, that is, constraints on the design process to assure ability to fabricate, e.g., feature size has to be larger than lambda. Three things must come together in arriving at rules: specification language, digital interface, body of knowledge of agreements (implicit/explicit understandings) between fabricator and designer.
- Develop reusable libraries of shape elements (spirit of VLSI). These may be process dependent.
- Develop reusable device function libraries (*e.g.* VLSI adder) *e.g.* hinge. These may be

process dependent.

- Develop methods for combining shape and function elements into higher order elements, *e.g.*, combine rotational and structural elements into a mechanism.

## 3.2 Laser-Based SFF

**Participants**
Joseph Beaman (University of Texas at Austin), Richard Crawford, Chair, (University of Texas at Austin), James Glimm (State University of New York), Aravinda Kar (Phillips Laboratory), Amar Mukherjee (University of Central Florida), John Stivoric (Carnegie Mellon University), Prashant Kulkarni (University of Michigan)

This group was assigned to examine the key questions raised in the workshop charge with respect to specific laser-based solid freeform fabrication (SFF) technologies, namely, Selective Laser Sintering (SLS) and Laser Aided Direct Rapid Prototyping (LADRP). In particular, the group focused on questions of the development of structured design methodology, abstraction levels, design languages, digital interface, and simulation models for SLS and LADRP. Ideally, a key feature of those SFF processes amenable to VLSI-like layering paradigm is that the processing steps are insensitive to the object's geometry. This implies that the process planning is done once and all objects are handled in a uniform fashion. Both SLS and LADRP can exhibit this characteristic for classes of parts when the layer thickness is chosen to be uniform, the material is homogeneous, and the object geometry is decomposable into layered slices. There may be parts that require additional manufacturing operations calling for additional process planning. However, both these technologies offer possibilities for novel manufacturing applications that can be exploited if we use a full 3D geometric model that incorporates attributes such as non-homogeneous material distribution, spatially distributed microstructures and physical properties, and non-uniform layer thickness.

### 3.2.1 Design Methodology for Laser-Based SFF

In order to determine the requirements for design methodology for laser-based freeform fabrication technologies that include levels of abstraction, languages for data exchange standard, design rules and digital interface, the group first discussed the nature of the design process for SFF. The group envisions a computer-aided design and fabrication system consisting of two major subsystems: Design and Fabrication separated by a digital interface. The fabrication subsystem may need an additional Process Planning stage in the general case, as shown in Figure 1. As discussed earlier, in cases where the layering paradigm holds true, the process planning stage will be unnecessary.

The Design subsystem performs the design for function; that is, it analytically derives the shape and form of the part that achieves the desired function and performance specification. The design subsystem could be essentially identical to mechanical design as currently practiced, but there is a need to develop formal design methodology at this level of abstraction, which is comparable to behavioral and functional level design for VLSI. This subsystem has its own data requirements based on the types of analyses and simulations that are used to determine part shape based on functional requirements and design constraints relating relevant mechanical and physical properties of the material. This module is impacted by SFF in that some capabilities allowed by SFF (*e.g.*, non-homogeneous material distributions) are difficult to account for by present analytic capabilities, and new tools that permit us to take advantage of these capabilities will need to be created. The analysis tools can also incorporate such capabilities as determining the proper build orientation to achieve preferred physical property directions or to minimize support structure volume. The design may go through multiple refinements and verifications before being delivered via the *digital interface* to the Process Planning subsystem, as shown in Figure 1. The output of the design subsystem is a 3D description of the geometry that includes attributes such as material distribution, tolerances, surface finish requirements, *etc.*

The Process Planning subsystem uses knowledge of the capabilities of the particular SFF technology, for instance, to slice the part geometry adaptively. The output of the Process Planning subsystem is a 2.5D layered representation augmented with information such as slice thickness, tolerances, etc. The Process Planning module contains the so-called *Design Rules* that ensure manufacturability of the part. These rules are analogous to the design rules in VLSI fabrication, and might be more properly termed *Fabrication Rules*. The commonly used interpretation of design rules in mechanical design are design constraints that ensure functionality.

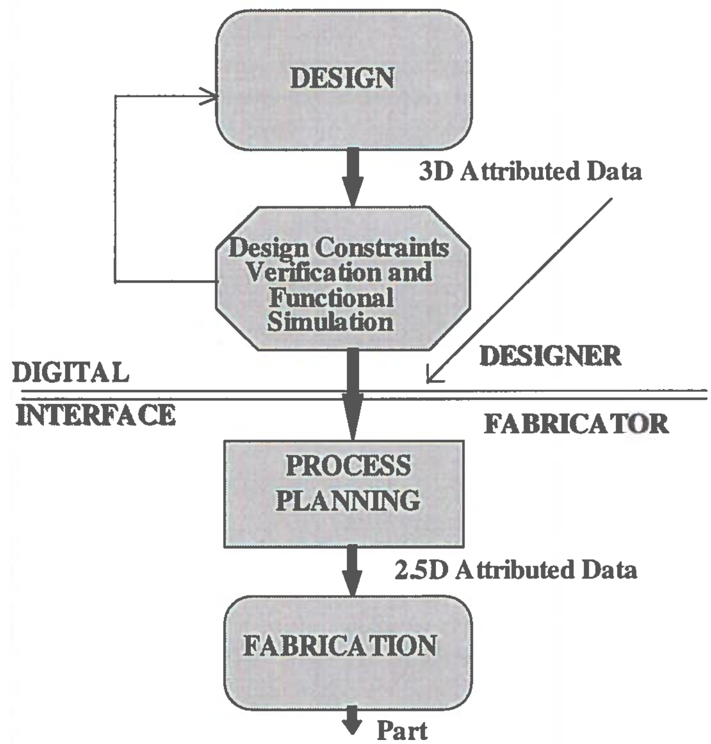



Figure 1. Design Process for SF

The third subsystem is the fabrication module, which resides with the SFF equipment. This module interprets the sliced 2.5D attributed data and controls the fabrication process to produce the desired part.

### 3.2.2 Design Languages for Data Exchange

What kind of language or geometry editor should be used to describe the object to be produced by layer-based manufacturing? This will depend on the levels of abstraction of the design process. The SFF industry has a *de facto* standard, the STL file format, for exchanging geometric information. The STL format describes the geometry of a part by tessellating the surface into a mesh of non-overlapping triangular facets. The format has been adopted by many CAD vendors and is readily available. However, problems with STL arise because it represents a first order tessellated approximation of complex three dimensional geometry. STL is inefficient and inaccurate in representing large curvature surfaces and surfaces that require fine resolution. Operations that require higher order information (*e.g.*, curvature), such as adaptive slicing, are not supported by the STL format.

The group agreed that the abstraction level to provide a more robust process-independent representation is the level expressed by a solid modeling system. The abstraction hierarchy for mechanical design is not as evident as it is for VLSI design. A feature-based language captures some aspects of functionality at a high level of design, but as current efforts to standardize features illustrate, there is no agreement which features will form the basis of such a language and ensure coverage of all present and future applications. The data exchange standard should probably be based on current boundary representation (B-rep) schemes, possibly augmented by non-uniform rational B-spline (NURBS) surfaces to represent free-form surfaces. Such a system should form the basis of a geometry editor and a 3D language, possibly called SIF (Solid Interchange Format, following Sequin's terminology), and should be developed as a standard for geometric data exchange. The basis for this language might be provided by the ACIS Save file format, since ACIS has been incorporated as the geometry engine for many commercially available geometric modeling systems. To incorporate material distribution, anisotropy, surface roughness, tolerances, support structures, and other physical properties and user-defined requirements, the language must have additional annotation capabilities.

Another language, L-SIF (Layered Solid Interchange Format), can be developed to describe 2.5D layers. Based on this standard, a slicer can be developed to generate the layered description from the 3D description. Other translators might be developed to perform certain transformations on layer-based data such as that obtained from laser digitizing measurements, computed tomography (CT), and magnetic resonance imaging (MRI). The layer thickness of the data from these sources are generally not compatible with SFF technologies, requiring some processing to enable dimensionally accurate manufacture of these parts by SFF and to make them compatible with the layer thickness of the specific fabrication process. Also, CT or MRI image data is volumetric data representing, for instance, tissue density. For layer-based manufacture, the volumetric information must be transformed into a contour representation of the layer. Translators for each expected type of input data can be constructed, all with the same 2.5D layer output format.

### 3.2.3 Design Rules

VLSI design rules are unique in the sense that they provided the designer a clean interface on top of which a structured design methodology can be developed. The operation of a VLSI device is abstracted hierarchically by formal systems that allow separation between design and fabrication. From the designer's point of view, the design rules provide assurance of manufacturability of working parts. As is well known to VLSI designers, this ideal hierarchical model can break down if we bring performance considerations (such as chip area, speed, timing, loading and power) into the design. In this case, design completion may need multiple iterations. The basic elements of design are customized, creating side effects that affect all the layers in the hierarchy. In fact, exact behavior of the device can only be described by an analog model, but for almost all VLSI applications an approximate digital model is more than sufficient.

For mechanical design, the performance considerations are an integral part of the design process. For a mechanical part, a solid model representation describes its shape, but the desired functional behavior cannot be assured unless the material satisfies certain physical properties and dimensional constraints imposed by the nature of application and its performance requirements. The limits on geometric dimensions of the object as they relate to correct function and performance of the object will be called *Design Constraints*. These design constraints are process and materials independent and can be derived by experimentation and mathematical modeling. We adopt this term to distinguish these functional considerations from considerations of manufacturability. In the context of VLSI, the phrase "design rules" has been interpreted to mean conservative geometric fabrication rules. Within the context of mechanical design and SFF, an analogous set of process dependent *Design Rules* must also be satisfied. This leads to an abstract tripartite model of design and fabrication as shown in Figure 2. The group discussed and formulated a possible set of design rules and design constraints applicable to both SLS and LADRP processes. These are enumerated below:

**Design Rules**

1. Minimum feature size ( the equivalent of l for VLSI)
2. Minimum/Maximum dimension
3. Minimum surface roughness
4. Density limit
5. Minimum/Maximum layer thickness
6. Dimensional tolerance

For the SLS process, there may be a few additional design rules:

7. Powder removal limitation
8. Multiple fabrication capability
9 Support structure limitation

**Design Constraints**

1. Material limitation
2. Strength limits (anisotropy)
3. Production rate
4. Thermophysical property limits
5. Aging stability
6. Post-processing requirements

The above set of design rules illustrates the types of design rules that must be defined precisely by each fabricator for each process or a class of processes. The layered description of the object in the 2.5D language consistent with the design rules will then provide an interface between the process planning stage and the fabrication stage.

By analogy to VLSI design, the digital interface should contain process-specific design rules, the information necessary for the designer to use the process and specific information that may be used to generate additional process planning steps. The process planning stage will verify whether or not the specified design can be translated into a sequence of process steps consistent with the design rules. For VLSI design, a set of standard and generic process steps evolved, and each of the mask layers precisely defined a process step such as oxidation, ion implantation, diffusion, metallization, *etc.* The laser-based SFF technologies are in a process of continued change and evolution. A common digital interface definition is practical, at this stage, only for a limited subset of SFF processes.

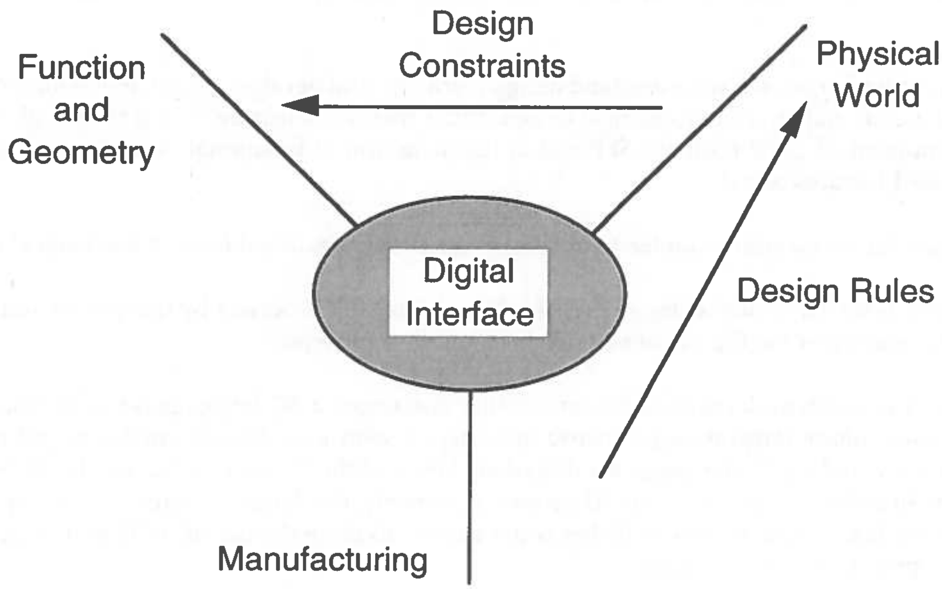



Figure 2. Abstract Model of Manufacturing

### 3.2.4 Findings and Recommendations

We conclude this section by summarizing our main findings and recommendations.

Findings

An automated design system for LSL or LADRP processes consists in general of three major subsystems: Design, Process Planning and Fabrication. The design is to be done using a 3D geometric modeler in conjunction with a performance modeler that verifies the function of the part taking into account the design constraints. The process planner is responsible for translating the 3D description into realizable 2.5D layered description satisfying the design rules of the fabrication process. The fabrication subsystem creates the part specified by the process planner.

At least three levels of design hierarchy are required for the laser-based SFF processes. They are: a functional design level, where the shape, size and material structure of the object is derived based on application and performance considerations; a geometry description level in three dimensional space, where a process-independent representation based on solid modeling techniques can be used; and a 2.5D process-dependent layered description level, which provides a process plan for the fabrication of the device using a specific SFF technology.

The group defined an abstract tripartite model of design and fabrication applicable to LSL and LADRP processes and stressed the importance of recognizing the difference between design constraints and design rules in the context of the definition of a digital interface. The design constraints define the limits of physical properties and mechanical dimensions in relation to desired functionality and performance of the part and can be derived by experimentation and mathematical modeling independent of fabrication process specifics. The design rules give conservative fabrication criteria that assure manufacturability. The group identified a set of design rules and design constraints applicable to both SLS and LADRP processes and a few additional constraints applicable to LSL only.

The group ended with a discussion of the needs of the SFF community's research with respect to the workshop charge. These needs are summarized below in a list of recommendations.

Recommendations

Support basic research to understand design hierarchy and develop formal design methodology at the functional level of mechanical design. What role will a feature -based system play in the development of CAD tools for SFF and in the definition of functional modules or libraries of standard features/parts?

Support basic research to understand constraints at the functional level of mechanical design.

Support research to derive design rules for laser-based SFF processes by theoretical and experimental studies or via the use of suitable benchmark prototypes.

Support research to develop design interchange languages: a 3D language for solid modeling at the design phase integrating geometric information with user defined attributes and material properties, and a 2.5D language for describing layers of the 3D object obtained by slicing algorithm. In order to reproduce the 3D geometry correctly, this language must also be capable of handling both linear as well as higher order curves. Explore the use of ACIS as a basis for the development of these languages.

Support research to investigate the use of the laser as a general manufacturing tool and to integrate SFF processes with other manufacturing uses of the laser. Can SFF be integrated with other manufacturing processes/technologies? How will this development impact the design languages for future manufacturing?

Support research to explore the feasibility of manufacturing facilities for laser-based processes.

## 3.3 SDM Shape Deposition Manufacturing

**Participants**

Susan Finger (Carnegie Mellon University), Tom Peters, Chair (University of Connecticut), Bob Sproull (Sun Microsystems Laboratories), Gerry Sussman (Massachusetts Institute of Technology), Lee Weiss (Carnegie Mellon University)

This section sketches a structure that allows a designer to control the fabrication of mechanical parts using all-digital interfaces between designer and fabricator. We suggest a set of initial interfaces for experimentation. Our scheme is heavily influenced by Shape Deposition Manufacturing (SDM) methods, which use a layering technique to obtain near-net shape, intermingled with machining and other steps that refine the part's shape and surface properties. We believe our scheme applies to both layered and to certain forms of machining fabrication.

Note: Throughout this section we use the word *part* for the physical object to be built, even though it may not be a part in the sense usually meant by designers. We use the word *fabricate* for the process of constructing a part from a digital description of the part.

Our proposal is not an attempt to solve all known problems in specification of mechanical parts (*e.g.*, specification of tolerances). Rather, it suggests a modest way to get started so that we can make progress and begin to uncover new problems.

### 3.3.1 Designer and Fabricator

A fabricator publishes a *Designer's Guide* for using his fabrication process. It will list available materials, describe design rules and establish other requirements for design. A designer can obtain this file from each fabricator in electronic form.

A designer then uses an existing CAD tool to design a part. The geometry from the CAD tool is converted into a SIF file, using a conversion program that reads either a proprietary format associated with the CAD tool or a standard form like IGES. (Note: Unlike CIF, we cannot assume that a SIF experiment will stimulate a new set of CAD tools that directly write SIF. Geometric modelers and CAD tools are hard to write, so we must focus on using existing tools.) However, we insist on distinguishing SIF from existing geometry standards because the information required by a fabricator is more than just geometry; moreover, we want to have an interchange format that we can use for experiments.

The SIF file is a text file that contains four kinds of information:

1. Geometry, in a flexible form described further below;

2. Annotations, information such as material and surface properties that are linked to the geometry;

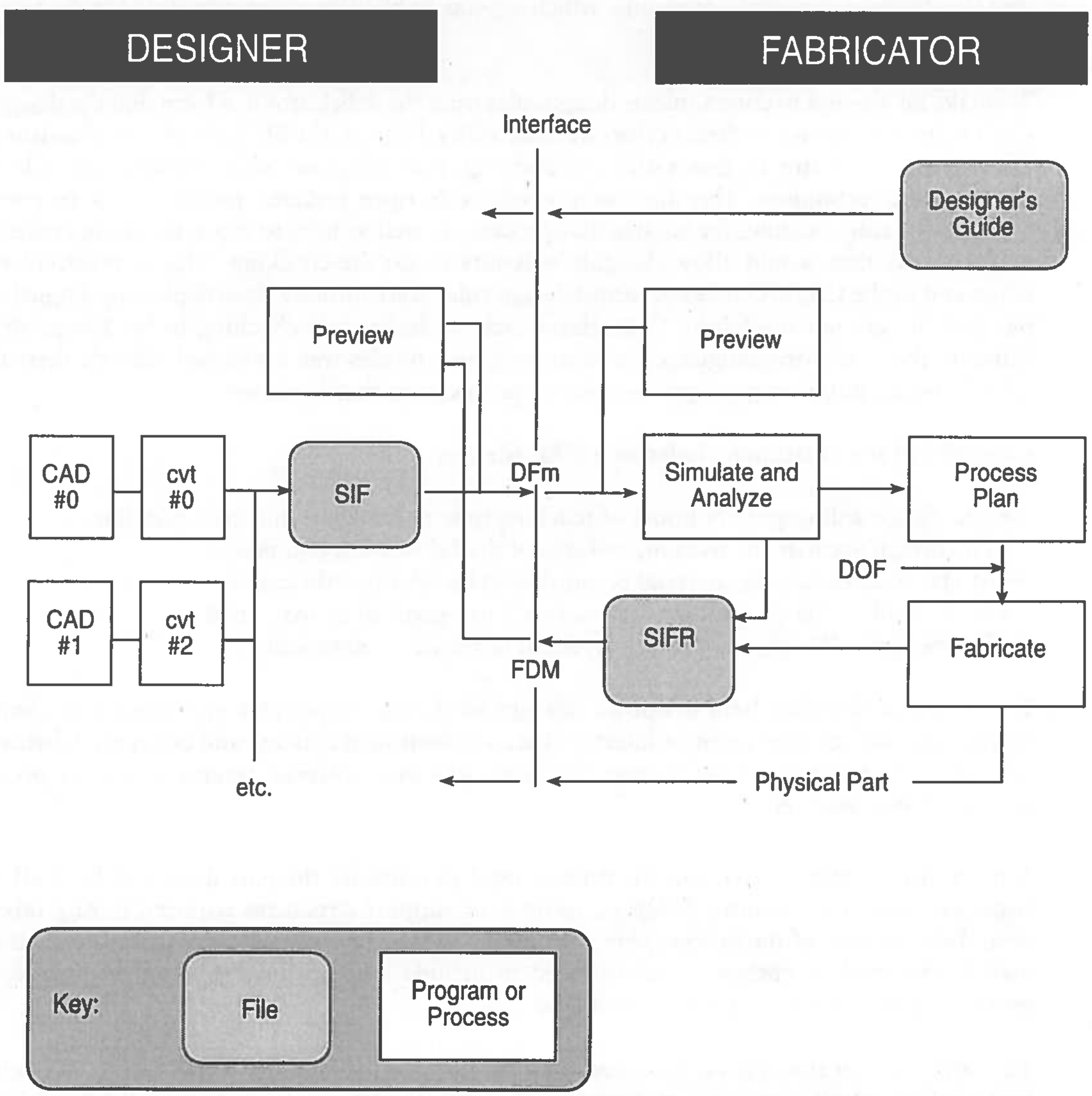



Figure 3: The roles of designers and fabricators

3. Hints, optional directives that may simplify the job of the fabricator; and

4. Integrity check, a check sum to verify that the data in the SIF file is not corrupted during storage or transmission.

A designer and fabricator may interact by exchanging electronic mail messages. A designer-to-fabricator (DF) message may contain a SIF file, but it must also contain "transaction information" that tells the fabricator what to do, *e.g.*, analyze and check the SIF file, build N parts, *etc.* These DF messages are patterned after a similar message format used by the MOSIS facility [2]. Although MOSIS uses custom software to interpret and process these messages automatically, we propose restraining early uses of SIF to a low frequency that favors processing DF messages by hand. If demand builds, we can build the necessary software automation or adapt the MOSIS software.

The fabricator's digital response to a DF message is a fabricator-to-designer (FD) message, which provides feedback to the designer. The FD message contains transaction information, again, patterned after MOSIS. In many cases, the FD will also contain a SIF response file (SIFR) that reports the fabricator's analysis of the design back to the designer. In particular, this file identifies specific design rule errors. Some errors are attached to $(x,y,z)$ coordinates of the design so that the designer can easily determine which aspects of his design cause trouble for the fabricator.

We make no attempt to communicate design rules from the fabricator in a form that the designer's CAD software can use to detect errors automatically. Instead, the SIFR file is a mechanism that allows each fabricator to detect and report design rule violations using process-dependent or idiosyncratic techniques. This approach leaves as an open research problem how to provide good design rule checking for a particular process, as well as how to represent design rules in a uniform way that would allow designer software to do the checking. This is precisely what happened in the CIF/MOSIS experience: design rules were initially described using English and pictures. It was not until John Ousterhout tackled design-rule checking in his Magic design software that a uniform language for expressing design rules was developed. Shortly thereafter, MOSIS began publishing design rules for its processes in that language.

Examples of information included in a SIFR file are:

- The design will require N hours of machine time to fabricate and cost X dollars.
- The design exceeds the working volume of the fabrication equipment.
- At (10, 5.1, 10.32) the material is too thin to be reliably fabricated.
- At (5, 3.44, 0.12) the radius of curvature is too small to be machined.
- The design will require 3456723 layers to fabricate — impractical!

The proposal sketched here is not an attempt to design a universal interface to mechanical fabrication. We are after a simple interface that a collection of existing and emerging fabrication processes can fruitfully share. It may lead some day to a universal interface, but we propose only a modest first step.

A fabricator is expected to plan the process used to fabricate the part described by a SIF file. Some processes may require designing fixtures or support structures required during fabrication; these are part of the process plan. The planning may be automatic or manual intervention may be required. A designer is encouraged to include hints in his SFF file that simplify the process planning or even make it automatic.

The remainder of this section provides more details of our sketch for a SIF design. We refrain from making specific syntactic and semantic specifications because we believe they are premature. Until there is some agreement on the overall approach to digital interfaces to fabrication, it is pointless to become too specific.

### 3.3.2 SIF

The principal role of SIF is to provide an unambiguous description of the geometry of the part to be fabricated. But there are several pragmatic concerns that influence SIF as well. First, we must be sure that several different popular CAD systems can generate, using a converter, a usable SIF file. Second, we want to make SIF easy to use by fabricators—to encode part specifications in a form that a fabricator can easily convert into detailed device-control instructions to make the part. These considerations suggest the following approach to specifying geometry:

1. Initially, geometry can be specified using a conventional B-rep, consisting of faces, edges and vertices. This representation should accommodate smooth curves and surfaces, not just linear edges and planar surfaces. This representation should be extensible to spatial enumerations, cell decompositions, *etc.* Some possibilities for such extensibility are discussed in Subsections 3.3.3.1 and 3.3.3.2, below.

2. A constructive approach to specifying geometry can convey valuable hints to fabrication processes. For example, a hole that can be drilled with a machine tool is readily detected in a constructive geometry specification, but quite difficult to detect in a B-rep. By "constructive" we mean not just boolean operations on simple shapes, but constructors such as surfaces of revolution.

3. The first two observations lead to a third: that it is advantageous to embed geometry specifications in a simple language that is not only editable, but can express data dependencies (*e.g.*, a surface is generated by revolving a curve, which is itself generated by another technique), parametric relationships and the like. This language could be likened to Postscript, an interpreted language that sits on top of a raster graphics library: we desire an interpreted language atop a solid modeling library. Of course, a SIF file may make only trivial use of the language, and represent literal geometry using only coordinates and a few structuring concepts to identify geometric elements.

4. Finally, SIF must be designed so that software tools can be easily constructed to deal with SIF files. This means that the "language" used in SIF should be a simple language that is already defined and that already has good public-domain interpreters (*e.g.*, Scheme, Tcl). Moreover, we could choose the constructs of the solid modeling library to match those of a package available to and selected by the SFF research community, *e.g.*, ACIS [3].

### 3.3.3 SIF Solid Geometry Sublanguage (GSL)

Our SIF proposal uses specifications of well-formed solids in several different contexts. We propose that the same solid geometry specification form be used in all such cases. The specification includes:

- Boundary representations, with geometry and topology clearly distinguished and with the ability to represent higher-order shapes (*e.g.,* NURBS).

- Construction operators, including boolean operations, volumes of revolution and extrusion operations that correspond to many part features, such as holes, pockets and fillets. Most of these can be represented easily by boolean operations on suitable solids. Some may benefit from specialized ways to construct a suitable solid (*e.g.*, fillets).

- Ways to name surfaces and solids to serve as anchors for annotations.

SIF geometry may be faceted, but need not be. The intent is to encourage designers to retain curves and surfaces as higher-order elements in the SIF geometry; the fabricator can facet these if necessary for his processing.

The GSL is essentially a scripting language that calls an underlying geometric modeler to construct a model of the part. Some GSL specifications may be entirely literal, *e.g.*, constructing a model from faces, edges and vertices. Others may make heavy use of constructive operators or numerical calculations in the script used to express a parametric design. The constructs of GSL can be designed to provide hints to process planning.

The Erep design [4] is an example of a way to design the GSL. As Hoffman notes, many modern geometric modelers retain a history of modeling commands that can be converted into a script, such as our envisioned GSL. Although there is no standard scripting language defined for this purpose, it might be feasible to design one that could be the target of a conversion from a modeler's proprietary form. Henderson [5] shows an extension to the ACIS command language to make a scripting language for modeling. One form of the Alpha_1 modeler is a modeling package called from R-LISP; the permanent representation of the design is the R-LISP program that is executed to build the model.

#### 3.3.3.1 Extensibility: Richer Geometric Domains

While mindful of our previously stated intent to propose "... a simple interface ... [as] only a modest first step," we expect that this simple interface would evolve over time. In keeping with good design practice, we consider graceful extensibility. In particular, this subsection notes some known limitations of existing solid modeling paradigms, in anticipation of the need for broader modeling approaches.

Novel topologies, embedded materials, anisotropic materials, *etc.* will not be adequately represented by current solid modelers. However, much of the supporting research for extending the solid modeling domain has already been completed. Such extensions include non-manifold topology (NMT) [6], spatial enumerations, cell decompositions, ray representations and real-function representations (See Voelker's abstract in this volume).

Existing SFF practice has already utilized NMT modelers. The Next Generation Geometry (NGG) remains unknown, but we can envisage extensibility to the leading current candidates to make sure that what we do now for solids will not hinder future extensions. For instance, one may wish to have the geometry accompanied by a separate companion topology representation. The contemporary solids topology would have restricted possibilities for the face, edge, vertex interrelationships that could be verified by simplistic rule checkers. Next generation designs might have more complex topology and very different rule checkers. We should anticipate this geometry/topology split now to be sure these data structures are sufficiently independent for later extensions. All geometry eventually would be integrated in the SIF.

#### 3.3.3.2 Extensibility: Separation of Geometry and Topology

Extensibility of the geometric domain (as discussed in the previous subsection) suggests an increasing recognition of the need for complementary, yet independent, perspectives upon geometry and topology.

In any interchange format, the geometry and topology information should be specified in distinct, separable abstract data structures. Of course, there is great interaction between these two types of information, but they represent distinctly different design perspectives. Much of the early historical difficulties in developing robust, reliable, solid modeling algorithms was because these notions were merged. It took Weiler's Ph.D. thesis [6] to educate the research community about the need for such modularity. As a closely related development, a recent algorithm on repair of .STL files concentrates upon fixing topology issues, but does so by accessing both topology and geometry information [7].

Let us not repeat past mistakes by merging geometry and topology. Perhaps, we should also be seeking a Topology Interchange Format (TIF).

### 3.3.4 SIF Elements

In this section, we discuss the elements of a SIF file.

1. **Ideal part shape geometry.** The ideal shape of the part is specified using GSL.

2. **Auxiliary geometries.** This section of a SIF file specifies GSL geometries for objects other than the finished part. Names associated with this geometry are used in annotations to specify things like material properties, *etc.* The reason for specifying auxiliary geometries separately from the final shape is that separate techniques may be used to generate them (*e.g.*, the ideal shape may come from a CAD system, while the auxiliary geometries might be entered using a text editor).

3. **Annotations.** Annotations are a way to link non-geometric information to the description of the ideal part shape, by way of a name attached to a surface, volume or intermediate construct in the ideal shape specification. Below is a list of annotations that would be a good starter set. The name of each annotation is a hierarchical name so that the annotations are easily extensible:

   surface/roughness <geom-name> <roughness-specification>

   Although we are uncertain of the precise method to specify surface characteristics, the intent is that an SFF fabrication technique will choose a layer thickness based in part on this parameter.

   surface/deviation <geom-name> <distance>

   Rather than tackling all the problems of tolerances, we let the designer indicate a distance by which the part is allowed to differ from its ideal shape. This parameter may govern the precision of a faceting approximation used by the fabricator in geometry processing, or the layer thickness in an SFF process, *etc.*

   volume/material <geom-name> <material-name>

   We associate with a volume the name of the material used to fabricate that volume. If the name corresponds to auxiliary geometry, then the material will be used wherever the auxiliary shape and the final shape coincide. The names of materials available from a fabricator are reported in the *Designer's Guide*. Names are hierarchical in order to allow a fabricator to select close approximations if possible. For example, if a design calls for "aluminum/531" a fabricator might substitute "aluminum/531/peened" or even "aluminum."

   Pieces that can be embedded within a structure are specified as materials, using the root word "embedded." Thus we might find "embedded/IC/SN7474." For the time being, each fabricator will have to establish his own conventions for coordinate systems used to locate and orient embedded parts; this will be covered in the *Designer's Guide*.

   volume/material-function <geom-name> <SIF language procedure>

   This annotation provides a way to compute a material name or mixture based on an (x,y,z) coordinate within a volume.

surface/material <geom-name> <material-name>.
A way to specify surface finish, *e.g.*, painting or anodizing. Again, the names of finishes available are published by the fabricator.

surface/pattern <geom-name> <pattern-name>
Specifies a particular surface treatment from a catalog of treatments published by the fabricator. This is intended to support surface textures of the form that *e.g.*, 3D printing can make; however, it is not clear that this will provide enough information or control, *e.g.*, for phase.

4. **Hints.** This part of the SIF file encodes hints the designer passes to the fabricator. These may include device-dependent notions and are deliberately designed to be open-ended. The *Designer's Guide* should describe hints that are useful to the fabricator. Some examples are:

build-direction *x y z*
A vector in the direction in which layers are built up, normal to the plane of the layers. This hint can vastly simplify the process planning for many SFF processes. It might also suggest what surface to align to the bed of a milling machine for a part that is to be machined. This is only a hint: the process plan may ignore it.

plan-order <geom-name> <geom-name> ....
Suggests an order in which features can be constructed that may help a process planner, especially if some or all of the features are to be machined.

support <geom-name>
Specifies that a piece of auxiliary geometry can be used as a support structure, *i.e.*, fabricated along with the part, and separated mechanically later on.

hold-down-hole <geom-name>
Name of a hole that can be used in conjunction with a simple threaded fixture to hold the work. The intent is that initial stock could be clamped, and the hold-down holes drilled. Then the work could be held down with bolts through these holes, and machining could proceed.

DD/...
A device-dependent hint. The rest of the hierarchical name specifies the name of the process (given in the *Designer's Guide*) and then a device-dependent name. Thus "DD/SDM-CMU-1.1/arc-voltage" would be a hint about the voltage for the Shape Deposition process at CMU.

### 3.3.5 SIFR Elements

The SIFR file is designed to be machine readable to encourage designers to construct tools that will process the file automatically. The simplest such tool is doubtless a viewer, which allows the designer to inspect his design and see "post-it-notes" attached to the design wherever the fabricator has flagged critical conditions or design rule violations.

A SIFR file is a sequence of design comments. Each comment in the SIFR file is named with a hierarchical name. As with hint names, if a name begins with "DD/" it is a device-dependent comment. After each name comes an optional coordinate triple $(x,y,z)$ if the comment is associated with a particular spot on the part. Finally, there is a printable text string that describes the comment. This string will often be redundant, *i.e.*, it can be uniquely determined from the name of the comment. But sometimes the string will contain additional helpful information; examples are shown below.

Here is a starter-set of SIFR comments. The "cant" comments mean that the fabricator cannot build the part.

cant/other
: Cannot build the part. Text string gives details.

cant/working-volume
: The design exceeds the working volume of the fabricator.

cant/material
: Cannot find requested materials or plausible substitutes.

cant/cost
: Too costly in time or money to build this part.

cant/support (x,y,z)
: Material at this spot cannot be supported by the build process.

too-thin-material *(x,y,z)*
: Material will be too thin to reliably fabricate.

too-thin-gap *(x,y,z)*
: Gap between two surfaces will be too thin to fabricate.

surface-property *(x,y,z)*
: Cannot achieve the desired surface property.

volume-property *(x,y,z)*
: Cannot achieve the desired volume property.

material-substitute *(x,y,z)*
: Must substitute a requested material; text string tells what is chosen.

build-vector *(dx,dy,dz)*
: Gives direction of build vector relative to (0,0,0).

Note that we could augment the role of the SIFR file to include estimates of the results of the fabrication, *e.g.*, the strength of material or the surface finish at a given point.

### 3.3.6 DFM and FDM Elements

A message from designer to fabricator (DFM) is a structured electronic mail message, patterned after the MOSIS design [2]. Examples of fields are given below. Do not be daunted by this list; the intent is to keep this simple so that it can be operated by hand if the volume of traffic does not warrant building special purpose software.

DF message fields:
- Account number: assigned to each user by the fabricator
- Job number: identifies all DFM and FDM traffic associated with this job ID: uniquely identifies a DFM request, which will be echoed in an FDM response
- Command: specifies what to do, *e.g.*,
  - Store SIF — associate the SIF file in this message with the job
  - Build — build it, using the SIF associated with the job
  - Check — analyze the associate SIF and return an FDM containing SIFR
  - Bid — return cost estimate in FDM
- When: optional completion date
- Cost: optional not-to-exceed amount
- Quantity: how many to build
- SIF-file: followed by the SIF file itself

FD message fields:

Account number: as above
Job number: as above
ID: ID of requesting DFM
Status: current state of job, e.g.,
no-SIF — fabricator has not received legal SIF
not-Checked — SIF has not been analyzed by fabricator
ready — fabricator thinks design is ready to fab
in-queue — queued for fabrication
in-fab — being fabricated
Time-estimate: estimate of when part will be fabricated Cost-estimate: obvious
SIFR-file: followed by the SIFR file itself

### 3.3.7 Device-Dependent Format (DDF)

Because SIFR is a device-independent specification, it will not be able to exploit every feature of a fabrication device or method. In order to let savvy designers and others experiment and test the limits of a novel fabrication method, we propose that a Device Dependent Format (DDF) be accepted by each fabricator. A fabricator would publish, in the *Designer's Guide*, a description of its file format. A DDF file could be embedded in DFM traffic just like a SIFR file, or it could be transmitted to the fabricator in some other way.

By way of example, a pure layered process such as SLA might devise a DDF that is a series of two-dimensional patterns to lay down successively. Each layer might be defined as a geometric outline or a raster scan (perhaps run-length encoded). In Figure 3, the DDF file is shown as the output of the process planner and input to the fabrication process. However, fabricators might use the DDF in different ways, *e.g.*, subjecting it to a reduced process-planning step or checking for machine or operator safety violations.

The DDF shifts the burden of process planning and geometric transformations to designers. They may find few or no tools for preparing or manipulating DDF files, though each fabricator could create such tools and make them available to designers.

### 3.3.8 Software Tools

For a SIF design to succeed, it must be supported with a suite of robust software tools. In the case of CIF, these tools were developed by universities over time. This approach will not work for SIF because the tools are more complex and many of the research groups are focused on developing fabrication processes rather than design software. We foresee the need to commission the development of a suite of SIF-related software that the entire community can use.

The following list gives some of the software components required:

1. Converters from popular CAD file formats into GSL, the geometric component of SIFR.

2. SIF language interpreter, including an interpreter for whatever scripting language is used within SIF.

3. Solid modeling library, corresponding to the constructs of the GSL, called by the scripting language interpreter. If the solid modeling library is based on ACIS, we will still need the glue between the GSL interpreter and the ACIS APIs.

4. SIF/SIFR preview program, which displays geometry specified by a SIF file, possibly with comments specified with a SIFR file.

5. Scripts or programs that perform useful conversions, *e.g.*, from SFF files to STL files.

Findings

In support of SFF we see the need for a standardized interchange format, as a design description of an interface between modeling systems and SFF systems. We have proposed an initial design for such an interface as our SIF. We see the need for the concurrent development of associated software tools to create, edit and parse the information transmitted across such an interface. As a subsequent, and substantially more complex, undertaking, we see the need for development of software design/manufacturing advisors, to serve to accelerate the convergence of design/manufacturing iterations.

Recommendation

Design and implement the SIF and SIFR.

Pursue the design and prototyping of supporting software tools enumerated in Section 3.3.8.

Investigate design/manufacturing advisors.

As the goals of this program will rely heaving upon industrial participation, a secondary objective should be to investigate how to pursue effective collaborative efforts between the research and industrial communities. This is more an organizational and administrative issue, but one that is crucial to the accomplishment of the stated objectives.

## 3.4 3D Printing

**Participants**

Ely Sachs, Chair, (Massachusetts Institute of Technology), Earlin Lutz (Bentley Systems), Fritz Prinz (Stanford University), Rich Riesenfeld (University of Utah), Carlo Sequin (University of California at Berkeley), Ram Sriram (NIST), Chris Kasabach (Carnegie Mellon University), John Staudhammer (NSF - Scribe)

The particular issues brought to light by examining the specific SFF process of 3DP are:

- Need to be able to describe local material composition.

- Need to be able to describe local geometries such as surface textures. In the limiting case, these surface textures may be close to the limiting resolution of the process and hence may benefit from a voxel representation.

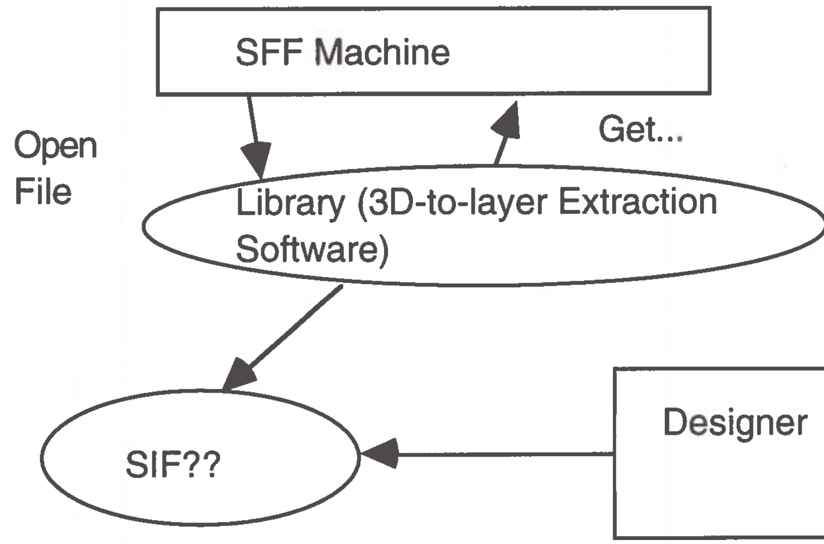
### 3.4.1 Procedural Model

1. Every designer chooses his/her own CAD package.

2. Designer can either transmit any of the following:
   a. export format of CAD files,
   b. simple format (3D), which is data intensive (*e.g.* STL), or,
   c. define SIF

3. Fabricator will do one of the following:
   a. take export format and will convert to machine specific format or
   b. Use simple format directly or
   c. Use library to read SIF format. Each fabricator will need their own library interface

One possible scenario is: A designer sends design information in SIF format. Fabricators have a library interface, which they use to query the data that they require for their domain. The library has interface to SIF format data files.




An open question in this approach is whether or not the designer will be able to aid in process planning decisions should it be desired. For example, the designer may want to participate in the economic versus quality tradeoff inherent in picking layer thickness. In this case, they would need to access a simulator, which may run locally or remotely. This simulator could also accept SIF.

### 3.4.2 Interchange Format

SIF is a 3D interchange format. 3D Format is deemed most useful for capturing design with greatest process generality. A 2D LSIF format that is process specific may or may not also be useful.

An open question is how the designer would deal with voxel information in cases where design was very close to limiting process resolution. Including voxels in 3D SIF leads to process specific SIF, which is undesirable. A 2D LSIF would logically include a voxel description capability.

Here are the requirements for SIF

1. Economical to transmit.

2. Captures
   Gross object geometry + topology
   Microgeometry (*e.g.*, textures)
   Local material composition
   Local material properties
   Surface properties.

3. Resolution of information should be greater than resolution of SFF (fabrication perspective).

Here is a proposal for SIF to meet the above requirements. The key idea is to capture the design (geometry) in its ideal, process-independent state.

Vertices (*x*, *y*, *z* and other properties)
Faces (Surface properties)
Compacts (Volume properties)
NURBS
definitions + instancing (Compact, ASCII transmittable)

B-reps and CSG + Surface (capture geometric intent)

Current CIF translators always interpret legal entities.

### 3.4.3 Levels of Abstraction

Definitions

*Level of abstraction* Set of information, used for specific purpose, often by individuals with specific background. Set of information is condensed version of full set. Often some portion of full set is frozen. User orientation is important.

*Perspective* View of subset of information taken for given purposes. There can be multiple perspective at same level of abstraction.

Candidate set

Multifunctional piece part
  Functional feature
    Material composition
    Unit cell geometry

Procedure

Examine specific applications and develop levels of abstractions for above. Then generalize.

## 3.5 Lamination Group

**Participants**

Marshall Burns (Ennex), Deba Dutta (University of Michigan), Jack Hilibrand (NSF), Vinod Kumar (University of Michigan), Sara McMains (UCB), Wyatt Newman (Case Western), Cesar Pina (USC/ISI), Dan Siewiorek (Carnegie Mellon)

### 3.5.1 Process Definition

The group considered processes that produced three-dimensional objects through the use of sheets. There are three types of sheet-based processes: subtractive (removing material by such processes as CNC), additive (building up the structure by adding material in such processes as SLA, stereolithography, DTM, sintering, FDM, extrusion, BPM, dropping material/glue) and formative.

Helisys is an example of a sheet-based process. Two fundamental steps are used for each layer: cut a pattern and bond the pattern to the previous layer. Helisys first bonds the layer to the previous layer and then cuts the material with a laser. This is a stack first process in that paper is deposited by paper rolls. A window is cut in the paper at each layer. The unwanted material is diced and acts as a support structure. Layers are 200 microns thick. The final part is broken out of the unwanted material. Depending on the thickness and feature pattern of the object, removal of the unwanted paper can be a difficult task.

### 3.5.2 Properties of an Interchange Format

The goal is to develop a generic layering model that can be used for all SFF fabrication processes. A variety of models are possible:

- Volume elements (voxels) with homogeneous material
- Voxels with non-homogeneous material
  - Layers with arbitrary (height) curvature and anisotropic material properties
  - Functionality based model with geometry or topology

VLSI design has several levels of representation: system, data path/controller, gates, sticks and process planning. Solids could also have several levels: object, slices and process planning. Two languages were proposed - SIF for the design level and LIF (or L-SIF) for the layer level. SIF consists of solid regions with attributes. SIF can be produced by scanning an existing object, generated from an algebraic representation, *etc*. The Lamination Interchange Format would be composed of contours with height, materials and other enclosed contours. The contours could be given by algebraic expressions, vectors, NURBS, *etc*. The process planner would fill in the curvature between layers.

### 3.5.3 Lamination Design Rules

The Lamination Group felt that a generic SFF exchange format could be developed for existing technologies.

Several required properties for LIF were identified. Rules marked with an "R" would be required. The other rules would be optional.

- (R) Minimum feature size: perpendicular to layer, within layer.
- (R) Minimum/maximum object size.
- (R) Minimum surface roughness: perpendicular to layer (build direction) and at top/bottom layers (non-build direction).
- Anisotropy in surface roughness (dependent on direction of surface measurement)
- (R) Material specifications:
  - (R) Mechanical properties , *e.g.*, tensile/compressive strength, shear, hardness
  - (R) Density - mass, porosity
  - (R) Thermal properties, *e.g.*, melting point, coefficient of thermal expansion, thermal distortion, warpage
  - Degradation (environmental stresses that change material properties) - thermal, chemical, mechanical, photonic and aging
  - Finishability - machinability, coatability, *etc*; anisotropy in tapping, *etc.*
  - Post processing requirements - material dependent.
- (R) Negative material removal.
- (R) Dimensional tolerances.
- Production rate.
- Parallel fabrication capability - in 3D, nested.
- Mechanisms without assembly.
- Support structure limitations.
- Cost.
- Environment.

# 4. Appendices

## 4.1 Workshop Attendees

| | |
|---|---|
| Erik Antonsson | California Institute of Technology |
| Joseph Beaman | University of Texas at Austin |
| Marshall Burns | Ennex Fabrication Technologies |
| Bernard Chern | National Science Foundation |
| Richard Crawford | University of Texas at Austin |
| Deba Dutta | University of Michigan |
| Susan Finger | Carnegie Mellon University |
| Paul Fussell | ALCOA Technical Center |
| James Glimm | State University of New York |
| Jack Hilibrand | National Science Foundation |
| Aravinda Kar | Phillips Laboratory |
| Chris Kasabach | Carnegie Mellon University |
| Prashant Kulkarni | University of Michigan |
| Vinod Kumar | University of Michigan |
| Earlin Lutz | Bentley Systems, Inc. |
| Nicholas Mankovich | UCLA Olive View Medical Center |
| Sara McMains | University of California at Berkeley |
| Amar Mukherjee | University of Central Florida |
| Wyatt Newman | Case Western Reserve University |
| Prakash Padmanabhan | Carnegie Mellon University |
| Thomas J. Peters | University of Connecticut |
| Cesar Pina | University of Southern California - ISI |
| Fritz Prinz | Stanford University |
| Richard F. Riesenfeld | University of Utah |
| Emanuel Sachs | Massachusetts Institute of Technology |
| Carlo Sequin | University of California at Berkeley |
| Daniel P. Siewiorek | Carnegie Mellon University |
| Robert F. Sproull | Sun Microsystems Laboratories |
| Ram D. Sriram | National Institute of Standards & Technology |
| John Staudhammer | National Science Foundation |
| Gerald Sussman | Massachusetts Institute of Technology |
| John Stivoric | Carnegie Mellon University |
| Herbert Voelcker | Cornell University |
| Lee Weiss | Carnegie Mellon University |
| Steven Yencho | Stanford University |

## 4.2 Workshop Agenda

### Monday - June 5, 1995

| | | |
|---|---|---|
| 8:30-9:00 | Continental Breakfast | DSSC Conference Room |
| 9:00-9:05 | Workshop Introduction - Dan Siewiorek | DSSC Conference Room |
| 9:05-9:15 | Workshop Charge - Bernie Chern | DSSC Conference Room |
| 9:15-Noon | State-of-the-Art Surveys<br>• VLSI Example - Carlo Sequin<br>• Process Development - Lee Weiss<br>• Modeling - Rich Riesenfeld<br>• Process Users - Paul Fussell<br>• Design - Chris Kasabach & John Stivoric | DSSC Conference Room |
| Noon-1:00 | Working Lunch | DSSC Conference Room |
| 1:00-4:00 | Breakout Sessions<br><br>•Stereolithography<br>•Laser-based SFF<br>•Shape Deposition Manufacturing<br>•3D Printing<br>•Lamination | Separate rooms |
| 4:00-5:00 | Plenary Session Wrap-Up | DSSC Conference Room |
| 5:00-6:00 | Tours of EDRC SFF Facility | |
| 6:00 | BBQ and Cookout | Physical Plant Patio |

### Tuesday - June 6, 1995

| | | |
|---|---|---|
| 8:30-9:00 | Continental Breakfast | DSSC Conference Room |
| 9:00-Noon | Breakout Sessions | Separate Rooms |
| Noon-1:00 | Working Lunch | DSSC Conference Room |
| 1:00-3:00 | Plenary Session - Breakout Wrap-ups | |
| 3:00 | Adjourn | |

# 4.3 Design Methodology for Rapid Prototyping

**Jack Hilibrand and Bernard Chern**

## 4.3.1 Introduction

Emerging national needs and recent technological advances point to rapid prototyping as a key basic research area. Rapid prototyping allows designs created using computer-aided design (CAD) systems and described in an all-digital format to be "prototyped" either by simulating their performance or by constructing, using an automated process, limited numbers of actual parts to ascertain performance. Shortening the design/evaluate/build cycle is critical to reduce "time to market," an increasingly important industrial need. Reducing the design/evaluate/build cycle time also permits multiple cycles of learning in the development and refinement of a product. In some areas, such as integrated-circuit design, tools and techniques for virtual and physical prototyping are advanced, in widespread use and effective in speeding product development. In other areas, notably mechanical design, there is much work to be done.

## 4.3.2 Rapid Prototyping

Prototyping consists of a virtual process (modeling and simulation) and a physical process (hardware realization). Virtual prototyping is the substitution of computer models for physical objects and processes. Virtual prototypes permit modeling a complete system (*e.g.* a computer and its operation) at several levels of detail. Examples include the switch-level models used in prototyping large-scale integrated circuits and the 3D solid models used in designing mechanical systems. To substitute for physical prototypes, virtual prototypes must accurately reflect the properties of objects, allowing a range of design and evaluation activities. Moreover, computational analysis techniques must be available to determine important performance properties. For example, computational fluid dynamics methods are used to analyze airfoil designs. To make effective use of virtual prototyping, we require accurate representation and rendering of objects, high-speed access to remote resources such as libraries of models, and high-performance computation to carry out the analysis, evaluation and refinement of these models.

Building physical parts rapidly based on the digital description of a design requires a combination of suitable computer-controlled manufacturing equipment and algorithms that convert the digital description into commands to the machine. Although computer-controlled machine tools are now commonly available, the algorithms and tools for driving them automatically are still a matter of research. In machining, for example, difficult problems of clamping the workpiece and of designing and building the necessary jigs still require human intervention and ingenuity.

In some fields, however, rapid prototyping is commonly used. The MOSIS system pioneered rapid physical prototyping of VLSI chips using digital descriptions created in universities and companies all over the country and sent by computer network to the MOSIS service. MOSIS converts the digital information into masks for building a limited number of wafers that contain at least one copy of each design. The chips yielded by each wafer are packaged and returned to the designer for evaluation. Critical to the success of this system is a set of explicit design rules (sent from MOSIS to the designer) that characterize precisely the structures that the VLSI fabrication process can manufacture reliably. Conformance of a proposed design to these design rules is checked by the CAD system before a design is submitted for fabrication. The CAD system also typically contains simulation tools that let the designer exercise and evaluate the design—using virtual prototyping—even before it is sent to MOSIS. Rapid virtual and physical prototyping play an important role today in enabling university-based research in experimental computer science to explore novel hardware and software systems. Industry also uses these techniques for VLSI designs.

Despite these successes in rapid prototyping of VLSI chips, there are huge challenges remaining. Designers increasingly want to simulate entire systems—not just chips—before they are built. The need to evaluate performance over many billions of clock cycles demands new levels of simulation capability. Also VLSI designers are increasingly required to analyze power consumption, heat flow, mechanical stresses and transmission line behaviors (especially for propagation of very high speed signals).

### 4.3.3 Rapid Design

One of the differences today between VLSI and mechanical design is the availability and the use of sophisticated CAD tools that communicate among the many levels of abstraction in the VLSI design methodology. It is less the lack of availability of descriptive and analytic tools in the mechanical realm than it is the problem of their disjoint nature. An effort has been made in VLSI to make the transitions between descriptions at various levels of abstraction smooth and automatic. In mechanical design these transitions often involve moving to a different descriptive model for the object and an entirely different language.

### 4.3.4 Rapid Design and Prototyping

The goal of research in rapid design and prototyping is to develop and integrate the methodologies, tools, environments and technologies needed to be able to design and build processes, artifacts and systems of artifacts rapidly, accurately and efficiently.

The individual steps of prototyping, encompassing computer-aided design of parts and systems as well as build and test, must be integrated. New technologies are available for use in rapid prototyping such as: compact rapid prototyping and manufacturing centers; mini-fab production lines; and new methods for producing parts and systems by adding material (stereolithography, laser sintering, spray on techniques) rather than deforming or removing material (forging, grinding, machining). These new fabrication technologies combined with new design methodologies and coupled with innovative services and an updated infrastructure that allows distributed use of tools over high speed networks and timely transfer of tools, technology and design descriptions between various design platforms hold the promise for truly integrated CAD systems aimed at rapid prototyping.

Key issues in creating such a system for rapid prototyping will involve:

- tailoring language formats for various levels of system design so that the information can be used to develop flexible design environments;
- identifying limitations in today's CAD methodology so that a new generation of system-oriented tools can be developed;
- reducing the cost and time for prototype fabrication with new tools, equipment and services;
- exploiting new fabrication technologies and new design methodologies; simplifying and automating some of the new fabrication processes; and
- establishing a better relationship between tool designers and those doing fabrication and prototyping, in areas such as requirements, integration and evaluation of performance.

### 4.3.5 Research Opportunities

An NSF workshop in 1994 [1] examined the VLSI Design Paradigm and whether such a paradigm could be applied to mechanical design and prototyping. The workshop concluded that particular technologies using layering processes, such as Solid Freeform Fabrication processes (SFF) and Micro Electro-Mechanical Systems (MEMS) processes, were the most promising candidates to focus on initially to encourage research for applying VLSI-like system design paradigms. The key to the importance of SFF technology is the basic process model that involves the assembly of two dimensional layers. It is this 2.5 D model that enables the representation of the manufactured object in terms of a multilayered geometric structure similar to that used in VLSI.

This workshop on design methodology for SFF technologies is being held to explore the research issues involved in implementing an SFF design methodology, with particular emphasis on digital design/build interfaces to SFF processes and related research on CAD tools, levels of abstraction and design rules. Other workshops and planning will follow to address CAD-based design methodologies for a wider range of technologies including, for example, MEMS. Successful introduction of a VLSI-like paradigm for designing and prototyping electro-mechanical objects and systems would have major impacts on today's mechanical design and prototyping capabilities (quicker turnaround, improved simulation and test verification, multiple cycles of product improvement before production, *etc.*).

### 4.3.6 Long Term Strategy

The long term objective of this NSF Design Methodology Research effort is to support research to evolve a design methodology that can be generally applied to mechanical, electromechanical and electrical systems. Such a design methodology might have the following properties:

- incorporates design heritage
- builds on hierarchically related elements
- creates a clean separation between the design and process communities and yet ensures manufacturability based on design rules unique to the selected manufacturing processes and technologies. Conversely, the design rules permit the designer to select among the available technologies based on process limits.
- incorporates process information as background knowledge on which the designer does not need to focus his attention
- incorporates multiple levels of abstraction to permit design at a high level with a smooth, hands-off transition (possibly including automatic partitioning and artifact synthesis steps) down to design rule checking of fabrication instructions
- provides rapid and accurate transitions between descriptions at different levels of abstraction so that the final product is the result of multiple cycles of learning about the system and its properties in the virtual realm, enabling optimum responsiveness to market requirements.

As recommended in the NSF Workshop on New Paradigms for Manufacturing, attention is now being focused on defining a design methodology for SFF technologies. Because the SFF technologies all involve layered fabrication techniques, it appears that a unified manufacturing process model can be used that emphasizes the 2.5 D nature of the geometry and calls to mind use of a VLSI-like CAD system. There are several SFF prototyping methodologies that are now in use but it *may be possible* to identify a single CAD description language (a digital interface language - see the next section below) that describes the requirements and a single design methodology that will be effective for all SFF technologies (based on the layering process model). Designers, then, will work at the digital interface language and higher levels of abstraction while fabricators will move from the digital interface language to machine instruction languages for their specific equipment.

The present workshop is intended to stimulate research on defining and implementing an integrated SFF design methodology. It is hoped that the extension of these techniques (in future workshops and future research) will proceed from SFF to MEMS to electromechanical designs and then to more general mechanical designs. The benefits of developing such an integrated design methodology are especially valuable for the rapid prototyping technologies since it is in this arena that fast and accurate transition from a high level system description to the fabrication technology instructions is most useful. The scope of these efforts should include not just the design methodology, but also the other tools, design environments and design technologies that will be used in rapid prototyping.

### 4.3.7 The Digital Interface[1]

The digital interface is intended to provide a format and a language for communication between the designer community and the fabrication services vendor community. More than that, however, it provides the clean separation between the designer and the technologist so that the designer works with proven technology and meets the design rules that tell him how far he can go in that technology and the technologist is sure that the submitted design can be built since it conforms to those design rules that he set forth as describing the limits of his technology. This is rapid design and prototyping, if only because there is no need to reset and redo design or process, since the design rules assure manufacturability to both communities. The digital interface describes the object desired by the designer. It does not incorporate process information explicitly but it may depend on shared knowledge between the designer and the fabricator about the properties of the process to be used. In this instance we are concerned to provide a digital interface for the fabrication of objects using one or more of the SFF technologies. For the SFF technologies, the language might define the geometry of the object in terms of a sequence of layers so that both designer and vendor can deal with it. In any case it is desirable to use a common descriptive language for all the SFF technologies, if that is possible, and to assign designs to specific SFF technologies based on the applicable design rules.

The digital interface language in the VLSI world is CIF (Caltech Intermediate Format), which describes the layers comprising an integrated circuit in terms of rectangles and lines. These layers are known by all users to correspond to mask levels and the associated process steps. This VLSI digital interface language permits the designer to do hierarchically nested designs in which established building blocks can be assembled. It also permits the fabricator to check a product design for conformance to design rules and to convert the description directly into the digital language used for driving his computer-controlled equipment.

The digital interface language for SFF should be as simple as possible, easily understood and easily handled since many people will have to learn it. It should provide a complete and accurate description of the object since the vendor should have no discretion in executing the order. In addition it is desirable that the digital interface defines the intended object itself and is independent of the process used to achieve it. The effectiveness of communication through the digital interface language should not depend on the possession by designer and vendor of large amounts of detailed shared information that must be kept complete and up to date.

---

[1]Much of this discussion of the digital interface originally came from discussions with Bob Sproull in connection with the 1994 workshop.

This digital interface is not a complete description of the mechanical system. It is the description of a single object for the purposes of communication with the fabrication vendor. It is at the lowest level of the design hierarchy so that it can be simple and easily used.

Some of the advantages of digital interfaces were cited by Bob Sproull in his paper at the Workshop on New Paradigms for Manufacturing [1]:

- The ability to communicate fabrication orders from one site to another, either because design and fabrication are carried out at separate sites, or because the order must be filled by a different fabricator than originally intended.

- The ability to broker the fabrication job to one or more vendors who offer the same interface definition. Part of this process might include soliciting quotes.

- The ability to store an order to be used later. For example, spare parts need not be fabricated in advance, but if the digital order for each part is saved, it can be used later, as needed, to fabricate replacements.

- The ability to automate a process that verifies that the order conforms to certain specifications or design rules; in effect, to form an agreement between requester and provider that the job being requested is reasonable and that it will be built as described.

- The ability for participants on both sides of the interface to enjoy direct benefits of further automation, *e.g.*, a designer who can use CAD software to prepare a design, and a fabricator who can prepare a process plan automatically. Importantly, the digital interface allows composition of fabrication processes, *i.e.*, in order to fill a complex digital order, a fabricator can perhaps use automatic software to prepare an order for a sub-assembly that will be filled by a different fabricator.

Some other advantages of using digital interfaces to fabrication processes were discussed in the Group 1 Report at the first Workshop [1]:

- synthesis systems can be constructed that use a hierarchy of building blocks based on the digital description

- analysis using the digital description can serve as the basis for a finite elements decomposition of the item along with analytic programs for assessing stress, strain, temperature, wear, *etc.*

With this background on design methodologies and digital interfaces set forth, we can now proceed to state the key questions that need to be addressed in the mechanical design and implementation methodologies and to separate those that should be the focus of this SFF workshop from the more general questions that can be addressed in future workshops.

### 4.3.8 Key Questions Relative to SFF Design Methodologies

**Design Methodologies** - For each SFF technology, what is the design methodology now being used in terms of: (1) the specification language used, (2) the appropriate design rules for that technology, (3) the design tools available for today's design methodology and (4) the levels of abstraction available for use in design. Comparing these different design methodologies used for the SFF technologies: What is the nature of the differences and are they substantial or are they minimal? What would constitute a generic SFF design methodology? Can we group the SFF technologies in some meaningful way (such as single- and multi-composition deposition processes, for example) that will enhance the commonality of SFF design methodologies within each grouping?

**Design Rules** - If the SFF design methodologies and design descriptions can be brought in line, can we get uniformity in the formulation of the design rules so that, although there may be differences among the technologies in the tightness of the design rules, those rules can be stated in the same terms? Are there some inherent differences in the structure of the design rules among the SFF technologies? Can we encompass those differences with a more general type of description (for example, by treating material uniformity in a layer as a special case of material gradients where the gradient is zero)?

**3D Geometric Models** - For SFF technologies, is there a common 3D geometric model that can be used for representation and manipulation of object models? What solid modeling languages describe 3D shapes so that the solid models can be reduced to 2.5 D layer descriptions most effectively? What design tools and frameworks will tie together these descriptions and how do we get compositional, structural and calculated property information into the models? Is there a generic methodology for representing and manipulating the 2.5 D models in the SFF technologies?

**3D Modeling for Composition, Structure and Properties** - What is the best way of handling 3D composition, structure and property variations in the descriptive model? How do we deal with these variations in the simulation and modeling areas? Can we find a common representational methodology for composition and structure variations in the several SFF technologies (based on the 2 1/2 D geometry)? How can we do top-down design involving material and property variations along the layers? Between layers? Are there *ad hoc* techniques that are now being used by some of the SFF design groups and are these techniques more broadly applicable to other SFF technologies?

### 4.3.9 Long Term Considerations

**Development Methodology** - Can we identify a design/development methodology for mechanical systems that takes full advantage of a hierarchy of modern computer-based design tools? In such a methodology, what are the appropriate roles of and transitions between virtual prototyping (modeling, simulation, validation and assessment), rapid physical prototyping (by SFF fabrication technology, for example) and full scale manufacturing?

**Design Methodology** - How do you proceed from an overall system description to a partitioned system description (in terms of artifacts or objects) to fabrication tool instructions? What levels of abstraction are appropriate for top-down mechanical design processes?

**Languages** - Can we identify and specify easy-to-use geometric description languages for the mechanical designer and a digital interface language between the designer and the fabricator? Can we incorporate into these languages the description of other key attributes: materials composition and structure, properties, tolerances, etc.?

**Design Rules** - How can we formulate the design rules at the digital description interface level that will assure manufacturability for the specific fabrication technology being used? How can we test object descriptions for conformance to these rules (design rule checking)?

## References for Sections 1 through 4

[1] *New Paradigms for Manufacturing*, NSF Workshop Report NSF 94-123, Arlington, VA, May 2-4, 1994.

[2] MOSIS User's Manual, http://broker.isi.edu/mosis. Chapter 4 describes the e-mail message formats for exchanges between the designer and broker.

[3] ACIS manuals, Spatial Technology Inc.

[4] C.M. Hoffmann and R. Juan, "Erep—An Editable, High-Level Representation for Geometric Design and Analysis," *IFIP Transactions, Geometric Modeling for Product Realization*, P.R. Wilson, M.J. Wozny and M.J. Pratt, eds., Elsevier Science Publishers (North-Holland), 1993, pp. 129-164.

[5] M.R. Henderson and L.E. Taylor, "A Meta-Model for Mechanical Products Based Upon the Mechanical Design Process," *Research in Engineering Design*, 1993, 5: 140-160.

[6] K. J. Weiler, "Topological Structures for Geometric Modeling," Doctoral Dissertation, Rensselaer Polytechnic Institute, 1986.

[7] J. H. Bohn and M. J. Wozny, "A topology-based approach for shell-closure," in *IFIP Transactions, Geometric Modeling for Product Realization,* eds., P.R. Wilson, M.J. Wozny and M.J. Pratt, August 1992, pp. 297 - 320.

## 4.5 Workshop Presentations

# What Can SFF CAD Learn from the VLSI CAD Revolution?

Carlo H. Séquin and Sara McMains
CS Division, U.C. Berkeley

## 1 VLSI Design and Fabrication

Before forging ahead with new standards for SFF, it seems worthwhile to look at design and fabrication in the domain of integrated circuits and to review the VLSI CAD revolution of the 1980s. There may well be some valuable lessons that carry forward to the domain of SFF.

IC fabrication is a 2.5-dimensional technology. There are many different classes of processes: NMOS, CMOS, Bipolar, BiCMOS, GaAs ... In each class there are many variations among different fabrication lines. But in all cases the principle is the same: one starts with a flat wafer, usually a silicon crystal lattice with a single orientation, and then produces a series of changes in areas that are defined by the geometrical patterns on a sequence of fabrication masks. In order to produce a "Si-gate NMOS" circuit (a vertical layering of Metal and Oxide, on top of a Semiconductor which has regions of Negatively charged carriers), one might use the following key conceptual steps. One mask pattern (often symbolically shown in green color) and the associated processing steps may cut open a pattern of trenches into the oxidized silicon crystal surface, into which small amounts of "impurities" are diffused in order to produce regions that are more conductive than the bulk of the silicon substrate. A second (red) pattern may define connection strips in a deposited polycrystalline silicon (Poly-Si) layer on top of the previous structure. Where these strips cross the "green" trenches, transistors are formed: the voltage on the red strips – or "Si-gates" – controls the amount of conductivity in the green trenches underneath. A fourth (blue) pattern defines another layer of interconnections, in a metal layer on top of the whole processed wafer. A third (black) pattern specifies where connection holes should be cut through the insulating layers placed between the "red" and "blue" interconnection levels.

For a given class of processes, the basic functionality of the various layers is always the same, even though the thicknesses of the many layers and the exact cross-sectional shapes of the trenches and conductor strips may vary. In a different process class, the levels may perform different functions; for instance, in an older "metal-gate NMOS" process, the amount of conductivity in the semiconductor bulk layer is controlled by gates formed in the metal layer. Whoever designs the masking patterns for a particular circuit has to understand what the basic functions are of the various patterned layers. In the early 1970s the design of a circuit layout was an "art" where each designer had to work closely with the people operating the fabrication line in order to define appropriate patterns that together with the envisioned processing sequence would then produce the desired circuit behavior.

One of the contributions of Mead and Conway ["Introduction to VLSI Systems," Addison Wesley, 1980] was to push for a simplification and standardization of this interface between designers and fabricators. For many applications, the performance of the digital logic circuits is not very critical compared to the capabilities of the technology and will be good enough even when the potential of the chosen fabrication process is not used to its limits. It was thus possible to define a simple set of geometrical layout rules that would assure that functional transistors and circuits of acceptable performance would be produced for fabrication lines whose processing parameters were kept within reasonable parametric ranges. This was done first for the Si-gate NMOS process and later for "CMOS" technology (Complimentary MOS, with transistors of both polarities). This simplification allowed an abstraction of the design process, where geometrical "recipes" would produce transistors of desired performance (speed, driving power) or circuits of desired functionality (AND-gates, NOR-gates, multiplexors ...).

To make this abstraction work, it is crucial that the semantics of the various patterns is well defined. After some initial controversy, the geometrical shapes were defined to represent "what you can see when you look down on the final IC chip." Since the actual masks used in fabrication may actually have different dimensions, owing to expansion or shrinking of the various features during the fabrication process, or owing to size changes when masks are copied, it became the responsibility of the fabrication houses to compensate for all these cumulative effects and to produce the masks for their own use that would deliver the specified geometrical dimensions on the final chip. Thus the fab-line dependent variations and idiosyncrasies became invisible to the circuit designer, and the same set of geometrical pattern specifications became portable from one fab-line to another. As technology improved and it became feasible to make ever smaller features and thus faster and more compact circuits, it was even possible to take the old patterns and to shrink them automatically to make them usable again for these advanced processes. At this point one has reached a true abstraction of the layout design process, where the designers may not even know what the dimensions are of the rectangles that they are drawing.

# 2 Higher Levels of IC Abstractions

After the lowest level of abstraction of implementing pieces of integrated circuits (by specifying the raw geometry of all the features on the chip) became routine, and ever larger collections of useful and reliable circuit elements had been compiled, it became practical to shift the level at which designers would think about their circuits to a higher level. The logic level had already become well established with the library of bipolar circuits described in Texas Instruments' famous TTL Data Book. The semantics or the functions of the various gates was quite clear, and library cells that implemented these various useful logic circuits could readily be designed and optimized for different fabrication processes. The designers now simply had to plug together these "logic gates" into more complicated digital subsystems.

Very soon, frequently used higher-level circuit components emerged: registers, data-paths, memory blocks, arithmetic-logic-units (ALUs); these became the natural – and often parameterized – building blocks for larger systems such as processor architectures. This "register-transfer-level" of abstraction is most conveniently used when a designer tries to trade off system speed against implementation costs and tries to find the most cost-effective degree of parallelism in the architecture.

Between the layout level and the logic level, there are a couple of other abstraction levels that were often used by many designers. First there is the "sticks" level in which the connections in the various layers of the integrated circuit are not represented by their exact geometry, but only by their topological ordering and arrangement on the chip surface. This level can be drawn more quickly and frees the designer from worrying about exact dimensions and about layout-rule correctness. A computer program, called a "compactor," then converts a sticks diagram into a densely packed, layout-rule correct geometric layout. However, this level never became useful for exchanging designs between designers in different groups, since the semantics of this description was never clearly defined and standardized.

Similarly, yet another abstraction level, the "switch" level, never played a mayor role as an interchange standard, even though this level is very important for circuit simulations and for optimizing the performance of a generic circuit. Too many parameters would have to be specified with each component at the switch level, and the actual performance would then still be dominated by many parasitic circuit elements, such as stray capacitances that are strongly affected by the exact layout and by the relative arrangements of the circuit components on the chip.

Above the register transfer level, there are also further levels of abstraction. But as one climbs higher up in this abstraction hierarchy, the concerns of the designers become richer and harder to define. At the "functional" or at the "behavioral" levels, one tries to formally capture what the system overall is supposed to do – but this is rather difficult. For a long time, certain semantic issues had been left unanswered: for instance: "What is the meaning of time in this domain ? Is it continuous or discrete ? Is it fully synchronous to some clock ? And if so, how are external asynchronous events handled ?" The lack of semantic clarity and completeness made it difficult to exchange systems specifications at this level. In the last few years, however, a standard language (VHDL) with reasonably well defined semantics has emerged.

# 3 Conversion and Checking Tools

Defining these languages and interchange formats with proper semantics is just the beginning; they are not much use unless we also have convenient tools to create descriptions at these various levels and to convert these descriptions from one level to another.

Programs that convert descriptions from higher levels of abstraction to lower, more detailed description levels are known as "generators" or "compilers." For example, we might have a parameterized "N-bit full adder" generator, which – upon specifying a value for N – will generate an adder circuit with N bits in a particular technology. This relieves the individual designer of the tedium of explicitly specifying all the geometric details of the layout of such a circuit.

The inverse process, finding a higher-level description from a large – possibly unstructured – mass of lower-level primitives is known as "extraction." This is often used as a verification step after a lengthy, multi-stage compilation process, e.g., to verify that a large collection of layout rectangles indeed constitutes a circuit with the desired properties. Such a circuit-extraction process may also find spurious circuit elements that never show up explicitly in the generation process. For instance, long interconnection lines may have associated with them a significant amount of capacitance to the substrate. To adequately model the performance of such a "transmission line," this capacitance has to be calculated and suitably incorporated into the model of the circuit.

To close the verification process, the extracted circuit would have to be compared to the original, intended circuit. This can basically be done with some graph-matching algorithm; however, some margin of error has to be tolerated in the parameter values for each individual circuit component because of the parasitic geometries mentioned above. There is another difficulty: the same flat circuit or logic description can often be hierarchically grouped in different ways that are equally valid or "natural." The extracted hierarchy may then be quite different from the one originally conceived by the designer, and thus hard to compare. It might thus be worthwhile to leave "hints" or other informal information in lower level descriptions that convey the original design intent.

There are also internal consistency checks that can be made on individual descriptions alone. For instance, in a circuit diagram, one can check that differently labeled signal or power lines are not short-circuited to one another, or that all terminals with the same labels are indeed internally connected. At the level of the layout geometry, one would typically apply some geometric checks to see that the chip can be fabricated with reasonable tolerances and will still yield – with high probability – the geometric patterns that the designer had in mind. Primarily, such layout checks would test for certain minimal separations between elements that should not fuse together, and for certain minimal internal dimensions for paths that should not be broken. Comparing the geometry on two different levels, one would check for sufficient tolerances in the mutual overlap of features, so that one retains coverage even in the presence of small mask misalignments. All these checks are tightly coupled to the semantics of the features represented and to an understanding of the capabilities of the processes that are controlled by these descriptions.

# 4 What Can We Learn from This ?

Many similarities become apparent when looking at some of the SFF processes in the context of the above experiences with VLSI. Here we present our initial reactions to the questions raised in the call for participation for this workshop. In many instances where the situation in SFF is sufficiently different from that in the domain of VLSI, we are not ready to take a definite stand; we then simply present the pros and cons of an issue to help focus the discussion at the workshop.

## 4.1 Modeling and Design Exchange

> *What level of abstraction should be used for describing the physical design? Should it be two-dimensional layers corresponding to the layers which are built during the fabrication process? Or should it be a form of a three-dimensional description with features identified?*

A single level of abstraction will definitely not be sufficient. There is a need to describe mechanical systems and their physical parts at one or more high-level abstractions that capture various aspects of their functionality. These higher levels of abstraction are outside the concern of this workshop.

**SIF**:
However, at some point, the shape of an individual part has to be defined, and a clean abstraction level is needed to describe the geometry of this part in a completely fabrication-process-independent way. This geometric shape specification should clearly be a 3D description. Many existing solid modeling languages would be quite adequate for this purpose. They should have the power and expressibility to describe smooth surfaces in a resolution-independent, non-tessellated, compact form, and should be able to represent cylinders and spheres perfectly. NURBS seem to be one of the obvious choices for that purpose.

However, the variety of existing solid modeling systems and the possible complexity of the constructs they offer may be bewildering and scary to the manufacturing people who should be able to accept and process models in this form. It might thus be worthwhile to define a lean and clean subset language that contains only the really essential elements and has unambiguous semantics. For discussion purposes we will call such a language "SIF," or "Solid Interchange Format." This should be a compact, human-readable ASCII language in the spirit of CIF. The format should be hierarchical to avoid the waste of transmitting repetitive data.

"Features" (e.g., a "hinge," or a "parameterized dove-tail slot") – like functionality – do not belong at the SIF level, but at a higher symbolic level. While including such features may make the job of the process planner and the rule checker easier (see Weiss and Prinz draft position paper), it may be too daunting to define a rich enough catalog of features for all current and future application domains. (Possibly, if a high-level description exists that includes features, this information might be passed down to lower level descriptions to make the extraction of fabrication features easier. But we would then need to discuss to what degree such "hints" are an informal local help and to what degree they should be officially supported in SIF.)

In general, any such language should be kept as simple as possible, but general enough to do all the jobs envisioned, and extensible in case new needs emerge that cannot be handled by the original format.

**L-SIF**:
There is also a need for lower level part descriptions that are much closer to actual fabrication plans. For SFF, a natural candidate would be some 2.5D layered description that is comparable to the CIF description of mask levels for a particular fabrication process. For the sake fo discussion, we call this format "L-SIF," for layered-SIF. The L-SIF descriptions for one and the same part may be quite different for different SFF fabrication processes.

Ideally, L-SIF would have a very similar style and semantics as the SIF language and should in principle be a derivative thereof. On the one hand, it might contain fewer geometrical constructs than SIF, since it might only have to express the 2D shapes of subsequent layers; thus, rather than having to describe NURBS surfaces, it would only have to express NURBS contours. On the other hand, the L-SIF languages for different SFF fabrication classes may need to convey some extra information such as the direction of a material-depositing nozzle movement. For some processes, such as Shape Deposition Manufacturing, the L-SIF process plan description might require almost the same richness of geometrical information as the idealized SIF part specification, since the individual layers are so thick and individually machined around their perimeters that they must be considered 3D solids in their own right.

Finally, below the level of L-SIF, there will be a plethora of machine-dependent control languages over which our task force will have little influence.

> *What type of model should be used to represent designs? What should be the role of traditional solid modeling? What attributes should the model provide in describing the design other than geometry? Candidates include strength, material, tolerance, etc.*

Traditional solid modeling systems have mostly relied on polyhedral or on smooth "boundary representations" (B-REP) or on "constructive solid geometry" (CSG) assemblies of a few clean primitives (cubes, cylinders, spheres...), or of a combination thereof. While other models exist for describing geometry that are mathematically purer, it is not clear that they are preferable to describe shapes at the SIF or at the L-SIF level or that they would be accepted by a generation of engineers that grew up with today's CAD and modeling tools.

For cleanliness and compactness of the description it may well be desirable to allow SIF descriptions to be a combination of B-REP and CSG: i.e., regularized boolean set operations trees whose leaves are B-REPs of 2-manifolds, half spaces, and partially bounded objects. This would allow, for instance, a compact, resolution-independent specification of a shape defined by one or more general NURBS with some cylindrical holes in it.

At the SIF level, the different geometrical regions defined would have to carry information on the "desired" volume or surface properties, which the manufacturing processes would then have to approximate as closely as possible. Material selection, density, color, tolerances, and surface roughness may be such properties that need to be specified. This could be accomplished by using a volume statement that takes a solid region and specifies the "content" or materials properties of that domain. Here is an example that demonstrates the use of CSG and of several geometric regions occupying the same space: graded coloring could be specified in a large bounding box that contains the whole part and which is formally intersected with it.

At the L-SIF levels, the volume or boundary information may then be much more dependent on the particular SFF process envisioned and specifically refer to some of the available materials that could be dispensed in each layer, or to the particular local treatment that these layers may experience at certain locations.

Specifying the material at the SIF level raises some interesting issues. Since different processes have different material capabilities, a design that includes a material specification is no longer process independent. For a manufacturing process where SFF will only be the first stage of the tooling, such as making wax positives for investment casting, the material the SFF uses won't be the material of the final part. But for a different SFF process that can make parts directly out of metal, the SFF will use metal instead of wax to produce what will ultimately be the same part. (We should probably limit ourselves to specifying the properties and materials of the primary part emerging directly from the SFF process.)

A key issue is that of auxiliary support structures that are not part of the final geometry but are required for a specific manufacturing process. For example, wax patterns for investment casting need to be attached to the sprue, gates, and runners in order to distribute the molten metal, while wax patterns that are to serve as conceptual models would not need this additional geometry defined. Similarly, a part produced by stereolithography (SLA) will need support structures for cantilevered portions of the model, whereas these would be unnecessary if the part were produced by selective laser sintering (SLS). Generating such "supporting" geometrical features is not yet fully automated. While the designer may have to think about the role that these supports play in her part design and how the supports will get removed and how the break-off ridges may get smoothed, the actual placement of the support geometry is normally done by the experts at the fabrication service, since they are much more knowledgeable about the needs for supports in their particular process than the designer.

Another issue that may get addressed quite differently at the SIF level and in the L-SIF descriptions concerns the way one may specify embedded components. At the SIF level, the designer may simply specify the location of the component and it could then be inferred that the embedding material has a corresponding cavity at that location; alternatively, a suitable cavity could be specified explicitly with a boolean difference operation. At the L-SIF level, the cross section of such a cavity may appear explicitly on each layer; alternatively, an explicit pause statement at the layer touching the minimum $z$-location of the part to be embedded could be generated by the slicer program that creates the L-SIF description from the higher level SIF part description. The .STL format currently has no explicit way to handle such embeddings, making it necessary for the designer to convey her intent to the fabrication house separately. The SLA machine operator then adds a "pause" statement to the control code at the layer where the embedded component will be placed, and then manually inserts the desired component.

*What should be the form of a design exchange format? Can the format support alternative SFF processes? Is there a common set of information required by all SFF processes?*

Ideally, the exchange format should be at as high a level as possible and should be process and machine independent; this is the level of SIF which specifies **"what"** shape is desired, but ignores **"how"** this is accomplished. However, this approach is appropriate only if the worst case process inaccuracies are smaller than the finest feature that the designer is concerned about.

As the design specifications get closer to the limitations of the technology, the idiosyncrasies of the fabrication process will have to be taken into account. Today, most SFF processes have quite different capabilities, and most are not even isotropic, so that the part orientation during the fabrication process becomes a crucial issue. A particular SFF process will thus have to be targeted and parameters such as part orientation, layer thickness, and auxiliary process steps such as curing or shot peening may have to be specified explicitly. Under those circumstances, it may be more appropriate to send a part specification to the fab-house in the form of a process plan at the L-SIF level. Suitable CAD tools must be developed to assist in this conversion process. The key is that each process and the corresponding conversion must try to approximate the desired geometry as closely as possible within the technological limitations.

Internally to the fab house, this L-SIF description may then be converted further into a sequence of instructions to drive a particular machine – again trying to realize the specified 2.5D geometry as closely as possible. This latter step should definitely be hidden from the client.

## 4.2 Design Rules and Tools

*What type of design rules can be defined that when applied to the representation will guarantee successful fabrication in a series of SFF fabrication processes?*

At the SIF level, we may employ generic tests to check that the representation describes closed solid objects (as already exist for .STL files). An other test would be looking for intersections of solids having incompatible volume property specifications, e.g., different materials.

At the L-SIF level, at the latest, different design rules must be applied for each different class of SFF process – otherwise fabrication will be limited to the least common denominator of process capabilities. Rules could include the minimum separation of surfaces that will be guaranteed not to fuse (e.g., to make a tight bearing), and the minimum wall thickness for a given geometry that will not punch through or collapse (e.g., to make a wine glass); this minimum thickness may be different for different orientations.

For SLA in particular, design rule checkers should test whether all parts of an object are anchored to the build platform during all stages of the build process, and whether features have adequate supports. Any process subject to curl could have rule checkers to test whether the curl distortion of the part is predicted to lie within acceptable limits.

> *What should be the nature of design tools to support SFF? VLSI research produced design capture, design rule checking, mapping of logical constructs into physical transistors, physical placement of transistors, and wire routing. Are there analogous tools for SFF including design capture, design critics to identify non-manufacturable features, and fabrication process planning?*

Clearly we want as much automated generation and compilation as possible to speed up the overall prototyping process and enhance its reliability. We also need extraction tools and certain verification tools at all levels. Like the design rules themselves, some design tools will be process specific. Process planners in particular have very different requirements for the different SFF technologies, and very different process planners are shipped with the various commercial SFF systems.

Another question that needs to be addressed is at what point in the process should the orientation for the final build be chosen? Different SFF processes may have different properties along different axes of manufacturing. Definitely the axis perpendicular to any layering is special, but even the remaining two axes may have different properties in some manufacturing processes. Designers may or may not know about these differences. If they do *not* know, they will have to design for a worst case orientation making the weakest assumptions for all directions. If they are very knowledgeable about the process to be used and its performance characteristics in different directions, then they will want to specify the exact orientation in which the designed geometry should be built in the process framework. A useful area of research might be the development of process specific tools that attempt to find the "best" orientation for a part, if the designer doesn't know. For processes that can build multiple parts in a single build cycle, the "best" orientation may also depend on the spaces left by the other parts being built in the same build cycle. Whether specified by the designer or derived automatically, design rules, simulation, and process planning will all need to know the orientation.

# 5 Conclusions

In the domain of VLSI CAD tools, the vision has always been "top-down" with a focus on the "holy grail" of a fully automatic "silicon compiler." However, the suite of tools that has actually made a difference and that has revolutionized the world has been built "bottom-up," with a first focus on simple plotting, circuit extraction, and layout-rule checking tools.

This was followed by synthesis tools that could automatically generate some lower description from the next higher one, and several of which could eventually be cascaded in a sequence of compilation steps.

The final round then was to include optimization in this compilation step; it was no longer good enough to just produce an acceptable solution – one wanted to obtain the "best" possible solution at the lowest possible "price."

We are convinced that this same evolution paradigm will also apply in the domain of SFF fabrication.

**Solid Freeform Fabrication Processes**

Lee E. Weiss
June 6, 1995

*In this talk I presented a brief overview of solid freeform fabrication (SFF) processes. I have attempted to describe here the key ideas represented in each of my slides.*

**Slide 1. What Is Solid Freeform Fabrication?**

The term "Solid Freeform Fabrication" was coined by Joe Beaman, of the University of Texas at Austin. "Solid freeform fabrication is the production of freeform solid objects directly from a computer model without part-specific tooling or human intervention." In other words, SFF is the automated fabrication of arbitrarily complex shaped, three-dimensional parts.

The original impetus for developing SFF processes was to be able to quickly go from CAD models to physical artifacts for rapid prototyping applications such as 3D visualization and form/function/fit testing. In addition, it can be used for quickly creating complex shaped patterns for sand and investment castings, and for making soft-tooling with RTV silicon and epoxy casts.

The continual improvement of material properties, accuracy and building speed for parts produced with SFF processes made it possible to think about directly creating functional parts. The parts created with SFF can be used not only for making better prototypes, but also for low-batch production for emerging "mass customization" markets. As SFF processes evolved, it also became clear that they could be used to fabricate complex designs which could not be realized with more traditional fabrication methods.

**Slide 2. How Is SFF Realized?**

The planning and execution approach underlying the majority of SFF processes in use today is to first decompose a solid 3D CAD model of the object into cross-sectional layers, then use material deposition and fusion techniques to physically build up these layers to form the object. Sacrificial supporting layers may also be simultaneously built up to fixture the object as required.

The layer topology and material composition vary with different SFF processes. In the following slides, I have attempted to categorize the basic SFF approaches and to show examples of some representative processes in each category. The process slides depict the underlying building approaches, but not necessarily the exact automation mechanisms used to implement each system.

**Slide 3. 2-1/2D Layers, Homogeneous Structure with Complementary Support Structures**

In this category, shapes are first decomposed into 2-1/2 dimensional layers (i.e., each layer can be represented by a planar cross-section with an associated uniform thickness). Each physical layer, which consists of the cross-section and a complementary shaped sacrificial layer, is then deposited. These shapes are homogeneous structures in that each consists of a single (primary) material.

The sacrificial material has two roles. First it holds the part, analogous to a "fixture" in traditional fabrication techniques. And second, it serves as a substrate upon which "unconnected regions" and over-hanging features can be deposited. The unconnected regions require this support since they are not joined with the main body until subsequent layers are deposited. The requirements for overhanging features will become apparent in the following process examples.

One advantage of building shapes with 2-1/2D layers is that the process to build each part is rapidly planned and executed *independent of the object's geometry*. There is a trade-off, however, with surface quality.

Several deposition and fusion processes are available which fabricate shapes using this building approach, including Selective Laser Sintering, 3D Printing, and Laminated Object Manufacturing.

**Slide 4. Selective Laser Sintering**

The Selective Laser Sintering (SLS) process was developed at The University of Texas at Austin, and is commercialized by D.T.M. Corp. Several types of materials are used in SLS including nylon, polycarbonate, investment casting wax, as well as metals and ceramics. In SLS, a layer of powdered material is spread over the top surface of the growing structure. A $CO_2$ laser is then used to selectively scan the layer to fuse those areas defined by the geometry of the cross-section; the laser energy also fuses subsequent layers together. The laser beam is directed using computer-controlled mirrors directed by the CAD data. The unfused material remains in place as the support structure. After each layer is deposited, an elevator platform lowers the part by the thickness of the layer and the next layer of powder is deposited. When the shape is completely built up, the part is separated from the loose supporting powder..

**Slide 5. Three Dimensional Printing**

The Three-dimensional Printing (3DP) process is another powder-based SFF approach. It was developed at M.I.T. and is commercialized by Soligen Corp. An ink-jet printing mechanism scans the powder surface and selectively injects a binder into the powder. The binder joins the powder together into those areas defined by the geometry of the cross-section. When the shape is completely built up, the "green" part is separated from the loose supporting powder. Additional post-processing such as sintering may then be usd to strengthen the part. For example, ceramic powers such as alumina bound with colloidal silica can be used to directly create shells and cores for investment casting applications. 3DP of metal powders, such as stainless steel bound with a polymeric binder, is also being explored; subsequent infiltration of the matrix is then required for densification.

**Slide 6. Laminated Object Manufacturing**

Laminated Object Manufacturing (LOM) was developed and commercialized by Helisys Corp. LOM builds shapes with layers of paper or plastic. The laminates, which have a thermally activated adhesive, are glued to the previous layer with a heated roller. A laser cuts the outline of the part cross-section for each layer. The laser then scribes the remaining material in each layer into a crosshatch pattern of small squares, and as the process repeats, the crosshatches build up into tiles of support structure. The cross-hatching facilitates removal of this tiled structure when the part is completed. LOM builds up large parts relatively rapidly because only contours are scanned. LOM is also being investigated by Helisys and The University of Dayton for building up ceramic and reinforced composite shapes using layers of "green" tape castings (i.e., sheets of bound powder); the final part must subsequently be sintered.

**Slide 7. 2-1/2D Layers, Homogeneous Structure with Explicit Support Structures.**

In this approach, support structures are built only when required, i.e., for the unconnected regions and steep overhanging features. One issue for this approach is who is responsible for the design of the support structures; the designer or the fabricator? Examples of this approach include Stereolithography, Fused Deposition Modeling, and Ballistic Particle Manufacturing.

**Slide 8. Stereolithography**

Stereolithography (SLApparatus) was the first commercialized SFF process and is the most widely used SFF technique. It creates acrylic and epoxy models directly from a vat of liquid photocurable polymer by selectively solidifying it with a scanning laser beam. Investigators at the University of Michigan are also exploring the use of ceramic loaded resins to build up "green" shapes. The part is built on an elevator platform which is lowered into the liquid vat. As the laser draws a cross section in the x-y plane, a solid layer is formed on the elevator platform. The platform is lowered and then the next layer is drawn in the same way and adheres to the previous layer.

Features with gradually changing overhangs can be built up without support structures. Large overhanging features, however, require supports since the initial thin layers which form them can warp or break off as the part moves down into the liquid. I have shown the supports in a different shade to highlight that they are typically drawn out as thin wall sections which makes it easy to break them away from the part upon completion.

Investigators at the University of Michigan are also exploring the use of photocurable resins loaded with ceramic powders to form "green" compacts with SLA. In this case, the photocured resin binds the powder which is then sintered upon completetion.

**Slide 9. Fused Deposition Modeling**

Fused Deposition Modeling (FDM) was developed and is commercialized by Stratasys, Inc. FDM is an extrusion-based process which deposits a continuous filament of a thermoplastic polymer or wax through a restively heated nozzle. In the current system the material is delivered as a wire into the extrusion head. The material is heated to slightly above its flow point so that it solidifies relatively quickly after it exits the nozzle. Therefore, it is possible to form short overhanging features without the need for explicit support. In general, however, explicit supports are needed. The support structures are drawn out as thin wall sections which can easily be removed upon completion. Investigators at Rutgers University are also exploring the use of FDM with "green" ceramic wires.

**10. Ballistic Particle Manufacturing**

Ballistic Particle Manufacturing (BPM), which was just commercialized, was developed by BPM Technology, Inc. Since this system is so new, I do not have a complete understanding of it yet. However, I believe that the basic concept is accurately represented in this slide. BPM uses a piezoelectric jetting system to deposit microscopic particles of molten thermoplastic. Like FDM and SLA, support structures are required for "unconnected" features. In this case, the supports are deposited in a perforated pattern to facilitate removal. The BPM jet head, however, is mounted on a 5-axis positioning mechanism so that overhanging features can be deposited without support as is shown in this slide. I'm not certain whether the BPM system uses the 2-1/2D representation for planning, so I'm not sure of the best way to categorize this approach yet.

Other investigators are also exploring the use of 5-axis deposition to build up shapes. For example, Los Alamos National Labs is develpoing a 5-axis laser welding system called Directed Light Fabrication.

**Slide 11. 3D Layers**

Freeform fabricated shapes can also be built up with three dimensional (3D) cross-sectional layers (i.e., the outer surface of each layer maintains the 3D geometry of the original model). This approach eliminates the stair-step effect on the surface. A process planner for this type of approach requires the complete 3D geometry to generate the manufacturing plan, and the manufacturing plan will now depend on the part geometry. While the final building plan will be dependent upon the part's geometry, as I will show in subsequent slides, proper shape decomposition will assure that a successful building plan can be automatically synthesized. Also, note that the layer thickness will vary depending on the part geometry. One process which builds with 3D layers is Shape Deposition Manufacturing.

**Slide 12. Shape Deposition Manufacturing**

Shape Deposition Manufacturing (SDM) is being developed at Carnegie Mellon and Stanford universities. SDM integrates material deposition with material removal processes. Layer segments are deposited as near-net shapes and then machined to net-shape before additional material is deposited. The sequence for depositing the primary and support materials is dependent upon the local geometry and will be described in more detail in Slides 13 and 14. After deposition, each layer segment is then precisely shaped to net shape with a CNC cutting machine such as 5-axis milling.

SDM can use alternative deposition sources which I'll talk about later. The particular example depicted in this slide, 'microcasting', is a non-transferred welding process which deposits discrete, super-heated molten metal droplets for the purpose of creating fully dense, metallurgically bonded structures. For example, we deposit stainless-steel as the primary material and copper as the sacrificial material. The copper is etched away with nitric acid when the part is completed. Molten copper can be deposited on steel without destroying the previously shaped steel because copper has a lower melting point than steel. Conversely, steel can be deposited on copper, without destroying the previously shaped copper, since copper has such a high thermal conductivity (i.e., the steel droplets immediately solidify upon impact with copper).

**Slides 13 & 14. Adaptive Slice Decomposition, and Compact Splitting**

These slides represent the decomposition and basic building sequence for constructing 3D layered structures. The underlying idea is to decompose shapes into layer segments, or 'compacts', such that undercut features need not be machined, but are formed by previously shaped segments.

In general, any shape can be decomposed into layers that are characterized by one of three categories (as represented in the example on the left of Slide 13):

- Category 1 - The layer has no under-cut features (relative to the building direction), or
- Category 2 - the layer only has under-cut features, or
- Category 3 - the layer has both under-cut and non-undercut features.

Straight-wall features are considered either to be under-cut or non-undercut features depending upon subtle processing steps.

The sequence for building this particular structure with SDM is displayed in the remainder of these two slides. For layers in the first category, the primary material is first deposited (i..e., step 1 in right panel of first slide) and then machined (step 2). The support material is then deposited (step 3); then the entire layer surface is planed (step 4). For layers in the second category, the aforementioned sequence is reversed (see left panel of Slide 14).

Layers in the third category must be further decomposed into layer segments, or 'compacts', which are deposited and shaped in a sequence such that all under-cut features (of either the primary or support materials) are formed by the previously shaped non-undercut feature. For example, to build the third layer (depicted in right panel) a support material compact is first deposited and shaped. Then in step 2, the primary material is deposited; its undercut feature is formed by the preceding support structure compact, and its non-undercut feature is shaped by machining. In step 3, the final support material compact is deposited and similarly shaped.

For more complex geometries, a category 3 layer may have to be decomposed into even more compacts to satisfy the requirement that undercuts be formed be previously shaped material.

In general, since the thickness, the number of compacts, and the sequence for depositing and shaping the primary and support materials in each layer will vary based upon part geometry, we call this an "adaptive" decomposition approach.

**Slide 15. Heterogeneous Structures**

In addition to ease of planning and execution, a unique capability of incremental material deposition techniques is the ability to build heterogeneous structures. A heterogeneous structure might include multi-material regions and/or regions with varying microstructures and pre-fabricated devices embedded into the growing shapes. Another feature of a heterogeneous structure could be the addition of surfaces with micro-geometric textures such as has been demonstrated in the Three Dimensional Printing process.

In addition to geometry, process planners which could deal with these type of structures would also require additional information such as material, material properties and embedded device locations.

Heterogeneous structures would not be practical, or perhaps would be impossible, to fabricate with conventional forming techniques. The capability to fabricate heterogeneous structures is important because this capability can make the realization of complex designs both feasible and practical.

You can imagine using various embodiments of the various SFF processes which I have just described to build heterogeneous structures. I will use the SDM approach, which we have been developing, to demonstrate one way.

**Slide 16. Multiple Processes In SDM**

Shape Deposition Manufacturing (SDM) integrates multiple processing stations which operate on each layer. The growing parts are built on pallets which are transferred from station-to-station using a robotic palletizing system.

The deposition station has another robot that can select from several alternative deposition processes. To deposit each material, the robot picks up the appropriate deposition head and manipulates it over the growing shape. The thermal deposition sources include thermal spraying (e.g., plasma and electric arc), laser welding, several micocasters (e.g., one for copper and another for steel), and conventional welding. A hot wax dispenser is used to deposit complementary support material for building polyurethane (PU) structures; PU can be deposited as two-part epoxy systems.

With thermal deposition, internal residual stresses build up as each new layer is deposited due to differential contraction and thermal gradients between the freshly deposited molten material and the previously solidified layer. Internal stresses can lead to warpage and to delamination. In order to control the build up of stress, each layer can be shot-peened. At a shot-peening station, small round metal spheres (called 'shot') are projected at a high velocity against the current layer's top surface. Peening imparts a compressive load which counters the tensile load of the internal stress field; in essence, we are controlling the microstructure with peening.

Embedding is another intermediate operation which can take place in the SDM cycle. At appropriate layers, discrete components can be placed on top of the current upper surface before subsequent deposition takes place. After deposition, those components become permanently embedded within the structure. Embedding operations are currently done manually.

**Slide 17. Multi-material Structure**

Given the capability to deposit multi-materials, a strategy for building 3D multi-material layers is depicted in this slide. This four layer structure (shown on the left) is composed of three materials selectively distributed in several regions throughout the part (i.e., material #1 is the support structure, material #3 forms both the exterior surface and the interior region, and material #2 separates the interior from the exterior). While each material in each layer is individually deposited and shaped, each material may also have to be decomposed into smaller compacts to satisfy the constraints described in Slides 13 & 14. For example, the sequence for depositing the second layer of this structure is shown on the right side of this slide. Here, material #3 must be decomposed into two compact groups which are deposited and shaped at two separate steps: step 2 and step 4.

In the next three slides, I show examples of heterogeneous structures to demonstrate the potential importance of this capability.

**18. Heterogeneous Example: Instrumented, Mutli-material Tool For Forming Composites**

Consider the design of an autoclave lay-up tool for forming composites such as a Kevlar/fiberglass airplane wing. These tools must be pre-heated prior to the forming process and the absolute temperature and the uniformity of the temperature over the surface is critical during the forming process. A conventional tool can take several hours to preheat, thus increasing the manufacturing time; also, maintaining a uniform surface temperature is problematic, thus reducing reliability. The heterogeneous tool design in this slide addresses these problems in several ways. A heating/cooling channel which conforms to the tool's surface could be used to help preheat the tool. The channel would be formed using sacrificial material. The interior of the tool could be made of copper for fast and uniform heating, while the outside shell could be made of Invar to closely match the thermal expansion of the composite. Additional materials might also be required to form a functionally graded layer between the Invar and copper to compensate for thermal expansion mismatches. The tool's thermal mass could be minimized by a geometry which minimizes tool volume. And, arrays of embedded thermocouples would permit the tool's surface temperature to be monitored for process control.

**19. Heterogeneous Example: Conformal, Embedded Electronics Structure**

The embedded electronic device represented in this slide is another example of a heterogeneous structure design. It is a computer in which the circuitry is integrally housed within a package with a complex shaped form-factor. The housing might be composed of different materials; some for strength, others for lightness, and still others for toughness. The 3D circuit, which is formed by embedding conventional planar circuits interconnected with '3D vias', is distributed throughout the shape. This structure is compact and would be extremely rugged in harsh environments.

**20. Heterogeneous Example: Human Tissue Engineering**

In the future, SFF technologies will also be used in human tissue engineering applications. Tissue engineering is the process of creating laboratory synthesized substitutes to help replace or to restore damaged tissues. For one example, both the SLS and 3DP processes have been investigated as ways to build complex shaped hydroxyapatite scaffolds which could stimulate bone growth invivo when implanted. These scaffolds, however, are not living tissues. I believe that building complete tissue substitutes may be possible using SFF.

For example, a patient with a tumor that has locally invaded his or her jaw often requires reconstructive surgery after a large tumor resection. Imaging of the patient's head would be performed a few weeks prior to surgery. A 3D CAD model of the diseased mandible would be created based on these images and then modified to a desired healthy shape with the use of simulation to verify functionality. The mandibular computer model would then sent to an advanced CAM bioreactor to fabricate a living mandible as depicted in this slide.

What will this biological CAM system will look like? There is growing evidence that by simultaneously embedding cells, cellular proteins, vasculatur, as well as other growth and differentiation related substances into layers of synthetic or naturally occurring structural networks built with SFF, it may be possible to create complex tissues, synthetic tissue composites, and solid organ substitutes. A specialized support tissue matrix will also have to be simultaneously deposited to support the growth and differentiation of these synthetic tissue composites. By nurturing the layered tissue in a bioreactor capable of controlling multiple environmental parameters these heterogeneous structures may have the possibility of thriving in-vitro for later transplantation into humans in need of tissue substitutes.

**21. Distributed Manufacturing**

Finally, the reason for this workshop is to discuss the issues and possible solutions for creating CAD tools and design representations which will ensure that a product design can be successfully transmitted to an SFF service and built to some required specifications. The issues involve not only the creation of standards for information exchange, but also how to give feedback to designers about the manufacturing constraints and trade-offs of alternative SFF processes. The physical artifact may differ from the design model due to shrinkage, accuracy limitations, anisotropic mechanical properties, and rough surface finishes. These properties, in-turn, may be affected by the orientation in which the part is built and even the required building speed.

**Slide 1.** What is solid freeform fabrication?

"Solid freeform fabrication is the production of freeform solid objects directly from CAD models without part-specific tooling or human intervention." (Bourell, Beaman, Marcus, Barlow - 1990)

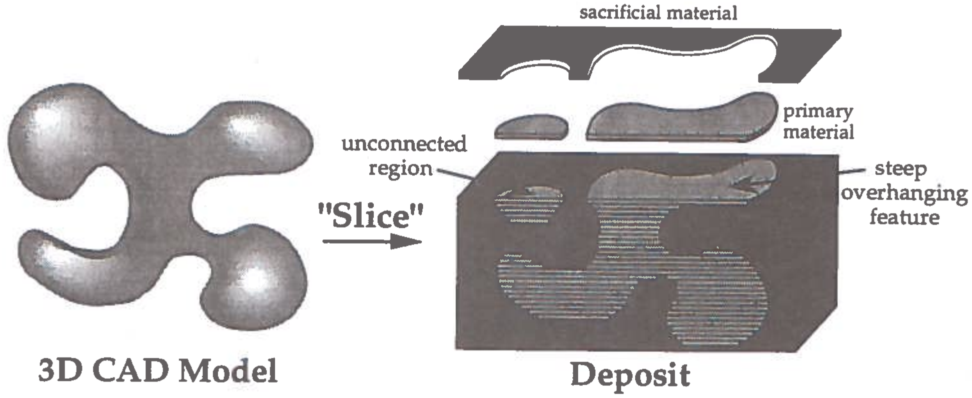
- ☛ automated fabrication of arbitrarily complex shaped parts
- ☛ rapid response
- ☛ "mass customization"
- ☛ enables complex designs

**Slide 2.** How is SFF realized?

Current practice:

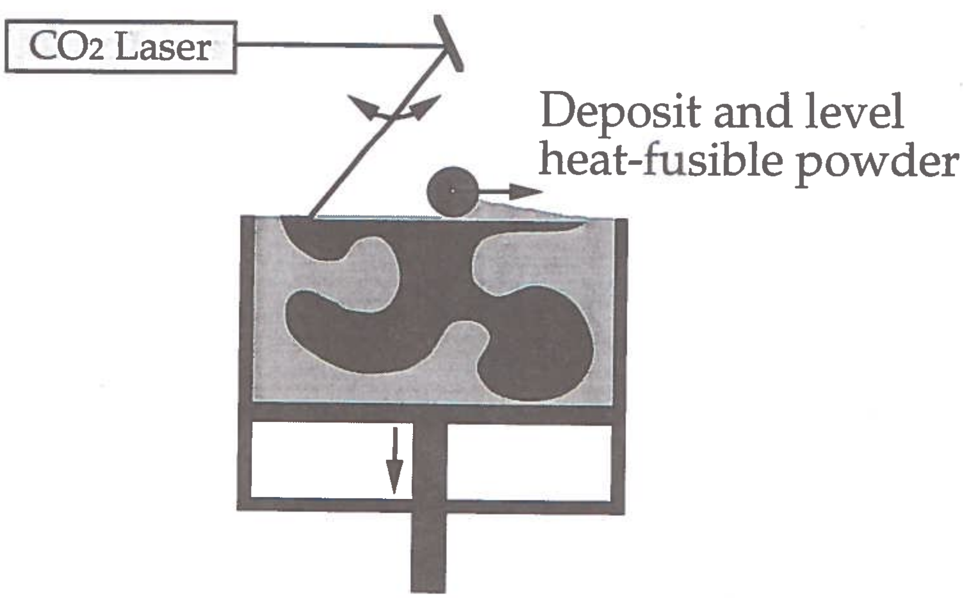
- ☛ Layered shape decomposition
- ☛ Selective material deposition
- ☛ Simultaneously deposit sacrificial support (or fixture)

## **Slide 3.** 2-1/2D layers, homogeneous stuctures with complementary support.

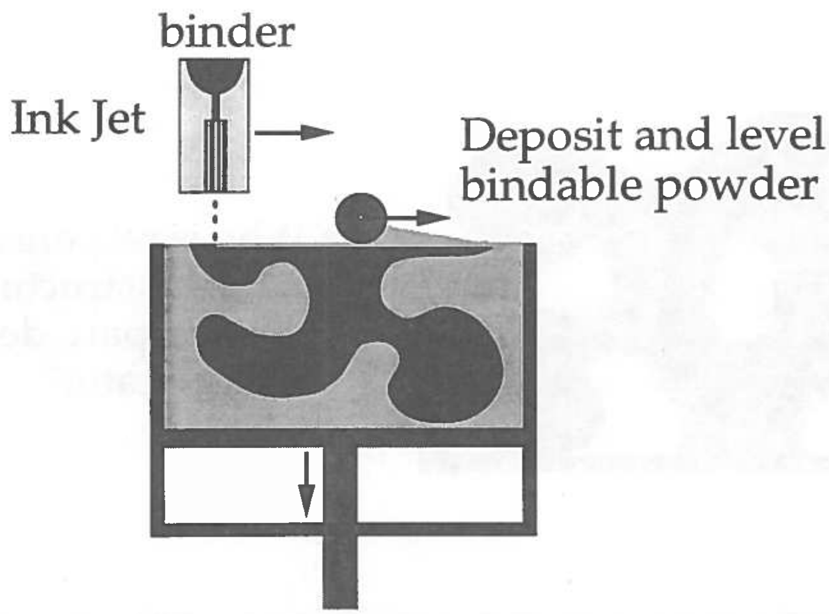



- Single material,
- Process planning and execution is independent of geometry, and
- Surface quality is a function of layer thickness.

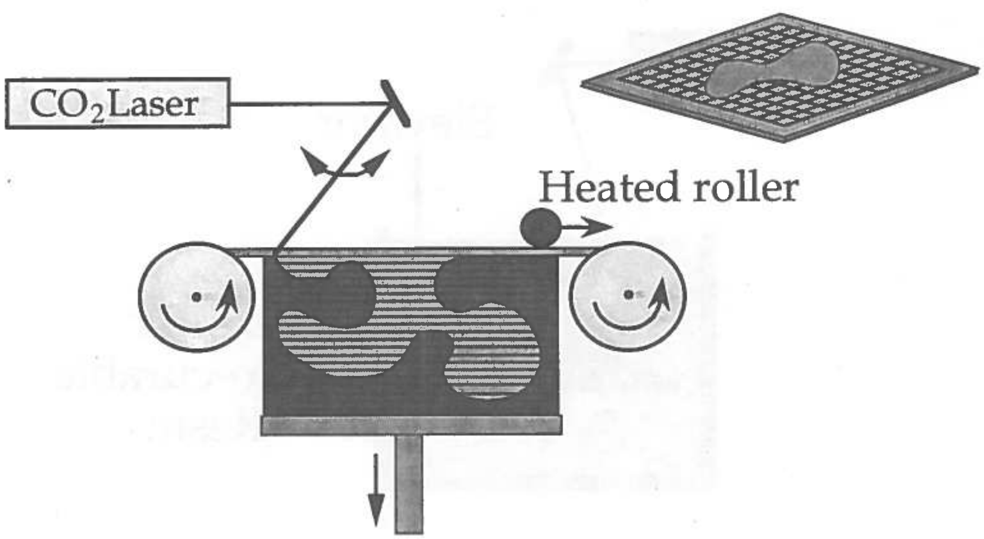
## **Slide 4.** Selective Laser Sintering



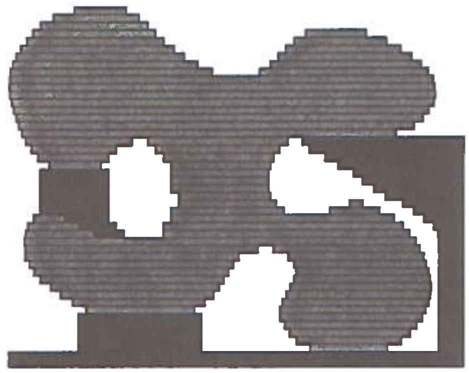
**Slide 5.** Three Dimensional Printing




**Slide 6.** Laminated Object Manufacturing

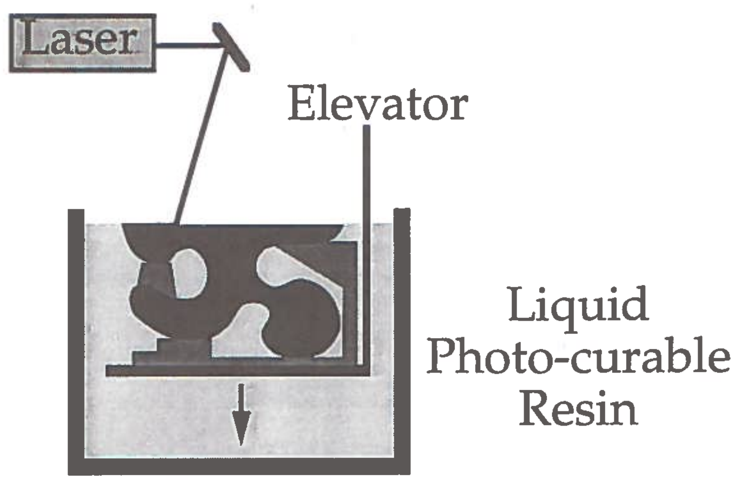


**Slide 7.** Explicit support structures

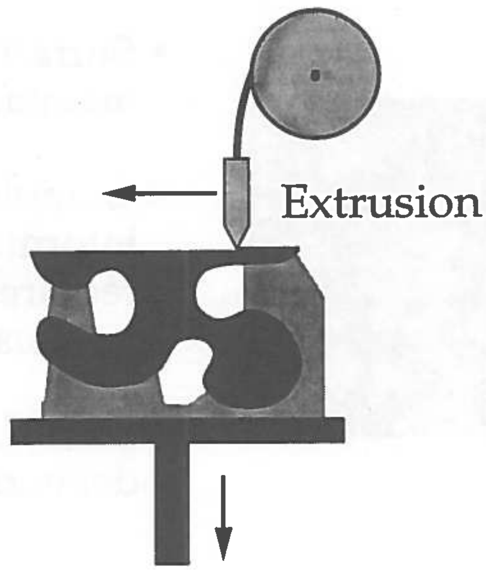


- Who is responsible for support structure design; part designer or fabricator?

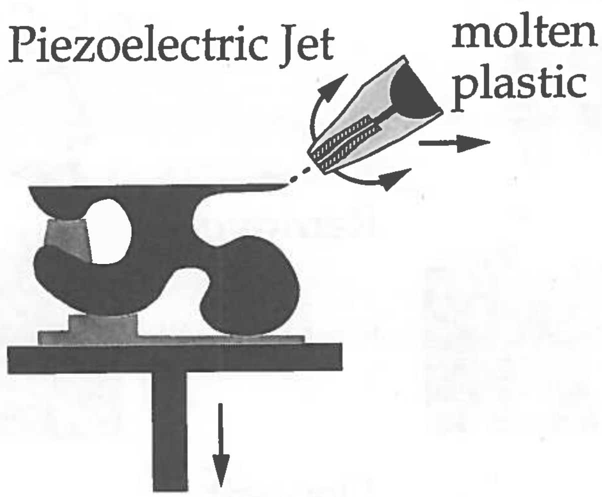
**Slide 8.** Stereolithography



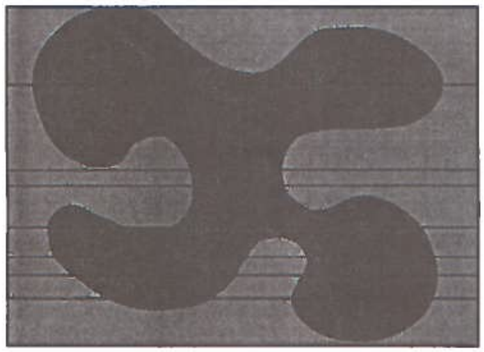
**Slide 9.** Fused Deposition Modeling




**Slide 10.** Ballistic Particle Manufacturing

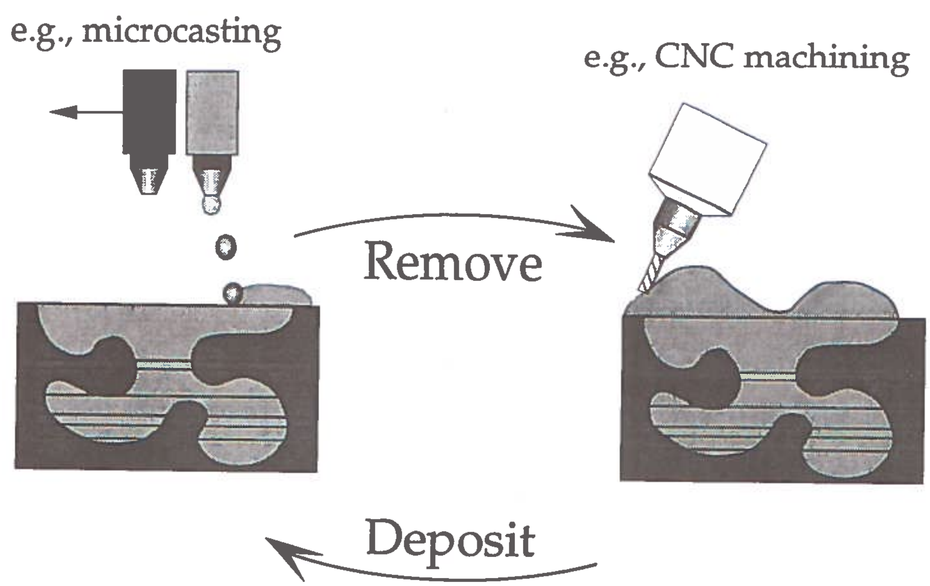


## **Slide 11.** 3D Layers

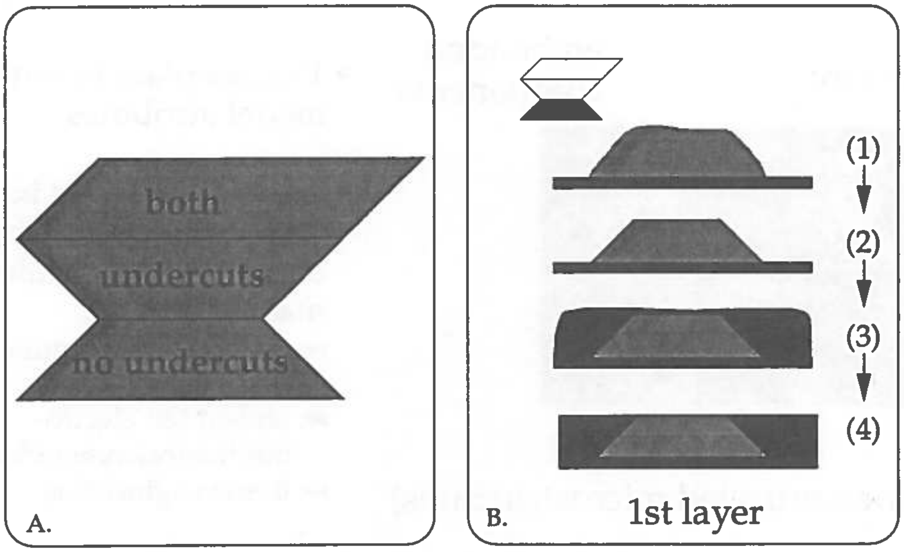


- Surface quality maintained
- Complete 3D information is required by process planner
- Manufacturing plan depends on geometry

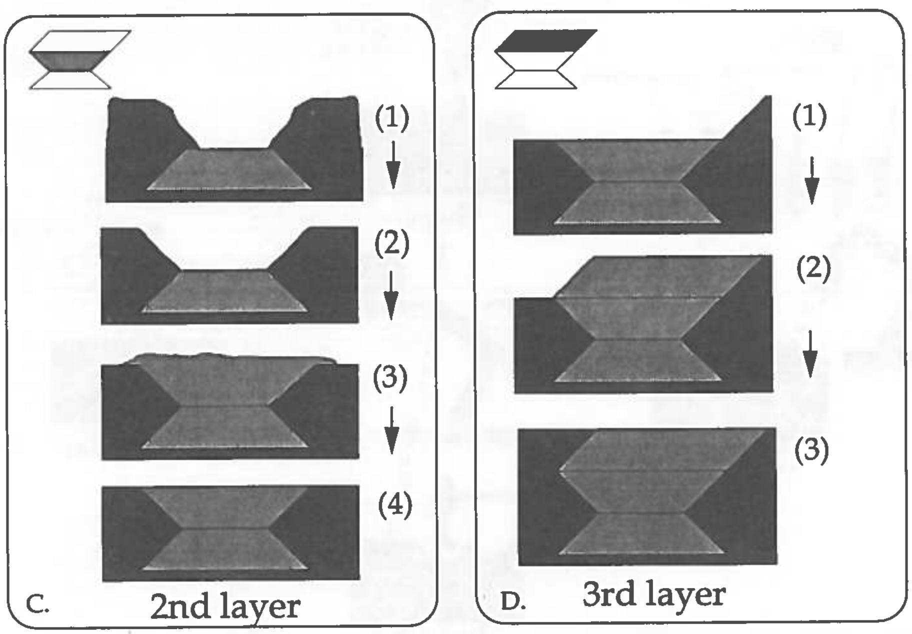
## **Slide 12.** Shape Deposition Manufacturing



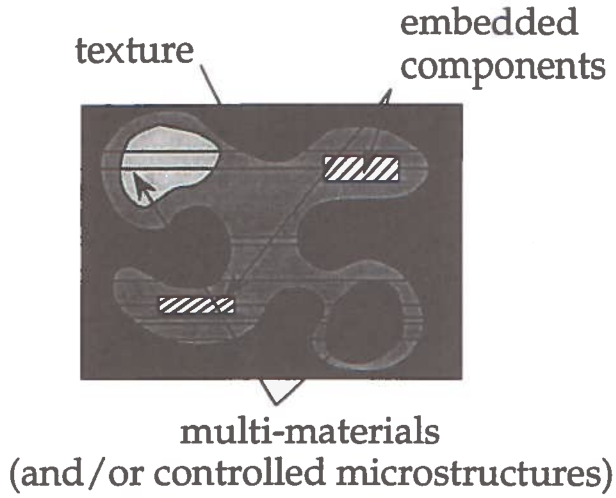
## Slide 13. Adaptive Slice Decomposition




## Slide 14. Compact Splitting

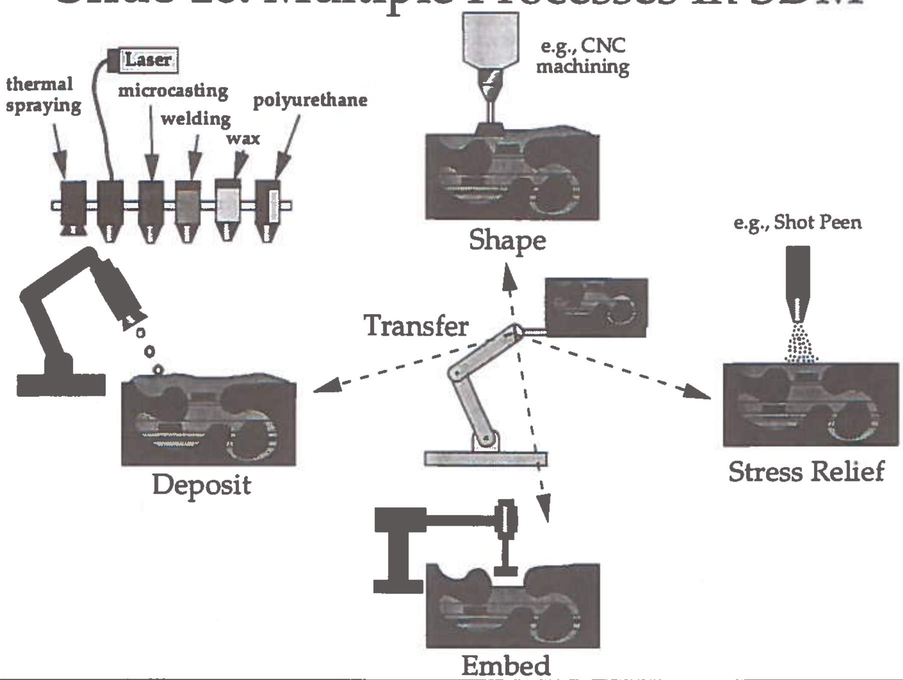


## **Slide 15.** Heterogeneous Structures

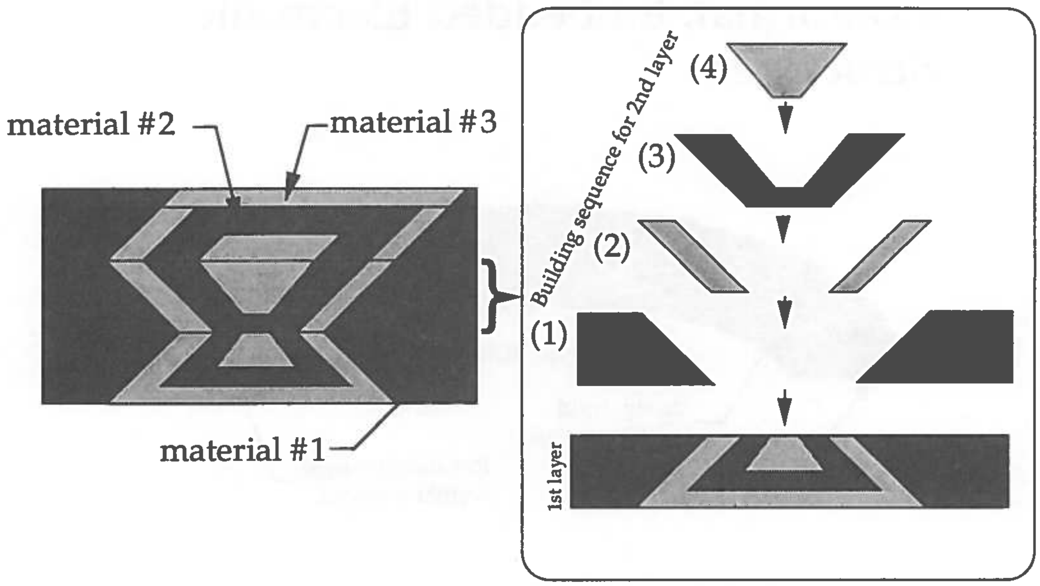



- Process planner requires model attributes
- Structures would be difficult or impossible to build with conventional manufacturing:
  - ☛ instrumented, multi-material tooling
  - ☛ embedded electro-mechanical assembles
  - ☛ tissue engineering

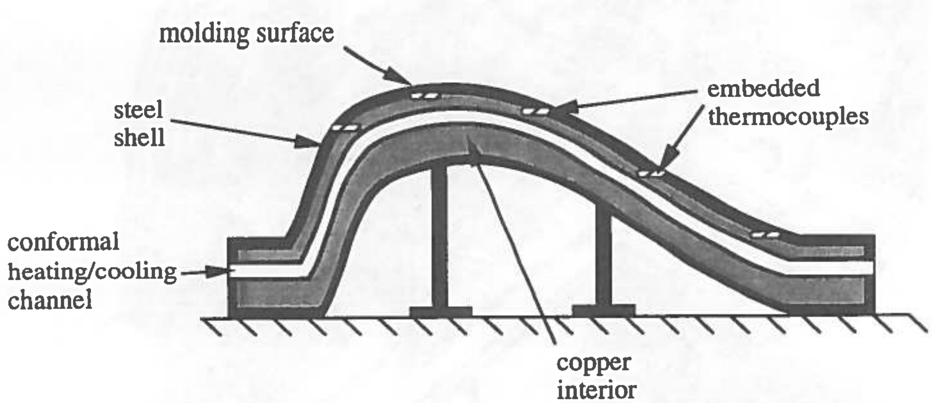
## **Slide 16.** Multiple Processes In SDM



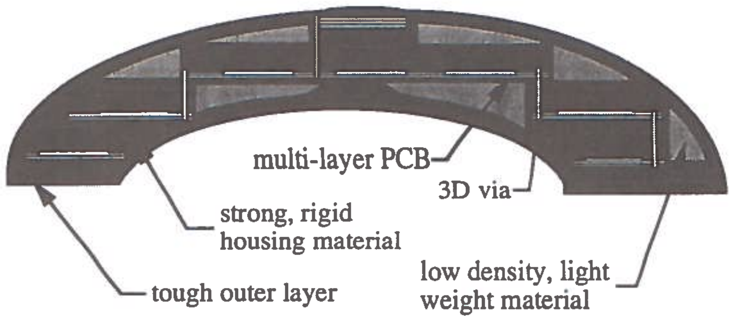
## **Slide 17.** Multi-material Stucture



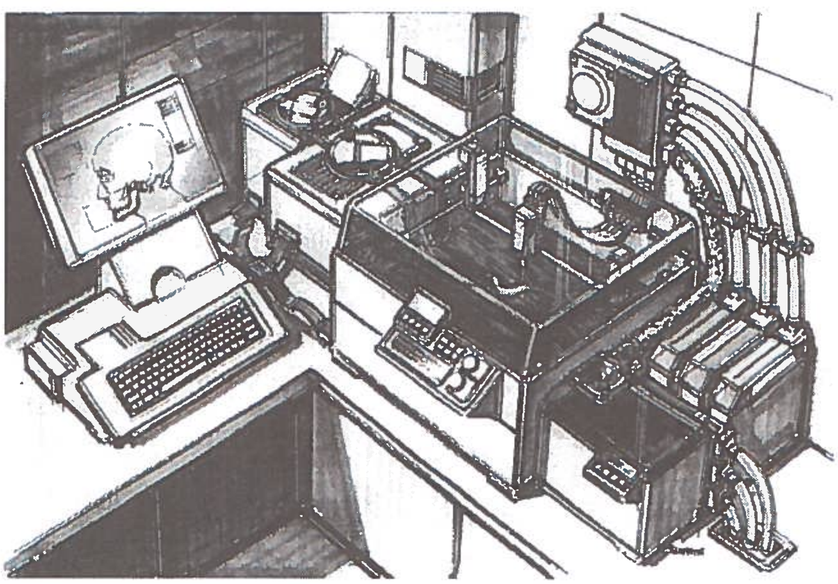

## **Slide 18.** Heterogeneous Example: Instrumented, Mutli-material Tool For Forming Composites



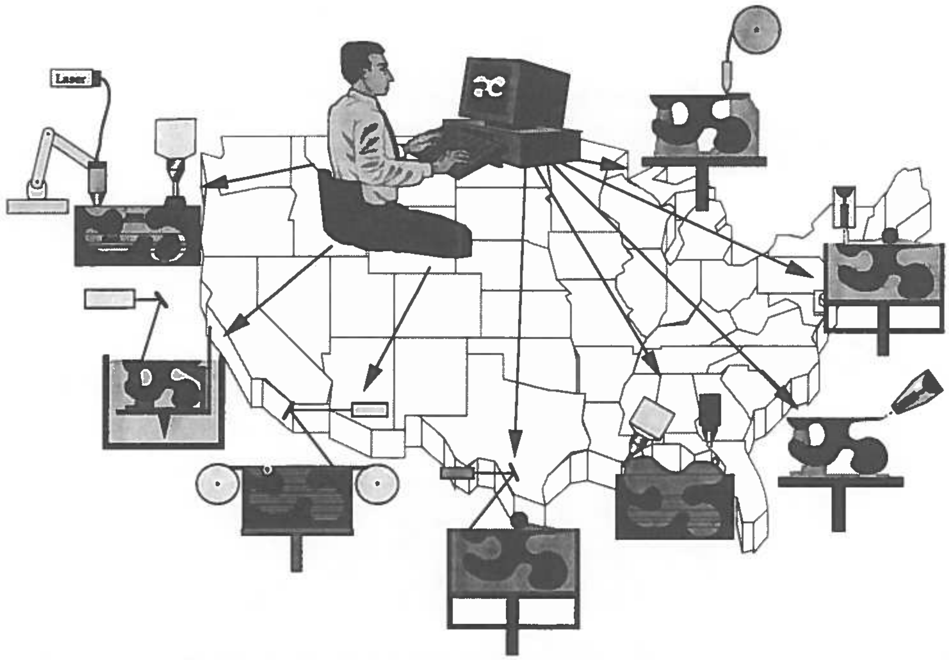
## **Slide 19.** Heterogeneous Example: Conformal, Embedded Electronic Structures




## **Slide 20.** Heterogeneous Example: Human Tissue Engineering



**Slide 21.** Distributed Manufacturing



# MODELING ISSUES IN SOLID FREE-FORM FABRICATION

Elaine Cohen, Sam Drake, Levent Gursoz*, Richard Riesenfeld
University of Utah

June 1995
*visiting from EDRC-CMU

## INTRODUCTION

In recent years, several Solid Free-form Fabrication (SFF) processes have emerged. Some of these processes are still evolving and new ones are still being developed. Although these processes share some common attributes, they fall under several distinct categories characterized by the nature of the underlying physical process. There have been several studies made in which relative merits of a wide variety of existing alternatives are discussed [1,2]. Below is just a sampling of current SFF processes.

- Stereo-lithography
  Successive cross-sections of the object are built in a vat containing a liquid polymer which is scanned with by a UV laser to effect solidification.

- Selective Laser Sintering
  Cross-sections are created by selectively sintering a layer of metallic powder under heat generated by laser. Common materials are plastic or wax powders as well as coated metallic powders.

- Laminate Object Manufacturing
  Layers are formed by cutting out the perimeter of the cross-sections in thin sheets which are bonded together.

- Fused Deposition Modeling
  Layers are built by dispensing hot thermoplastic filament with X-Y control over the cross-section.

- 3-Dimensional Printing
  Each layer is created by depositing a powdered material which then is selectively joined by the application of a liquid binder using ink-jet technology to build the cross-section.

- Shape Deposition
  Materials for each layer are deposited with microcasting and the contours are shaped with 5-axes CNC milling machine.

Despite the fact that all of these processes build objects one layer at a time, they represent substantially different approaches. This leads to considerable differences in the nature of modeling and planning for these processes. Furthermore, each process is applicable for different range of products and imparts different qualities on the produced objects.

In this position paper, we shall attempt to outline the common threads as well some of the differences in modeling for the existing range of SFF processes. We shall also discuss some issues in planning to the extent that they have impact on modeling and reverse.

## BACKGROUND

The SFF technology is still quite young and each different process is expected to mature in its own way. Nevertheless, the following characteristics, positive as well as negative, appear to be common to all SFF processes:

- High cost: although cost is expected to go down over time, many of the processes will remain expensive with respect to traditional manufacturing costs.

- Slow speed: all of the processes rely on gradual buildup of shapes and will likely be slower than conventional processes that can be optimized for high volume production. It is reasonable to expect, however, that as the new technologies mature and as there would be demand for higher throughput, there would be improvements in the processing speed.

- Flexibility: in principle, any given shape can be built within one process without part-specific fixturing and planning. Obviously practical considerations do impose some restrictions.

- Model driven: conceptually, a valid solid modeling description of the shape is sufficient to drive the planning and fabrication.

Even though SFF processes today are clearly too slow to be considered for high volume production, their usage is an appropriate choice for the following cases where single or very limited lot sizes are acceptable.

- Prototype fabrication: where the aim is essentially to have a sample, which renders the shape of a modeled object. An example for this or this is a prototype for a ski-pole grip where "feel" is critical to design. For this particular case, the material and geometric precision requirements are not very strict.

- Tool fabrication: where a single fabricated tool will be used in another process that achieves high volume production. Examples would be tools for injection molding or molds for investment casting. As contrast to the case above, the material and precision are often very stringent.

- Unconventional artifacts: where the same artifact cannot be manufactured using conventional manufacturing processes. Examples would be objects with embedded structures such as electronic or thermal components.

For the purpose of obtaining a typical sampling of the deployment of at least one SFF technology, we contacted a local service bureau to gather data on the utilization of the stereo-lithography process. The data exhibit the following break down in their orders: 75% one time prototypes, 15% patterns for Silicon Rapid Molding, 5% patterns for Investment Casting and 5% functional parts. It has been pointed out that the orders for Investment Casting are poised to grow the most. The same data indicate that 60% of the orders are placed with STL files and the balance is submitted as 2-D drafting or 3-D wire-frame representation for which the service bureau creates solid models using commercial CAD systems to create STL files. It was indicated that the data was communicated to them mostly via modem connection until recently, but increasingly the internet has been the medium of transfer. These figures bear out the expected breakdown.

## REPRESENTATIONAL REQUIREMENTS

The most essential representational requirement in planning for solid free-form fabrication is the solid modeling information. In fact, quite arguably, one of the reasons for the recent move towards a wider adoption of solid modeling in product representation has been the potential offered by the emerging solid free-form fabrication processes.

In addition to making it necessary to have an underlying solid modeling representation, the demands imposed by the SFF process planning on the solid modeling system are much more stringent than for conventional processes. In terms of just the representation, planning effort for SFF requires completeness and integrity in solid models. In terms of analysis and operations to be performed on those representations, a much higher degree of robustness and reliability is necessary. When solid objects are decomposed into a multitude of layers, frequency of basic analysis operations increase greatly and so does the likelihood of failure. These operations usually involve geometric degeneracies. This tends to be be more of a problem than in many traditional processes since it becomes impossible to intervene in the details of the planning actions as they become intuitively non-obvious from a higher level. One cannot expect to correct mistakes either in modeling or in geometric computations during the manufacturing.

At the other extreme with respect to a solid modeling representation, it may be necessary to accommodate data originating from some form of digital sampling. In medical imaging, for instance, the representation is often acquired in the form of bitmaps corresponding to a series of cross-sections. It would be natural to directly use this information in modeling and planning if that would be appropriate. This makes it desirable to have the possibility of accessing the low level planning phase of the processes.

For a number of SFF processes, the representational requirements demand a more enhanced form of geometric modeling. Particularly for those fabrication processes capable of fabricating objects with multiple materials and embedded electronics, a dimensionally nonhomogeneous geometric representation becomes vital to capture internal structures.

Since different SFF processes exhibit varying shortcomings with respect to such criteria as surface finish, strength, precision, etc., the modeling framework should accommodate such specifications to assess the viability of a given SFF alternative. To support such reasoning it is logical to consider a representation that supports features. It should be noted, however, the mapping between the design and the manufacturing features may be weaker than in the case of conventional manufacturing.

## PROCESS PLANNING ISSUES

Since the layer based fabrication approach is intrinsically anisotropic, it becomes important to optimize the build-up direction. Depending on the nature of the process, the direction may be optimized with the appropriate criteria. This may be viewed as a preliminary planning activity. Although to what extent this planning phase can be generalized for the range of available SFF processes is unclear, a set of modules customized to the individual processes can be constructed. A number of SFF processes require support structures to be built along with the object. It is important to optimize the buildup direction with respect to the scaffolding requirements and to synthesize the necessary support structures.

Obviously, the primary planning activity is the generation of layers from a solid modeling representation. Depending on the nature of the process, it may not be sufficient to merely generate planar cross-sections. Processes that individually shape the contours in the orthogonal direction require surface normal information at layer contours. In fact, a preceding analysis of the surface normals may be necessary to determine the location of the layers. Despite the fact that such variations may need to be accommodated for different processes, layer generation is common to all SFF processes.

What follows the layer generation seems to fall into fundamentally two categories. In some processes, like stereo-lithography, process actions span the cross-sectional area. In other processes, like the material deposition, the geometry is achieved by the shaping actions on the perimeter of the cross-section. In this case, the process planning is rather similar to that for CNC machining.

Another issue that needs to be considered is the overall size of the objects to be manufactured. Most of the SFF processes proposed so far impose limitations on the size of parts that can be manufactured. Although in prototype fabrication, scaling down object size is often acceptable, functional parts will have to be as large as specified. Quite possibly, a way around this would be to decompose the objects into components which are separately fabricated and combined later. This then becomes another aspect of the process planning that needs to be incorporated into the overall planning module. One also has to bear in mind that whether such an approach would be feasible for a given part, say, a windmill blade.

In conventional manufacturing, depending on the features exhibited (and also depending on the batch size), a variety of manufacturing processes are employed to fabricate a given product. By their nature, SFF techniques typically would be the primary process and, in those cases where such fabricated parts are not amenable to subsequent post processing, the only process used in manufacturing. Since the available SFF processes have different relative weaknesses and strengths with respect to the shape and material characteristics, a preliminary analysis of the design with respect to both becomes necessary to judge the viability of a particular SFF process. For example, Laminate Object Manufacturing is less suitable for thin-walled objects and stereo-lithography may not be an appropriate choice when dimensional precision is required.

## CONCLUSIONS

SFF processes do offer the potential for fabricating unconventional products with an unconventional approach. It is clear that departing from the conventional in both of these aspects generate interesting results. As it has been recognized, this class of processes can fabricate complex shapes with intricate multi-material compositions. This particular aspect is clearly an advantage over conventionally manufactured parts which are either of complex shape or of complex material but rarely both at the same time. With the adopted unconventional approach, SFF processes can and must be driven by a solid representation. Provided that the process planning issues are sufficiently resolved, the potential for a VLSI-like manufacturing paradigm for mechanical parts does exist, at least for a suitable category of objects.

It is not expected, however, that the difficulties in process planning for the manufacturing of 3-dimensional parts are necessarily going to disappear with the adoption of free-form fabrication techniques. Although the difficulties in the areas of fixturing and tool access are greatly alleviated, in those processes capable of fabricating functional parts (of intricate shapes and sophisticated material quality), those gains are offset by more complex planning requirements.

One can say that SFF processes pose a challenge for modeling. It is clear that the present practice of modeling does not sufficiently support the design of parts that are meant to be manufactured by SFF processes. It is also the case that the modeling environments today do not adequately facilitate the planning activity.

## ACKNOWLEDGEMENTS

This work was supported in part by N00014-92-J-4113. All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

## REFERENCES

1. Aubin, Richard F., "A World Wide Assessment of Rapid Prototyping Technologies," Proceedings of SFF Symposium, Austin, Texas, August 1994, pp 118-145
2. Schmidt, L. D., "A benchmarking Comparison of Commercial Techniques in Rapid Prototyping," Proceedings of 5th Int. Conference on Rapid Prototyping, Dayton, Ohio, June 1994, pp 333-351

# The Industrial/Mechanical Design Process
# Design of Wearable Computer Systems

Engineering Design Research Center Carnegie Mellon University

Dan Siewiorek, Director
Chris Kasabach, Industrial Designer
John Stivoric, Industrial Designer

## Introduction

This paper defines the role of industrial and mechanical designers in the wearable computer team at Carnegie Mellon's Engineering Design Research Center (EDRC). Through a detailed outline, the paper examines the designers' process when creating complex body-worn computer systems. It extends from the day word is passed about a potential client to the final production of working wearable computers.

There are two issues that are not explicit within the outline that are important to understanding the process and should be included in this preface. First, the industrial and mechanical designers' role in the EDRC is not stand alone. Projects are developed concurrently with teams of software, electrical, and human computer interaction (HCI) designers. Our timelines and decisions are completely linked, and we meet continuously throughout the projects.

Secondly, the word 'ideation' is mentioned many times in the outline. Ideation means concept generation; whether it is purely thinking about how the product will fit the body or look, or drawing those thoughts on paper, or building those thoughts in the form of 3D model. The timeline, because it is set up in discrete sections, may imply that ideation stops and starts a bunch of times. It doesn't. Ideation occurs all the way through the project, simply fluctuating in intensity, type and level of detail. The problem statement and our technology research are the same way. They are continuously examined and updated as the market releases new products and we better understand our client's needs. Only when a product must be specified for production is a design fully solidified.

Following the Outline is a Conclusion that helps summarize the process. It is followed by an Observations section that addresses the question blocks posed by the workshop.

## The Industrial/Mechanical Design Process for the Design of Wearable Computers.

### Conceptual Design

**The Project Begins (Ideation phase 1)** - The wearable computer team receives word we will be working with a client to solve problems that wearable computing can believably solve. This starts a first phase of general ideation characterized by thoughts about where the computer could be worn on the body and how it might be interacted with (speech recognition, a dial, gesturing?). The problem is not really understood yet but this 'blue-sky' ideation phase, free from practical constraints, occurs anyway.

A wearable computer is worn on the body and moves with the user. It provides real-time, or immediate, information over the user's workspace, giving the user a detailed understanding of their task. Wearable computers thus remove the traditional need for oversized blueprints or volumes of manuals to use as references for construction and maintenance information.

**Meet the Client** - We meet with the prospective client and get an understanding of the problem. Part of this meeting is a warming up of our team to theirs and vice-versa, and establishing who needs to talk to who in the future and how communication should take place. At this point we also evaluate the appropriateness of a wearable computer for solving the problem.

> **Task Analysis** - While with the client we try and understand how their business currently does work. If asked to design a computerized maintenance assistant for mechanics, we observe mechanics in their environments, notice their clothing, the tools they use, sounds in the environment, and the terminology they communicate with. We also shoot video, take photographs and bring documents from the site back to our team. This enables everyone to visualize the problem and helps yield better and faster solutions.
>
> **Understanding the System** - Still with the client, we take time to learn the entire process the computer will be fitting into. For example, one of our clients want a computer for aircraft maintenance, which includes inspection tasks, troubleshooting of problems, repairs, etc.. By understanding the events that occur around these activities, such as the scheduling of aircraft for inspections, the planning procedure for repairs, and work shift hours, we can integrate the computer more appropriately into the overall culture of the business.
>
> (In the first meeting with the client, ideas are sometimes impulsively proposed concerning what the system could be. Someone might say: "Maybe there's a "clipboard" type monitor you hang beneath the plane and you have a small wearable computer that powers it, with a ...." This is a form of ideation.)

**Creating a Problem Statement or Statements** - From our meeting(s) with the client the team (including the client) works to develop a problem statement that we can look back on and match our solutions to.

**Ideation Phase 2** - A good amount of sketching now occurs concerning product placement, and also the perception of the product. Metaphors are often created. During our last project there was the idea of a computer being like a pillow that rests in the small of the back. This gets at issues of comfort and appearance.

**Technology Survey** - While doing ideation we start to in-depthly examine available technologies for each subsystem. The electronic design team, for example, researches hardware components, power options and connectors. Software examines a range of operating systems and may also research speech recognition systems and radio options. The industrial and mechanical design team explores available input technologies, manufacturing processes, materials and thermal management products.

**Technology Evaluation Matrices** - For each technology researched in the Technology Survey we create an evaluation matrix that lets us choose the best option from a range of competing products. For example, when designing a computer housing, one matrix created is a materials matrix. It includes products like nylon, urethane, aluminum, aluminum alloys, and various plastics. These products are compared in areas such as durability, cost, molding potential, surface quality and color availability — each weighted to reflect its importance in the design. The material(s) with the overall highest marks becomes our primary pursuit.

**Creating a Product Definition** - With the best technologies chosen and purchased, we now understand exactly how large, complex, costly and heavy the system will be, as well as how much heat it will generate. This helps guide where to locate the computer on the body and the material it should be made from. For example, a system weighing 6 lb. that generates a lot of heat can not be worn comfortably on a belt clip for an 8-hr. work shift. Nor can it be made of a plastic with a low melting temperature. The product definition includes the creation of a hardware system diagram that shows all components necessary for the system to operate (number and size of batteries, number and sizes of cables and connectors...) and how the components must be connected (see Appendix 1, sample Hardware System Diagram).

**Ideation Phase 3** - This phase consists largely of configuration studies, that is, taking the system diagram components and trying them in various layouts. This is done using CAD software like Ashlar Vellum to ensure correct component sizes and relationships. In the configuration studies special consideration is given to the placement of heat producing components, components that need to be accessed by the user such as batteries or memory cards, minimization of the computer's overall dimensions, and convenience of production and assembly (among others).

**Form Studies** - A second ideation activity that occurs in this phase is the creation of form studies. Here the industrial and mechanical designers try and create a visual character for the product. Ideation includes thumbnail sketches, rough clay, foam or wire models, bringing in pictures of other products or objects that embody the feeling we want to capture, and discussing color, material and texture issues. There is a lot of brainstorming in this phase and everyone is responsible for generating a wide range of concepts.

**Configuration Evaluation Matrices** - Once a large number of configurations have been explored, an evaluation matrix is created to compare the configuration concepts against one another. Again criteria is weighted by importance in the design. Criteria such as comfort and simplicity — two of our most important considerations — receive very high weights. Other criteria include complexity of wire management both within the processor housing and outside (to peripheral devices such as the display), and ease of assembly. The configuration that receives the highest marks becomes our primary direction for development. However, if other designs seem meritable they are also pursued.

**Form Evaluation Matrices** - Around the same time that matrices are created for our configurations, so are evaluation matrices created for our Form Studies. Evaluation criteria include comparing the forms' overall perceptual aspects — perception of simplicity, intrigue, comfort, durability, power and beauty — size, production simplicity, comfort, durability and ease of use.

**Getting Physical** - The most meritable component configurations are built from closed-cell foam, fiber board and wood using conventional power tools such as band saws and lathes. This lets us physically see (and show others) what the shape and dimensions of the computer's inside will be. By the end of this phase the entire team has a common vision of the product: the problem is understood, configurations are physically in front of us, and because we have searched out and tested a wide range of technologies, configurations, and forms, we can solidly defend the path we are on.

### Design Planning

**Division of Labor** - With an understanding of the product comes the ability to focus. The industrial and mechanical design team divides within itself, and each member is given a part of the subsystem to design and test. In our last product this meant designating a member to focus on the processor or main unit design, another member to center on thermal management solutions, two members on the input device and two members working on the harness that connects everything. It is important to note that despite focusing, the group continues to give input in each area, and there is still a collaborative effort going on. However, now a group member or members is responsible for interpreting and carrying out a design in each area.

### Design Realization

**Ideation Phase 4** - Members in each area begin to explore concepts through both orthographic (flat) and perspective drawings, as well as models made from materials like foam, wood, plastic or canvas. Members are asked to create at least three possible directions that their area of focus might take. Again, this visual exploration helps the entire team visualize the potential of the overall system. When the potential seems high enough, these models often inspire the software and electronic groups to put forth greater efforts so as to achieve a more complete product.

**Group Evaluation** - The designs generated by this ideation phase test each potential technology and/or configuration solution. Each design is evaluated by the subsystem and group members on the following criteria: form, usability, how well it integrates a solution technology/configuration, how flexible it is to integrate and use with other subsystems, how simple it is both on its own and with the system, and manufacturabililty. These criteria help assess which specific technological solution or hybrid of solutions will best meet the team and user's product specification. Results of the evaluations are passed around to the overall wearable computer design team for confirmation.

**Further Design Iteration and Detailing** - With evaluation complete, members of each subsystem area are challenged to take the most promising direction and explore at least three variations on it. Particular importance is placed on developing form and function issues in parallel. Three dimensional models become the primary mode of communication now and objects are mocked-up more and more accurately in terms of form, material, moving parts, construction and weight.

**Subsystem Evaluation** - A formal evaluation process again takes place with each subsystem part. This time a definitive final form and configuration must be decided upon so that detailed design can be produced of each subsystem. Once the evaluations are completed, the subsystem is again presented to the entire team to confirm the direction of the final design. Again, what is important here is maintaining a common vision of the system amongst the team members.

A final system design is specified and each subsystem identified to detail for implementation

**Detailed Design** - Each subsystem group (main unit, input device, harness) creates a rough mock-up of the final design to use as a reference when detailing mechanical drawings of the product. Detailed CAD drawings of the design are developed, with dimensions. All details need to be refined and resolved, including how their internal components fit into the form, how the product is assembled, the placement of connectors and screws, materials, texture, and color. The CAD drawings are verified by a final 'physical' mock-up and the design is presented to the team members to make sure it is still consistent with the specifications of other subsystem criteria. Any changes are noted on the model and, if necessary, CAD drawings revised.

The drawing files and the model are then handed to a mechanical engineer who will build a 3D CAD model of the product in ProEngineer. For example, the model of the main unit would include all parts, internal components, connectors, and our thermal management solution. (The engineer will refine any details as necessary and test the assembly within the 3D CAD environment.)

While refining such details, the engineer may send a preliminary CAD file for translation into a stereolithography (SLA) model. This SLA model represents the exterior form and confirms our dimensional, visual, and structural details.

In specific products, the CAD drawings are sometimes handed over to an electrical engineer in order to collaboratively design and build a non-working printed circuit board(s) to be populated by actual components. This board can test both the assembly of the board(s) and the physical fit of the components into a realistic model of the product.

**Gearing up for Rapid Prototyping** - The 3D CAD file is then translated and sent via network to an off-site prototyping house for final SLA production. Once made, the parts are cleaned and assembled into the detailed model. The model is then sent back to the design team for evaluation. The electronic printed circuit board is also sent back to the team to evaluate fit and component placement of the electrical components of a product.

The stereolithography model is then evaluated against the vision that the team shares of what the final product should be. This point in the process is the first time a subsystem is in its most real form. Fit, internal configurations, assembly, and can be physically visualized and tested.

Now the 3D CAD model is refined to represent the decided upon final details. Fit and assembly are again tested within the CAD environment. The refinements necessary for the electronic printed circuit board are also noted and communicated to the electronic engineering team for integration into the final electronic board(s) design.

The 3D CAD model is again sent out to be realized in stereo-lithography. This time the model is of greater quality and finish, so as to allow us to use each part as a pattern for low- production, rapid prototyping molds.

Once we receive the parts, a final non-working prototype can be assembled and verified, ready for final production.

## Implementation of the Design

**Rapid Prototyping** - After detailed design is complete, the Industrial/Mechanical Design Team implement their subsystem using the available rapid prototyping technologies and production methods.

Now, rapid prototyping methods are chosen based on cost, time, and quality expectations. Possible prototype companies are identified based on these criteria. Each company is asked to examine CAD files of the part and provide the team with prices and turnaround times for one as well as multiple pieces.

Based on these results and our comfort level with the people we speak with, the companies are chosen. CAD files and SLA models are sent out across the nation to the prototype companies that will produce finished, durable parts in our chosen materials.

(Whether machining or casting parts, we have found that prototype companies find physical models to be important references for product planning, scale, and finish possibilities. The models directly assist in the interpretation of the CAD, lessening confusion of our needs and expectations, and ultimately expediting the delivery of complete and accurate parts.)

**Verification** - As companies complete their first prototypes, we ask to receive them so as to verify materials, durability, accuracy, finish, and color. More than likely, the first part does not meet expectations and the team must communicate back to the companies any changes before the full run is implemented. The initial parts are often sent back with changes written and drawn directly on the part, again for immediate visualization and to lessen the risk of misinterpretation of the necessary changes.

Once the final part requirements are verified and agreed upon, the production run begins. (When casting in urethane, we can generally produce upwards of 30 parts before a mold begins to break down.)

**Post Production** - Upon receiving the lot of parts from each company, the necessary post-production processes are applied. (For cast urethane parts, this sometimes means sanding, cleaning up thru-holes, or putting in threaded inserts.)

**Detailing Final Parts** - Once post-production is complete, each part is detailed. (Detailing may include spraying EMI shielding, color and/or texture to a part, anodizing or coating aluminum, and silk-screening graphics onto the parts.)

### System Integration

**Assembling the System -** Once all the parts have been detailed, the final assembly and integration of the system begins. Electronic parts are assembled, prepared for placement, and attached inside their respective parts. Each product is fully assembled and the necessary software installed.

**Testing the System -** When each subsystem is complete and working, each product is connected to each other and the system is tested. Any minor changes in the parts, electronics, or software are tweaked and debugged to insure the system will remain operational.

**Delivering the System -** Finally, a specified number of systems are completed and presented to our client(s).

### System Evaluation

A user group to test the system in the field on an application is defined and the procedure designed so that information can be gathered in a consistent and complete manner. A number of complete systems are delivered to the application site to be tested by potential users. The information gathered at the site is analyzed and the results made explicit back to the design team. From this information the product can be evaluated and decisions made on what changes should be implemented in the next iteration.

From this information, a new wish list from the user's can also be started and integrated into the next generation problem and product definitions

In addition to the system evaluation, the design process is also evaluated. Shortcomings of the system may be a result of bottlenecks that occurred during the process. These problem areas need to be identified so that the next generation process design can plan for it better.

Outside of the product shortcomings, there are also communication and methodology problems, that may not have impacted the success of the final system design, but did impact the tight integration of the team process. These areas must also be identified so that the methodology can continue to evolve into a more efficient, and non-taxing process

## Conclusion

The Industrial and Mechanical Design Group, above all, maintains a role, throughout the process, of helping the team physically visualize the product everyone is designing. Without a common understanding or vision between design groups, each would be forced to rely on their own set of assumptions and criteria based on only a single view of the product.

Even though the process has been described linearly, it is a dynamic set of steps, that overlap into one another, and where many steps are revisited later in the process. Each step that has been defined simply initiates a procedure that will continue with different intensities and interpretation along the entire process timeline.

The process outlined by the Industrial and Mechanical Design Group occurs concurrently with the process timelines of the Electronic Design, Software Design, and Human-Computer Interface Design Groups. Each group follows a similar methodology, while meeting and reviewing each other's work regularly, keeping the team tightly integrated to assist in developing and delivering a complete, well thought out, functional, and usable wearable computer system.

Wearable computer design is a strongly iterative process. Design must be an evolutionary, responsive process that accommodates new experiences, knowledge, and situations. The vision of the product begins very general and undefined, yet with each step, the design spirals in toward more focused and detailed directions. Technologies are constantly changing and the team must stay informed and flexible throughout the process, prepared to change directions toward a more integrated and simplified solution at all times, yet also being able to maintain a tight time schedule. Under these conditions, the ability to rapidly iterate and prototype — no matter what the medium — is essential. However, getting the right product out the door on time cannot be achieved just by getting the technology right; we must also produce the right product. Getting the right product required feedback from all the participants — including team members, sponsors, and users — throughout the design process.

With each generation of wearable computer, we have seen the demand for decreased product development times and increased product functionality. Systems continue to become more and more complex, requiring tighter integration between team members, and the need to delay final production as late as possible so that as many issues can be worked out as possible. The more time we can squeeze into the early stages — delaying production to the last possible moment - the more resolved the product will be, both formally and functionally. In addition, the more iterations that can occur on a product design, the more detailed the product can become, and more opportunities for participant feedback will exist.

### Observations (In Direct Response to the Workshop Issues)

- Whether the physical design should be described in layers or 3D is difficult to answer. Perhaps at some point in the design process the design should be capable of being described in the manufacturing process it will be made in; if stereolithography, this would mean layers, if pressure forming this would mean a continuous sheet representation, deformed and stretched to the shape of the housing.

- Creating in 3D is a critical part of the wearable design process. Traditional solid modeling allows us to evaluate weight, comfort and perception issues, and material characteristics very early in the process with minimum cost investment. Furthermore, we can bring these models directly into the context of the human body as well as the user's environment.

  Having a physical model to experiment with also leads to serendipity, or 'happy accidents'. For example, our team once gave someone a crude 3D computer model designed for the back. The person put it on upside down and we realized it was actually more comfortable that way.

- In terms of attributes that a model should provide, the ones mentioned (strength, material, and tolerance) are important. Other ideas might be weight (have a database of various component weights), weight distribution (a feature that calculates the product's center of gravity), and attributes that ask the user about environmental characteristics. For example, the system could make material recommendations for waterproofing, fire retardance, etc... Another attribute might be a fit test that ensures all seams, thru-holes, components and contours that are supposed to line up, do. The system could also inspect wall thickness and dimensions for fit and calculate shrinkage for various production processes.

- Thoughts on a design exchange format: It would be nice if software was compatible between electrical and industrial/mechanical design. We could share ideas easier and ensure internal and external features are being designed compatibly.

  It would also be nice if contractors and suppliers could continuously dump their new products and processes into a database that the designer's expert system had direct access to. The expert system could then provide state-of the-art recommendations on components, material and process selection.

### Additional Observations

- Frequently, the time savings in rapid prototyping can be challenged by their necessary post-production processes. Even though rapid prototyping methods are becoming more fluent at producing complex parts, these secondary operations many times threaten issues of quality, precision, cost, and ultimately time.

- In addition, many times the rapid prototyping method does not transfer well to production methods. many times another iterative design process is involved in transferring the prototype designs into production parts. This requires additional investments is time and money, delaying the time to market and profit gains. Therefore, it can be imagined that a CAD system, throughout the product's development provide expert advice about the end production needs, continually, up front within the rapid prototyping, and even conceptual, stages. This advising system may even offer alternative design directions and/or production methods to pursue (with regards to cost, time, and defined expected quality). And as the design becomes more specific, those decisions would be fed into the database to allow the system to provide more detailed and appropriate advice. The final outcome of such a process is a prototype whose CAD files can be used directly by the manufacturer, without any new manipulations, directly producing molds form the data. This type of activity streamlines the process for the manufacturer, and also confirms for the client, what they see (in prototype form) is what they get with minimal reinvestment in process time (the cost occurs only in the of mold production).

- The design process is very much team oriented. Communication between team members and different disciplines remains a challenge. Each group has its own set of language and design representations (some more visual or physical than others). If a system could unite all the different groups, yet still maintain the unique representation levels, changes could be integrated into the product design much more efficiently and accurately. For instance, as soon as a electrical engineer changes the placement of a component, the change would occur immediately in a 3D CAD file of the board configuration (a different form of representation, yet necessary for mechanical design work). The system would be tied into how certain changes affect other team parameters and appear in the necessary locations, noting whom and what such a change will affect, informing these participants, and possibly advising how to proceed. This would also require for instance that electronic component CAD documentation also carry mechanical and dimensional specifications to be tied into other representations, etc.

  In addition, as moves are made over time, the trail of changes and decisions are also captured, making post process evaluation more accessible, and providing a clearer idea of a part, and therefore product history, justifying decisions, and identifying problems to be remedied in future generations, of both product and process

## 4.6 Position Papers

# Design Automation for MEMS
# A White Paper

May 25, 1995

Prof. Erik K. Antonsson, Ph.D., P.E.
Engineering Design Research Laboratory
Division of Engineering and Applied Science
California Institute of Technology
Pasadena, CA 91125
erik@design.caltech.edu

We are presently beginning the development of design automation methods for Micro-Electro/Mechanical Systems (MEMS) with the goal of automatic mask-layout synthesis for a desired final etched shape. While considerable progress has been made in the areas of etch simulation [4, 6, 7, 8, 18], finite element analysis [3, 5, 11, 14, 16, 17, 22], corner compensation [1, 13, 15], and design [10, 19, 20, 21] the fabrication of MEMS has been made without the benefit of design automation techniques. In contrast, the design of VLSI systems has become highly formalized and automated. One of the goals of Mead and Conway's early work in the VLSI area was to permit "ordinary engineers" to perform design [12]. Prior to their work, VLSI design was the exclusive domain of highly trained and experienced specialists. Other engineering domains (e.g., MEMS design) have not had the benefit of the same level of formalism and automation in design, and engineering design in these areas remains the province of highly trained and experienced specialists.[1]

Our objective is to continue development of MEMS Engineering Design infrastructure recently begun in several research groups (including our own) to permit rapid, accurate, conservative mask-layout synthesis of MEMS in a way analogous to present-day VLSI design. Our long-range objective is to enable a MEMS designer to specify a desired micro-mechanical function (e.g., a mechanical spring with particular characteristics), and have a system automatically generate the information (mask-layout, and other fabrication instructions) to create the shape that exhibits the desired function. This new approach will mean that MEMS designers will be able to concentrate on the desired function of the device, rather than the details of its physical manifestation.

Our goal is:

- To develop a MEMS mask-layout synthesis program that will automatically create a mask-layout for a given desired final 3-D shape.

Prior MEMS CAD research has focused on the analysis of function for a given shape. For example, S. Senturia's group at MIT has developed a highly sophisticated finite

[1] Here the term "formal" is used to mean computable, in the sense that a design process can be automated. There are many methods that are in daily use in mechanical engineering design, but are insufficiently formal to permit automation of the design process. "Systematized" is a commonly used synonym to "formal" as the term is used here.

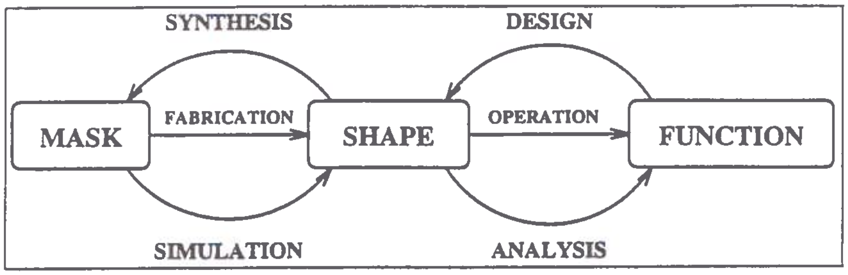



Figure 1: Simulation vs. Synthesis, Design vs. Analysis

element analysis of stress and charge on micro-silicon shapes [17]. K. Wise's group at the University of Michigan has explored the design of micro-mechanisms [10, 20, 21]. R. Buser et al. [3, 4] have developed a useful analysis tool called ASEP (Anisotropic Silicon Etching Program) which can predict the output shape based on traveling planes. C. Sequin at Berkeley has introduced the Slowness Method [18].

The primary method for creating a MEMS mask-layout today is trial-and-error, guided by experience. Consequently, many iterations, and hence many prototypes, are typically required to develop a mask-layout that results in the desired shape and desired function. As Brysek, Petersen, and McCulley recently observed (1994):

> "In-depth knowledge of the [fabrication] process is needed because in micro-machining 'what you see' is often NOT 'what you get'." [2, Page 25]

An illustration of this point is shown in Figure 2. This will be particularly true for future MEMS systems which will involve many degrees of freedom and/or complicated 3-dimensional shapes.

Because of the geometric complexity of surface fabricated MEMS devices, the present MEMS design procedure can be characterized as a *mask-to-shape-to-function* process. Even though the designer may start with a function and shape in mind, the complexity of the fabrication process forces the design cycle to iterate around the mask-to-shape-to-function evaluation process, as shown by the bottom arrows in Figure 1. However, the desired approach is exactly the reverse: *function-to-shape-to-mask*. That is, the designer conceives of a MEMS function, then through an automated (but perhaps iterative) process determines a shape that will exhibit the desired function, as shown by the top arrows in Figure 1. For example, the designer can develop a tentative shape, and then use FEA methods to iteratively refine the shape until it exhibits the desired characteristics. Next, using an automated (but again perhaps iterative) process, the shape description is used to determine a mask-layout and a set of fabrication instructions that will create the desired shape, or the best possible approximation to that shape. In the case that the desired shape can not be fabricated, the designer may again need to use FEA tools to evaluate the suitability of the best approximate shape. For this reason, a standard

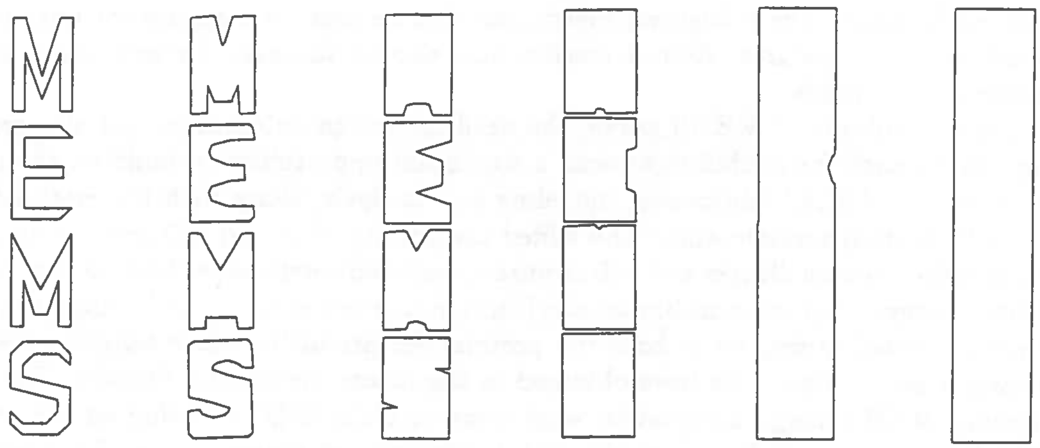


Figure 2: Input mask shape (on the left) and changing shape with time; anisotropic etchant simulation.

communication format for the transfer of information between different levels of MEMS design is proposed.

Our goal is to *complement* current FEA (and related MEMS CAD) approaches so as to include the complexities of typical MEMS fabrication processes into the design cycle. Current MEMS FEA methods focus only on the relationship between function and (3-D) shape. Our goal is to understand the relationship between (3-D) shape and mask-layout.

To develop formal and computable methods for the "shape-to-mask" process, a more exact computational model of the MEMS fabrication process is required. Such models will form the basis for the *forward* or *simulation* problem. That is, the solution to the forward problem determines what shape results from a given mask and a given etching process. More importantly, the model for the forward process is necessarily the basis for the *inverse* or *mask-layout synthesis* problem–i.e., determine the mask shape that yields a desired processed shape for a given etchant. One result that can commonly occur, is that no such mask shape exists. In this case, a *shape approximation metric* will need to be applied to determine the closest shape (or perhaps the closest function).

A closely related problem is that of the optimally "robust" shape. Even if a desired shape can be produced by an idealized fabrication process, deviations in the processed shape from the desired shape will occur due to process variations, small errors in mask alignment, errors in etch rate diagram data, non-ideal effects, and finite mask resolution. One might alternatively define a *robustness metric* or a *sensitivity metric* for a shape. That is, how likely it is that the desired shape will be obtained (or obtained within an acceptable tolerance), assuming an expected range of processing errors? Analogously, how sensitive is a given shape to processing variations? Such metrics are useful for many applications. In the case that more than one mask shape will lead to the same processed shape under ideal conditions, the robustness metric could be used to select the most robust mask shape. The robustness metric can also be used to compare the output of different design procedures. Robust metrics may also be the basis for procedures that estimate process yields.

As the complexity of MEMS grows, the need for design automation will also grow. Design automation for MEMS represents a significant opportunity to build on the pioneering work in MEMS fabrication, modeling and analysis, along with the established work in VLSI design automation. The added complexity of 2- and 3-D mechanical devices introduce new challenges and will require considerable extensions beyond the VLSI domain. However, the inherent limitations (limited number of materials, limited shapes and sizes, limited forces, etc.) hold the promise for producing more tangible design automation results than have been obtained in the macro-mechanical domain. The introductory MEMS design automation work done at Caltech [9], building on the prior MEMS fabrication, modeling, analysis, and design work of others, is one of the initial steps towards the goals outlined here.

## References

[1] Abu-Zeid, M. Corner undercutting in anisotropically etched isolation contours. *J. Electrochem. Soc. 131* (1984), 2138–2142.

[2] Bryzek, J., Petersen, K. E., and McCulley, W. Micromachines made of silicon. *IEEE Spectrum 31*, 5 (May 1994), 20–31.

[3] Buser, R. A., Crary, S. B., and Juma, O. S. Integration of the anisotropic-silicon-etching program ASEP within the CAEMEMS CAD/CAE framework. In *MEMS 92* (Institute of Electrical Engineers, 1992), pp. 133–138.

[4] Buser, R. A., and de Rooij, N. F. ASEP: A CAD program for silicon anisotropic etching. *Sensors and Actuators 28* (1991), 71–78.

[5] Crary, S. B., and Zhang, Y. CAEMEMS: An integrated computer-aided engineering workbench for micro-electro mechanical systems. In *MEMS 90* (Institute of Electrical Engineers, 1990), pp. 113–115.

[6] DeLapierre, G. Anisotropic crystal etching: A simulation program. *Sensors and Actuators 31* (1992), 267–274.

[7] Frank, F. C., and Ives, M. B. Orientation-dependent dissolution of Germanium. *J. Appl. Phys. 31*, 11 (Nov. 1960), 1996–1999.

[8] Fruhauf, J., Trautman, K., Wittig, J., and Zeilke, D. A simulation tool for orientation dependent etching. *J. of Micromechanics and Microengineering 3* (1993), 113–115.

[9] HUBBARD, T. J., AND ANTONSSON, E. K. Emergent Faces in Crystal Etching. *Journal of Microelectomechanical Systems 3*, 1 (Mar. 1994), 19–28.

[10] KOTA, S., ANANTHASURESH, G. K., CRARY, S. B., AND WISE, K. D. Design and fabrication of microelectromechanical systems. *ASME Journal of Mechanical Design 116*, 4 (Dec. 1994), 1081–1088.

[11] MASEEH, F., HARRIS, R., AND SENTURIA, S. A CAD architecture for MEMS. In *Transducers 90* (Institute of Electrical Engineers, 1990), pp. 44–49.

[12] MEAD, C., AND CONWAY, L. *Introduction to VLSI Systems.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

[13] OFFERIENS, H. L., KUHL, K., AND SANDMAIER, H. Methods for the fabrication of convex corners in anisotropic etching of (100) silicon in aqueous KOH. *Sensors and Actuators A25-27* (1991), 9–13.

[14] POURAHMADI, F., AND TWERDOK, J. Modeling micromachined sensors with finite elements. *Machine Design* (July 1990), 44–60.

[15] PUERS, B., AND SANSEN, W. Compensation structures for convex corner micromachining in silicon. *Sensors and Actuators A21-23* (1990), 1036–1041.

[16] SCHWARZENBACH, H. U., KORVINK, J. G., ROOS, M., SARTORIS, G., AND ANDERHEGGEN, E. A micro electro mechanical CAD extension for SESES. *J. of Micromechanics and Microengineering 3* (1993), 118–122.

[17] SENTURIA, S. D., HARRIS, R. M., JOHNSON, B. P., KIM, S., SHULMAN, M. A., AND WHITE, J. K. A computer-aided design system for microelectromechanical systems (MEMCAD). *Journal of Microelectomechanical Systems 1* (Mar. 1992), 3–13.

[18] SEQUIN, C. H. Computer simulation of anisotropic crystal etching. In *Transducers '91* (San Francisco, CA, USA, 1991), pp. 801–806.

[19] TRIMMER, W. S., AND GABRIEL, K. J. Design considerations for a practical electrostatic micro-motor. *Sensors and Actuators 11* (1987), 189–206.

[20] WISE, K. D. Integrated microelectromechanical systems: A perspective on MEMS in the 90s. In *MEMS '91* (Institute of Electrical Engineers, 1991), pp. 33–27.

[21] WISE, K. D., AND NAFAJI, K. Microfabrication techniques for integrated sensors and microsystems. *Science 25* (Nov. 1991), 1335–1342.

[22] ZHANG, Y., CRARY, S. B., AND WISE, K. D. Pressure sensor design and simulation using the CAEMEMS-D module. In *Tech. Dig. IEEE Solid-State Sensor and Actuator Workshop* (Institute of Electrical Engineers, 1990), pp. 32–35.

# REQUIREMENTS OF FUTURE FABRICATOR DATA FORMATS

**Preliminary participant's paper for**
**Workshop on Solid Freeform Fabrication**
**Carnegie Mellon University**
**June 5-6, 1995**

**Marshall Burns**
**Ennex Fabrication Technologies**
**Los Angeles, California**
**(310) 824-5185, marshall@ennex.com**

The most important characteristic of a fabricator data format is flexibility. Because if there is one type of data that I can guarantee will be needed in the year 2000, it is the data type "unknown in 1995."

## REQUIREMENTS TODAY

In 1995, a data format capable of communicating to all the fabricators on the market today needs to convey the following classes of information.

-- PERIPHERAL GEOMETRY. Either (a) a set of closed surfaces representing the peripheral shape of the object to be built, or (b) a sequence of a set of closed curves representing the peripheral shape of flat slices of the object. If the geometry is given in three dimensions, the user may be able to specify an orientation of the object in space so as to determine the slice planes.

-- HATCH. Fill pattern within the slices, if applicable.

Other issues, in particular the fabrication material, are determined on a case by case basis by the operator of the fabricator. Usually these issues are decided on the basis of the needs of the technology, with little regard available to the preferences of the creator of the design file.

## REQUIREMENTS TOMORROW

We can enumerate many elements of information that may be worth conveying to a future fabricator. Our challenge is to conceive a data format which is efficient for today's primitive devices, yet is capable of maturing naturally to handle this kind of complexity.

-- MATERIAL WITHIN A REGION. Aside from just stating what material is present, we must allow for various levels of detail in the representation. For example, it may be sufficient to specify that an entire region be filled with fiberglass. In other cases, we may need to specify the curved contours within which the fibers should lay. In the most demanding cases, however, there may be a concern for specifying the precise path of each individual fiber.

-- ABSENCE OF MATERIAL. This should be included in the specification of material as the specification, "no material." But the format should allow this to be done efficiently, so that all the bits needed to specify a complex material are not wasted on an empty region.

-- COLOR. Properly, this is a subset of the specification of material, because pigmentation is a physical characteristic of a material. But since color can be achieved by mixing of base colors, one may not want the data to specify the voxel-level distribution of pigmentation, but only specify the net hue and saturation. (If we can specify a luminescent material, we may also want to specify intensity! This is an example of why the format has to be flexible.) The distribution of pigmentation would then be calculated by the fabricator according to its own capabilities to reproduce the specified color.

-- MATERIAL PROPERTIES, such as strength, electrical conductivity, index of refraction, melting temperature, etc. As in the case of color, these properties can also be determined at the voxel level by a detailed specification of materials, but it may be more efficient to treat them at a higher level. Also, perhaps we would rather specify only material properties, and we don't care what materials are used as long as these properties are achieved.

-- VARIATIONS OF PROPERTIES. We may wish to specify inhomogeneous and/or nonisotropic properties. For example, we may have certain strength requirements along the axis of a shaft, while the transverse strength is less important.

-- LOCATION OF REGION. In 2-D raster scanning, the location of a pixel is implicit in the logic of the scan. But in vector scanning, the instructions must tell the hardware where to go. We need this capability in 3-D fabrication.

-- SIZE AND SHAPE OF REGION. In some cases, we may wish to specify the fabrication voxel-by-voxel. But this will be wasteful in large regions with some element of uniformity. So we should be able to describe a closed surface and specify the material and/or properties to fill it with

-- TEXTURE AND PATTERNS. Without specifying the detailed geometry, we should be able to specify that a surface is stippled or that a region is foamed (i.e., permeated with bubbles of a certain size and frequency). A precursor of this is found in today's hatch patterns and in perforated interfaces between the primary object and support structures.

-- TOLERANCE. We should specify how accurately the geometrical and mechanical parameters listed above have to be reproduced in solid material, such as by specifying acceptable error ranges.

-- TEMPORAL ORDER. All of today's fabricators build in sequences of flat layers. When we do not have this restriction, we may wish to specify the order in which certain regions are fabricated.

-- ALTERNATIVES. In porting data from one fabricator to another, there will be cases where the second machine will not have all the capabilities of the first. Rather than allowing the second machine to substitute what it thinks is the best alternative material (or worse, causing the second machine to crash), we may want to include some instructions analogous to "if you don't have ABS, use nylon," etc.

-- COST AND TIME. In a fabricator capable of working in multiple materials and various processes, the cost of material used and the fabrication time may vary widely for any given region being fabricated. We may want to tell the fabricator not to spend more than a certain amount of time and/or money on a certain region.

-- PRIORITY. Many times, the fabricator will not be able to meet all the requirements we specify. Rather than have the process fail, we may wish to set relative priorities for some of the requirements. For example we might want to say something like, "Flange A must have an ultimate tensile strength of 50 MPa, whatever it takes, but for flange B, which is reproduced 300 times in this design, make it as strong as you can without spending more than $0.02 and 3 seconds on each one."

-- FLUID COMPONENTS. If the fabricator is capable of making foamed materials and other configurations that involve closed cavities, the question arises of what is inside those cavities. Is it air, vacuum, or some liquid of a specified density and other properties?

-- SMART MATERIALS. At the fringes of what is imaginable today, what happens when we start to fabricate in smart materials, so that geometry and other properties are not fixed and constant? Future users designing robots will need to specify not only the static material strength of limbs, but also the active strength of joints (e.g., ability to lift a certain amount of weight). Today we think of thermal properties in terms of melting temperatures and heat capacities, but future designers will design homeostatic systems in which they want to specify the actual temperature of individual components.

-- BI-DIRECTIONAL COMMUNICATION. Another future capability to consider is that of feedback between the fabricator and the transmitting computer. Future fabricators will monitor their progress and be able to report problems back to the computer. In this scenario, fabrication will be a real-time, interactive process and the data format must be bi-directional.

## A POSSIBLE STRATEGY: HANDSHAKING PROTOCOL

It may be impractical to design a single data format capable of growing from meeting today's requirements to tomorrow's. Perhaps a preferred approach would be a handshaking protocol that allows for the implementation of a variety of formats, with the transmitting computer and fabricator beginning their conversation by agreeing to a format they can communicate in. This would be similar to the procedure used today between two modems or fax machines to establish a compatible baud rate and other parameters of communication. So, for an example in terms of today's formats, the computer might say to the fabricator, "I would like to send you a series of SLC slices, but if you are not equipped to interpret that, I can send you the data in StL."

# Data Interchange for Solid Freeform Fabrication

**Richard H. Crawford**
**Joseph J. Beaman**
**Department of Mechanical Engineering**
**The University of Texas at Austin**
**Austin, TX**

Solid Freeform Fabrication (SFF) is a group of emerging technologies for fabricating physical objects directly from computer-based descriptions (such as solid models) of the geometry of the parts. Originally used for creating models for visualization, many industrial users of SFF technologies are realizing the greater potential of SFF as legitimate manufacturing processes for producing patterns and, in some cases, functional parts. Thus, SFF is becoming an important aspect of the product realization process in these industries.

Most Solid Freeform Fabrication processes produce parts on a layer-by-layer basis. The process proceeds by first slicing the geometric description of the part into layers. The slicing operation generates the contours of the part for each layer. The contours are then processed in a manner dependent upon the particular SFF technology. For instance, for selective laser sintering, under development at The University of Texas, the contours are discretized into Rtoggle pointsS at which the laser beam must be modulated to produce the desired solid.

The layer-based nature of SFF invites an exploration of similarities to VLSI manufacturing technology, which is also layer-based. In particular, the VLSI industry has benefited from establishing a common data interchange format that allows separation of the logical and physical design of VLSI devices. This representation allows design to proceed somewhat independently from fabrication considerations. One naturally wonders whether a similar standard can be established for the SFF industry. In this paper, we briefly explore this question. The paper consists of a description of representations in common use today, a short discussion of requirements on the geometric description, and a critique of how each of the common representations would serve as a standard with respect to the requirements.

## Forms of Geometric Input

Solid freeform fabrication was motivated by the desire for Rpush-buttonS prototyping, in which solid reproductions of three-dimensional geometric models are created automatically under computer control. Central to this process is presentation of the part geometry to the process. Three common forms of presenting geometric information to SFF processes are described below, along with their relative advantages and disadvantages.

**Faceted Geometry.** The SFF industry has a *de facto* standard, the so-called RSTLS file format established by 3D Systems, Inc., for exchanging geometric information about a part to be manufactured. This format is generated by tessellating the surfaces of the geometric model into a mesh of non-overlapping triangular facets. This format has been adopted by many CAD vendors, is readily available, and is considered adequate for most visualization applications. However, for producing accurate patterns and functional parts, the accuracy of the STL format is unclear. In many cases, the tessellation operation itself introduces errors in the model. Tessellation of surfaces with large curvature can result in errors at the intersections between such surfaces, leaving gaps or RholesS along edges of the part model. Tessellation of fine features is susceptible to round-off error, which leads to non-manifold models of parts, where more than two facets are adjacent to a single edge, or facets with opposing outer normals meet at a single vertex. These problems are difficult to handle for slicing algorithms.

**Higher Order Geometric Descriptions.** The problems with the STL geometry exchange format arise because tessellation is a first-order approximation of more complex geometric entities. An obvious solution to these problems is to exchange higher order geometric entities, preferably the source geometry with which a part is designed. Processing higher order geometry for SFF processes offers several advantages over exchange of faceted geometric descriptions. Generally speaking, files containing higher order information will be smaller and more accurate than comparable faceted geometry files. Also, many of the problems that result in non-manifold geometric information in faceted descriptions can be avoided. Instead, it is incumbent upon the SFF geometry processor to ensure that the results are realizable for the particular SFF technology that is used to fabricate the part. Potential problems in the slicing operation can be solved because more information is available about the intended geometry of the part; thus, higher order descriptions are easier to troubleshoot when necessary. Finally, when approximations are necessary for the given input geometry, the approximation process is driven by the particular SFF technology rather than by generic criteria meant to satisfy the requirements of many SFF technologies. This provides a rational basis for approximating the geometry when necessary.

There are disadvantages to higher order geometric data exchange as well. First, there is no single geometry form that is satisfactory for all applications. There are many different geometric descriptions that are used in product design, each with different requirements for a slicing algorithm. Designers of commercial SFF processing software will have to make compromise decisions about which geometric forms to support or risk losing potential customers from lack of geometric coverage. Also, because the geometric input is more complex, algorithms for processing the geometry are more complex as well.

**Layered Geometry.** Many potential applications of SFF naturally provide data in layer-based formats, including data from laser digitizers and medical imaging data from computed tomography (CT) and magnetic resonance imaging. While these technologies seem to mesh naturally with SFF, there are problems in using their data directly. Typically the layer thicknesses from such sources are not compatible with SFF technologies. Producing accurate models from layered data requires interpolating between slices to realize the resolution necessary for the SFF process. Also, CT data consist essentially of raster images which provide a measure of the relative density of the material at each pixel in the imaging plane. However, SFF processes require boundary information rather than interior density data. Raster images are converted to contour images by specifying a minimum threshold density below which the data are ignored. Typically, the threshold value is selected manually by a skilled radiology technician. However, to realize the full potential of fabricating such models with SFF, threshold values must be determined automatically. Algorithms for determining optimal threshold values are needed for preprocessing layer-based SFF input.

## Requirements of Geometric Descriptions

While process planning for SFF is considerably reduced compared to conventional fabrication technologies, there are several considerations which will require reasoning about the geometry of the final part. Scaling and orientation of the part within the work space of the SFF machine have a significant impact on the efficiency of the process. Aside from other factors, the part should be oriented in a manner which minimizes the number of layers. Other factors, however, may override this consideration. For instance, tolerances tend to be directionally dependent. Likewise, the mechanical properties of the final part will depend upon its orientation during the process. These issues require geometric reasoning on both global and local scales. Global reasoning will indicate the best part orientation within the workspace of the SFF machine. Local reasoning refers to considerations of the geometry of each layer to determine scanning and build patterns that maximize geometric accuracy and mechanical properties of the part. Geometric descriptions for SFF must support both local and global reasoning.

The layer-based nature of SFF inherently introduces errors in surface accuracy. Curved surfaces perpendicular to the build plane are actually fabricated as a series of discrete linear approximations in the direction of the build. For most SFF technologies, all layers have the same thickness for a given part, regardless of the local geometry. One way to improve surface accuracy is to slice the part adaptively, generating layers of varying thickness based on some geometric error measure. Support for adaptive slicing is best provided by the source geometry, from which, for instance, curvature information can be obtained. However, adaptive slicing is not entirely a geometric problem, as layer thickness is often a function of material as well. In selective laser sintering, for instance, varying the layer thickness will probably have to be coupled to modulated laser power to ensure uniform material properties.

SFF processes offer the promise of providing manufacturing capabilities that are not realizable by other techniques. One such possibility is selective material property distribution within the part. With conventional material removal processes, the bulk mechanical properties of a part are determined by the stock material chosen, aside from any surface treatment that is applied as a post-process. With SFF technologies the potential exists for the mechanical properties to vary continuously within the part. To realize this possibility, design tools are needed to guide the designer in determining optimal material distribution, and data exchange standards must provide the capability to represent the spatial distribution of materials.

## Conclusions

Based on the requirements described above, it is not clear whether SFF would benefit from a *unique* standardized information format. Such a format, the STL file, is used in practice, but has many shortcomings and will not support future developments in SFF technology (*e.g.*, multiple materials) without modification. Also, faceted models do not support global geometric reasoning, certainly not as well as higher-order geometric representations.

A two-dimensional layer-based standard that includes attributes such as layer thickness and material distribution is possible. Generation of such a representation requires slicing faceted or surface geometry (which are already done), and thresholding and interlayer interpolation of layered data. However, such a representation will not support global geometric reasoning, such as orientation and adaptive slicing. These operations can be performed prior to generation of a two-dimensional representation for exchange. However, as discussed above, even global reasoning is SFF technology dependent. Thus, the process independence sought in a layer-based exchange format will not be realized.

A third avenue to standardization is a format based on three-dimensional surface/solid geometry, augmented with material distribution information, mechanical property preferences, tolerances, etc. This ideal is not very different from other efforts at data exchange standardization for mechanical design (e.g., PDES/STEP). Under this scenario, each SFF equipment vendor would be responsible for providing adequate geometric coverage in terms of slicing algorithms, as well as supporting appropriate attributes, such as multiple materials. The real standardization effort would then be developing common support algorithms such that the designer can reliably predict the results of processing the part description.

# LAYERED MANUFACTURING IN PROJECT MAXWELL

DEBASISH DUTTA,
DEPT. OF MECHANICAL ENGG. AND APPLIED MECHANICS,
THE UNIVERSITY OF MICHIGAN,
ANN ARBOR, MI 48109-2125.

Manufacturing technologies provide methods for the physical realization of designs. Layered Manufacturing (LM) refers to the creation of parts (solid objects) by depositing materials layer-by-layer. It is a fundamentally different method for the realization of designs when compared to the (conventional) metal removal/forming processes. The difference stems from the fact that in LM processes:

1) geometric and topologic complexities are factored out (i. e., a solid with an interior void and bounded by a sculptured surface is no more difficult to "build" than is a solid cube).
2) heterogeneous (i.e. multi-material) parts can be readily built.
3 no cutting/forming tools or dies/electrodes are required.

While advances in manufacturing technologies, in the past few decades, have dealt primarily with improvements in part quality and production speed, LM processes yield a larger domain of realizable parts. Consequently, for a designer, LM provides an expanded design space.

At the University of Michigan, research in LM is driven by the opportunities provided by, and requirements of, a new design technique. Project Maxwell is an integrated design and layered manufacturing research project sponsored by various federal agencies and industry. The main topics of research in Project Maxwell are:

- shape and material design using the homogenization technique.
- process planning for LM.
- development of a "solid" representation scheme.

We remark on each of these briefly.

In the homogenization technique, developed by Bendsoe & Kikuchi [2], the shape and topology design is reformulated as a problem in material distribution. Based on a given design domain (discretized by finite elements) and specified external loads, an optimum material distribution is derived (for a user specified objective, e.g. minimize compliance). The material distribution is a grey scale image corresponding to the density distribution within the domain. A typical result consists of fully dense regions (density = 1), fully void regions (density = 0) and possibly large areas of intermediate densities. This homogenization technique can be used effectively to synthesize material microstructures that correspond to designer prescribed part characteristics such as Poisson's ratio (positive or negative), eigenfrequencies, etc.

Homogenization based design is a powerful tool that can often and sometimes impossible, to manufacture such parts using conventional methods. However, since geometric/topologic complexities are factored out to a significant extent in LM, it is ideally suited for Maxwell parts.

Manufacturing these complex parts brings us to the second topic process planning for LM - in Project Maxwell. Layered manufacturing has often been advertized as a technique that converts a 3D CAD model to a physical prototype at the "press of a button" (presumably by the designer). However, the computational support necessary for "press of a button" layered manufacture is not in place yet.

Process planning is a term that we use to refer collectively to the host of issues that need to be considered, planned for, and executed prior to the commencement of the LM process. The following have to be determined, with a view towards manufacturing speed and part quality:

- part orientation for buildup.
- support structures that fixture and stabilize the part during build up.
- slice thickness based on bounding surface geometry and acceptable cusp height.
- path planning for deposition nozzle (LM process specific).

In Project Maxwell, we are developing computational tools for the above using the ACIS geometric kernel. Details of these developments are available in the technical reports [1,3,4]. In addition, prior to the layer manufacture of Maxwell parts, a key step is to "threshold" the density distribution. This is a process planning task that is design process-specific. In this step, all regions of intermediate densities are reclassified into full or zero densities, thereby converting the homogenization output into a binary (dense/void) region. Several prototypes have been manufactured using our Stratasys machine.

The thresholding step described above brings to the surface a fundamental limitation, with respect to layer manufacture of heterogeneous solids, prevailing in current solid modeling techniques. Hence our focus on topic three - development of a "solid" representation scheme - in Project Maxwell.

Current solid modeling techniques allow the representation of solid objects by accurately representing (i) the geometry of the bounding surfaces of the solid, and (ii) the topology (i.e. connectivity). This method of representing a solid by its boundary is rooted in orthographic projections/three view engineering drawings, developed by the French engineer/mathematician Gaspard Monge in the early 1800s. With the advent of computers, this elegant technique evolved from manual drafting systems to wireframe, surface and finally to present day solid modeling (B-Rep or CSG systems).

However, the principal deficiency of such a boundary-based modeling scheme is that heterogeneous solids cannot be modelled. As evidenced by the homogenization output, a density distribution corresponding to the initial design problem (i.e. domain and loading conditions) contains material information for the entire design domain which needs to be modeled without thresholding. However, the thresholding process results in representing only the object boundary. It is unfortunate because the material information is completely lost in the process.

Layered manufacturing is a powerful technique that provides us explicit control at each layer (of material deposit). However, there does not exist a solid modeling technique that can incorporate the material (and microstructure) attributes. Since the solid modeler/kernel is at the heart of any process planning and automated fabrication system, improper or inadequate representation will severely limit the system capability.

In summary, we would like to explore issues related to topics two (process planning for LM) and three (solid representation) in this workshop. Layered manufacturing is a fundamental development in manufacturing that will likely parallel the introduction of numerical control machines. Given the variety and importance of engineering materials, LM is uniquely poised to enable the rapid creation of highly efficient products (parts, assemblies, etc.). This is, in essence, the guiding philosophy of Project Maxwell.

## REFERENCES


(1) Allen, S., and Dutta, D., "On the Computation of Part Orientation Using Support Structures in Layered Manufacturing," Technical Report UM-MEAM-TR-94-15, Department of Mechanical Engineering, The University of Michigan, Ann Arbor, July 1994.

(2) Bendsoe, M. P., and Kikuchi, N., "Generating Optimal Topologies in Structural Design using a Homogenization Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, 1988, pp. 197-224.

(3) Kulkarni, P., and Dutta, D., "Adaptive Slicing of Parametrizable Algebraic Surfaces for Layered Manufacturing," submitted to, ASME Design Automation Conference, Boston, MA, 1995.

(4) Sreeram, P., and Dutta, D., "Determination of Optimal Orientation based on Variable Slicing Thickness in Layered Manufacturing," Technical Report UM-MEAM-TR-94-14, Department of Mechanical Engineering, The University of Michigan, Ann Arbor, July, 1994

# LAYERED MANUFACTURING IN PROJECT MAXWELL

DEBASISH DUTTA,
DEPT. OF MECHANICAL ENGG. AND APPLIED MECHANICS,
THE UNIVERSITY OF MICHIGAN,
ANN ARBOR, MI 48109-2125.

Manufacturing technologies provide methods for the physical realization of designs. Layered Manufacturing (LM) refers to the creation of parts (solid objects) by depositing materials layer-by-layer. It is a fundamentally different method for the realization of designs when compared to the (conventional) metal removal/forming processes. The difference stems from the fact that in LM processes:

1) geometric and topologic complexities are factored out (i. e., a solid with an interior void and bounded by a sculptured surface is no more difficult to "build" than is a solid cube).
2) heterogeneous (i.e. multi-material) parts can be readily built.
3 no cutting/forming tools or dies/electrodes are required.

While advances in manufacturing technologies, in the past few decades, have dealt primarily with improvements in part quality and production speed, LM processes yield a larger domain of realizable parts. Consequently, for a designer, LM provides an expanded design space.

At the University of Michigan, research in LM is driven by the opportunities provided by, and requirements of, a new design technique. Project Maxwell is an integrated design and layered manufacturing research project sponsored by various federal agencies and industry. The main topics of research in Project Maxwell are:

- shape and material design using the homogenization technique.
- process planning for LM.
- development of a "solid" representation scheme.

We remark on each of these briefly.

In the homogenization technique, developed by Bendsoe & Kikuchi [2], the shape and topology design is reformulated as a problem in material distribution. Based on a given design domain (discretized by finite elements) and specified external loads, an optimum material distribution is derived (for a user specified objective, e.g. minimize compliance). The material distribution is a grey scale image corresponding to the density distribution within the domain. A typical result consists of fully dense regions (density = 1), fully void regions (density = 0) and possibly large areas of intermediate densities. This homogenization technique can be used effectively to synthesize material microstructures that correspond to designer prescribed part characteristics such as Poisson's ratio (positive or negative), eigenfrequencies, etc.

Homogenization based design is a powerful tool that can often and sometimes impossible, to manufacture such parts using conventional methods. However, since geometric/topologic complexities are factored out to a significant extent in LM, it is ideally suited for Maxwell parts.

Manufacturing these complex parts brings us to the second topic process planning for LM - in Project Maxwell. Layered manufacturing has often been advertized as a technique that converts a 3D CAD model to a physical prototype at the "press of a button" (presumably by the designer). However, the computational support necessary for "press of a button" layered manufacture is not in place yet.

Process planning is a term that we use to refer collectively to the host of issues that need to be considered, planned for, and executed prior to the commencement of the LM process. The following have to be determined, with a view towards manufacturing speed and part quality:

- part orientation for buildup.
- support structures that fixture and stabilize the part during build up.
- slice thickness based on bounding surface geometry and acceptable cusp height.
- path planning for deposition nozzle (LM process specific).

In Project Maxwell, we are developing computational tools for the above using the ACIS geometric kernel. Details of these developments are available in the technical reports [1,3,4]. In addition, prior to the layer manufacture of Maxwell parts, a key step is to "threshold" the density distribution. This is a process planning task that is design process-specific. In this step, all regions of intermediate densities are reclassified into full or zero densities, thereby converting the homogenization output into a binary (dense/void) region. Several prototypes have been manufactured using our Stratasys machine.

The thresholding step described above brings to the surface a fundamental limitation, with respect to layer manufacture of heterogeneous solids, prevailing in current solid modeling techniques. Hence our focus on topic three - development of a "solid" representation scheme - in Project Maxwell.

Current solid modeling techniques allow the representation of solid objects by accurately representing (i) the geometry of the bounding surfaces of the solid, and (ii) the topology (i.e. connectivity). This method of representing a solid by its boundary is rooted in orthographic projections/three view engineering drawings, developed by the French engineer/mathematician Gaspard Monge in the early 1800s. With the advent of computers, this elegant technique evolved from manual drafting systems to wireframe, surface and finally to present day solid modeling (B-Rep or CSG systems).

However, the principal deficiency of such a boundary-based modeling scheme is that heterogeneous solids cannot be modelled. As evidenced by the homogenization output, a density distribution corresponding to the initial design problem (i.e. domain and loading conditions) contains material information for the entire design domain which needs to be modeled without thresholding. However, the thresholding process results in representing only the object boundary. It is unfortunate because the material information is completely lost in the process.

Layered manufacturing is a powerful technique that provides us explicit control at each layer (of material deposit). However, there does not exist a solid modeling technique that can incorporate the material (and microstructure) attributes. Since the solid modeler/kernel is at the heart of any process planning and automated fabrication system, improper or inadequate representation will severely limit the system capability.

In summary, we would like to explore issues related to topics two (process planning for LM) and three (solid representation) in this workshop. Layered manufacturing is a fundamental development in manufacturing that will likely parallel the introduction of numerical control machines. Given the variety and importance of engineering materials, LM is uniquely poised to enable the rapid creation of highly efficient products (parts, assemblies, etc.). This is, in essence, the guiding philosophy of Project Maxwell.

## REFERENCES

(1) Allen, S., and Dutta, D., "On the Computation of Part Orientation Using Support Structures in Layered Manufacturing," Technical Report UM-MEAM-TR-94-15, Department of Mechanical Engineering, The University of Michigan, Ann Arbor, July 1994.

(2) Bendsoe, M. P., and Kikuchi, N., "Generating Optimal Topologies in Structural Design using a Homogenization Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, 1988, pp. 197-224.

(3) Kulkarni, P., and Dutta, D., "Adaptive Slicing of Parametrizable Algebraic Surfaces for Layered Manufacturing," submitted to, ASME Design Automation Conference, Boston, MA, 1995.

(4) Sreeram, P., and Dutta, D., "Determination of Optimal Orientation based on Variable Slicing Thickness in Layered Manufacturing," Technical Report UM-MEAM-TR-94-14, Department of Mechanical Engineering, The University of Michigan, Ann Arbor, July, 1994

# SFF MANUFACTURING WHITE PAPER

**James Glimm**
**Department of Applied Mathematics and Statistics**
**University at Stony Brook**

There are four basic components to the modeling of the SFF manufacturing process. These are (1) a geometric description of the evolving surface, (2) a materials description of the deposition, in terms of sticking probabilities,etc., (3) shape optimization, and (4) a description of post formation cure, to account for warpage, shrinkage, spring back and residual stresses. Each of these modeling components should be modular in construction, using common data structures, so that communication betweeen modules is transparent. Various aspects of the first three problems identified above are under investigation at Stony Brook.

The Geometry Editor. What is needed is a description of a general surface in three dimensions with boundary. This geometrical object comprises 2, 1, and 0 dimensional objects. The surfaces have boundaries (curves), and are non self-intersecting except at their boundaries. The curves are either simple closed curves, or segments with endpoints (boundaries). The surfaces and curves can be described by piecewise linear elements (simplices), and in principle higher order elements can be used also. Boundary and coboundary operators relate curves and surfaces to one another, and curves and their endpoints (called nodes) to one another. Code to implement these ideas has been developed at Stony Brook as part of the Front Tracking method developed there. The two dimensional version of this interface library is discussed in: J. Glimm and O. McBryan, ``A Computational Model for Interfaces" Adv. Appl. Math. 6 (1985) 422-435. The two dimensional code is by now fairly mature, and supports such features as collision detection and post collision reconstruction of an interface.

The Deposition Process. A flux function, or sticking probability, is needed to determine the motion of the material surface during the SFF process. The flux function will in general depend on process parameters, surface geometry and on angles relating the deposition process to the surface. Abstract physics models of these processes should be compared to real processes to determine the adequacy of the conceptual framework and abstractions being assumed. Thermal spray and semiconductor deposition and etching provide examples of SFF process to be considered.

Shape Optimization. This is the problem of tool or jet motion pattern, to produce a desired final surface. Because the tool or jet is not perfectly sharp, or perfectly columnated, or the mask infinitely narrow, there is an inverse problem to be solved in going from the geometrical shape to the tool motion which will produce it.

Cure. Many processes include post manufacture cure, which can modify shape through shrinkage, warpage, (and for dye forming of metals, spring back). etc., and can induce residual stresses which may be desirable or undesirable. The direct problem is to model the cure process. The inverse problem is to design the formation (pre-cure) geometry and the cure process to obtain optimal post-cure geometries and material properties.

# Design Rules and Digital Interface for Laser-Aided Direct Rapid Prototyping (LADRP)

Aravinda Kar[1] and Amar Mukherjee[2]
University of Central Florida, Orlando, Florida
[1]Center for Research and Education in Optics and Lasers (CREOL)
Mechanical and Aerospace Engineering Department
[2]Computer Science Department

## Introduction

Lasers have made a considerable impact in manufacturing sciences and technologies. Various industries are interested in laser technology because of its simplicity and economic impact on materials processing. Since laser beams are inertialess and contactless tools, they are readily adaptable to computer control for process automation. Also, laser tooling leads to the unification of various conventional tools because a laser beam can be used to carry out different types of operations such as cutting, welding, and drilling. Due to this unique feature of lasers, an integrated process involving various manufacturing tasks can be designed and successfully accomplished by using laser technology. Such process integration is very important for the rapid production of three-dimensional parts directly in near net shape.

Lasers have been used to deposit thin films and thick coatings by using laser chemical vapor deposition and laser cladding techniques, respectively. The laser cladding technique, which uses material powder, can be utilized to rapidly prototype three-dimensional parts by direct deposition of materials. However, only a fraction of the powder is melted at the laser-irradiated spot, and the rest of the powder remains unutilized. Instead of using powder, a wire, rod, or bar of the material can be used as a sacrificial nib to write/deposit the desired pattern to rapidly produce three-dimensional parts directly. This technique will be investigated in the proposed **Laser-Aided Direct Rapid Prototyping** (LADRP) paper.

We will develop a rapid prototyping technique to produce three-dimensional parts directly by using a laser beam and a sacrificial nib. This technique is analogous to the process of writing with a pen, where the ink flows along a nib and deposits on a piece of paper to form a pattern according to the way the pen is moved. In this rapid prototyping process, the nib is a wire, rod, or bar of the material with which the part has to be produced and the molten material obtained by melting the nib acts as the ink. Thus, the nib is sacrificed in this process to produce the ink (molten material) in order to write (deposit) the desired pattern to prototype the part. A schematic diagram for the proposed process is given in Fig. 1. In this process an inert shielding gas is also used to prevent the formation of oxides and nitrides due to interactions between the laser-heated material and the air.

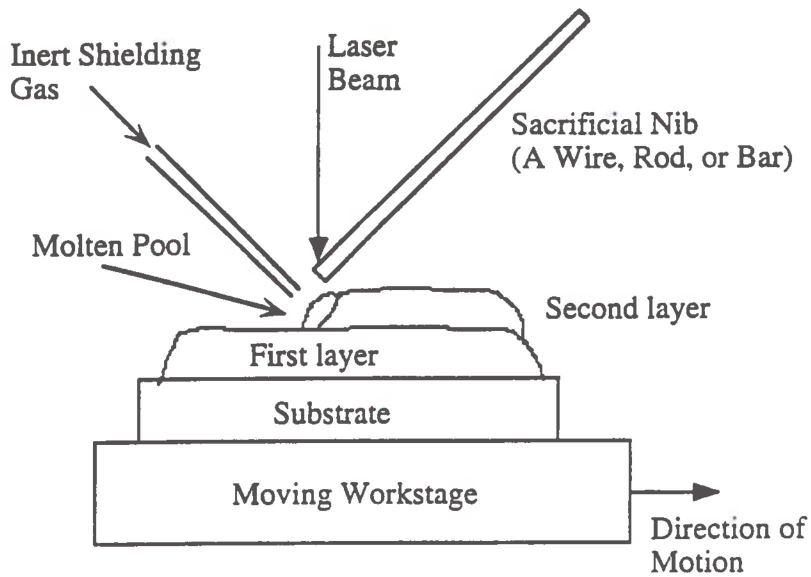



Figure 1. A schematic diagram for Laser-Aided Direct Rapid Prototyping using Sacrificial Nib.

The success of laser-aided prototyping will depend on the development of systematic design methodologies, CAD tools and simulation algorithms incorporating proper understanding of various aspects of laser-materials interactions. Developing a suitable digital interface between design and manufacturing will separate the issues of high level design abstractions with low level physical parameters of design. Such an approach has been used in the past very successfully for the design of VLSI (Very Large Scale Integration) circuits using a multilevel hierarchical design methodology. In this method, the operation of the circuit is abstracted at different levels by formal systems which allows separation between systems design, component design and fabrication. A clean digital interface between design and fabrication is established by providing a set of design rules in two-dimensional geometries. For mechanical and electromechanical systems, such an approach using multilevel hierarchical decomposition does not work very well because the underlying models involve energy transformation and physical parameters, the elemental components *share function* and behave differently in a system due to back loading [Mukherjee and Hilibrand (1994)]. As depicted in Fig. 2, an abstract model of a mechanical system must be able to represent complex interaction between geometry and physical world and might involve multiple iteration between these two domains. The digital interface for mechanical system takes the form of a process planning step which defines the fabrication rules concerning the feasibility of parts with respect to the required specifications. In this paper, we present our preliminary work in defining the digital interface in terms of a set of design rules that are formulated as the basis for the interface of the high level CAD tools.

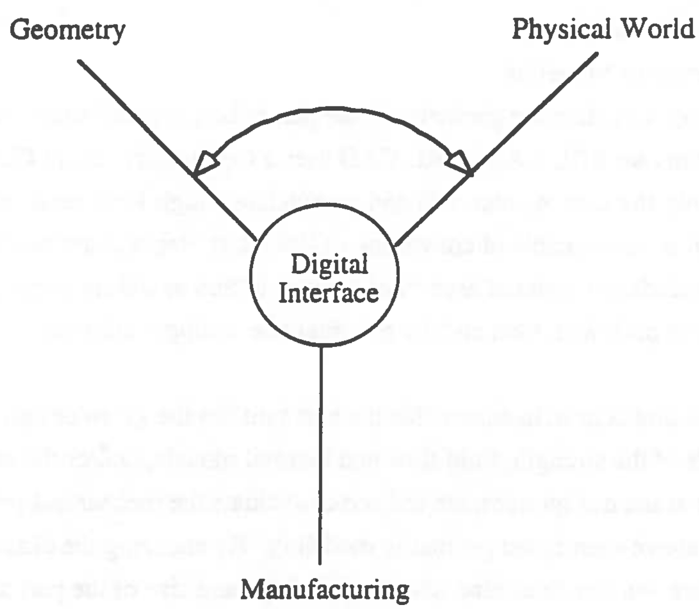



Figure 2. Abstract model of manufacturing.

## Computer-Aided Manufacturing (CAM) for LADRP

We envisioned our Computer-Aided Manufacturing (CAM) system for the LADRP process to consist of three subsystems as depicted in Fig. 3. A more detailed breakdown of the three subsystems of Fig. 3 is shown in Fig. 4.


Figure 3. Computer-Aided Manufacturing System for LADRP

**Modeling:** The Modeling subsystem has two main components:

(i) Geometric Modeling

(ii) Performance Modeling

Geometric modeling is used to define the geometry of the part to be produced using some geometric modeling system like BRL-CAD. BRL-CAD uses a Constructive Solid Geometry (CSG) system which allows the user to enter, edit and manipulate a high-level representation of a three-dimensional part. It is also capable of converting to Boundary Representation (BREP) in IGES format. We also include the material aspects of the part in this modeling stage. For example, a portion of the part may be built with steel and the rest might be a copper alloy etc.

Performance modeling is used to ensure that the part satisfies the given design requirements. The results of the strength, fluid flow and thermal models, and/or the empirical data, which are classified as the design rules, are utilized to evaluate the mechanical performance of the object created by the above-mentioned geometric modeling. By checking the dimensions of the object with the design rules, we can determine whether the shape and size of the part are suitable and optimized for the desired application. The output of this modeling subsystem would be a part geometry with optimum dimension which is used to develop a plan for the fabrication process as discussed below.

**Process Planning:** The process planning subsystem has three important roles in the rapid manufacturing process:

(i) Fabrication rules checking,
(ii) Determining the best fabrication orientation, that is, the best way to produce the part, and
(iii) Digital interface to drive the fabricator.

This subsystem takes the geometry of the object and verifies with the laser processing parameters to ensure the feasibility of the part fabrication - Fabrication rules checking.. This is done by checking the tolerances of the fabricator with the part dimensions. For example, if we want apply a thin coating of 200 micron width, we will need a laser delivery system to produce a laser beam of at least 200 micron diameter, and a wire of 200 micron diameter. If the diameter is less than 200 micron, we need additional process steps and planning to achieve uniformly thick film or coating. If the diameter is more than 200 micron, the required coating cannot be produced without additional process steps and planning. The second important feature of this subsystem is to examine the geometric orientation of the object and determine the best way to fabricate the part. At this stage, the computer knows how the part needs to be fabricated. Now, we need a digital interface to transfer the computer instructions to the fabricator in order to drive it in an appropriate manner to fabricate the part created by using the geometric modeling. For these reasons, we need a set of fabrication rules to define the process planning operation, and a digital interface to execute the actual fabrication of the part. The three components of the process planning subsystem, and a flow diagram for the overall scheme of the LADRP process is shown in Fig. 4.

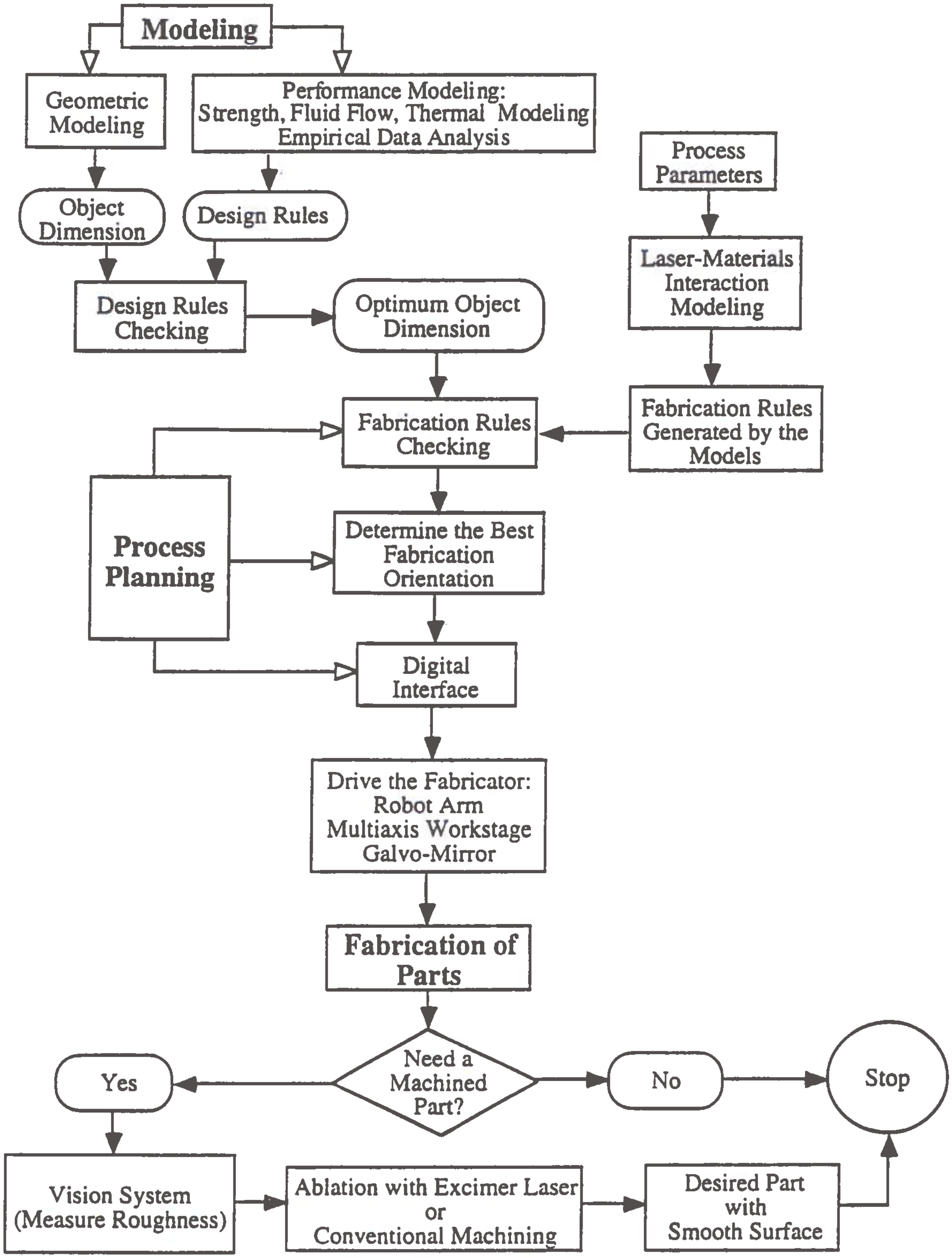





Figure 4. A flow diagram for the LADRP process with digital interface between design and manufacturing.

**Fabrication:** The fabrication step takes the instructions from the process planning step and manipulates the robot arm to fabricate the part. In this LADRP process, the three main requirements are: laser beam, feed wire and shielding gas. The proposed robot arm will have three veins, one carrying the laser through an optical fiber, the other one carrying the feed wire and the third vein will carry the shielding gas. This concept is similar to writing on a piece of paper with a pen which is held by three fingers simulating three components of our LADRP arm.

**Formulation of the Design Rules:**

The design rules for VLSI technology are usually specified in the form of minimum dimensions of transistors, rules governing mutual separation of wires and minimum width of the interconnecting wires. These rules can be specified in a conservative generic form which applies to a wide spectrum of fabrication processes [Mead and Conway (1980)]. For example, a set of design rules for CMOS could be used to design circuits at a higher level that can be fabricated by a variety of CMOS vendors.

In the case of Mechanical design, such generic rules applicable to a large class of fabrication processes do not seem to exit. This can be attributed partly because of the dependence of the functionality of the part on the particular material, geometry and fabrication technique used to produce the part. So the design rules in this case cannot be stated in terms of some minimum and maximum dimensions like in VLSI technology, rather it should be stated as a set of mathematical expressions that relate the geometric dimensions with the performance requirements of the parts. Let us take a simple example of designing a bracket. It has a general shape but its actual dimension will depend on the load specification. The same load-bearing capability of the bracket can be achieved by using various types of materials such as aluminum, iron, plastic, etc. But in each case, the dimension will be different since the mechanical strength of these materials are different. However, for all these materials, the strength and dimensions can be related by the same equation [Timoshenko and Goodier (1982)] which we will formulate as the defining design rule for this case, and they will act as dynamic design rules. The next step is to verify whether the given manufacturing process is capable of fabricating the part consistent with these design rules.

## Formulation of the Fabrication Rules

In this section, we will show how the process parameters are related to the part geometry.

Fabrication rules can be classified into three categories:

(1) Fabrication rules associated with the laser device. This will define the laser power, mode structure, etc.

(2) Fabrication rules associated with the beam delivery system. This will include beam diameter, fiber optic delivery system, ease of automation and fabrication.

(3) Fabrication rules associated with laser material interaction, laser properties , material properties and the geometry of the object.

The first set of rules are usually specified by laser engineers. The second set of rules are give by optical engineers and robot designers. In this paper we will assume that these two sets of rules are given by the supplier of laser machine tools. We will now focus our attention to the third set of design rules that will be utilized in the process planning stage.

## Energy Balance for Fabrication Rules

An energy balance between the input laser energy and the energy required to melt the wire at a specific rate [Charschan (1993)], such as Eq. (1), provides the starting point to relate various process parameters in order to generate the required fabrication rules. Eq. (1) is written for a wire with circular cross section.

$$PA(1-e_l) = \pi r^2 F\rho\left\{c_p(T_f - T_i) + L_m\right\} \quad (1)$$

where P = Power of the incident laser beam

A = Absorptivity of the material

$e_l$ = Fraction of the absorbed energy lost due to conduction, convection and radiation

r = Radius of the molten wire

F = Wire feed rate (M/s)

$\rho$ = density of the wire material
$c_p$ = specific heat of the wire material
$T_f$ = Temperature of the wire at the laser-material interaction zone
$T_i$ = Temperature of the wire before it reaches the laser-material interaction zone
$L_m$ = Latent heat of melting

For a given laser power and wire feed rate, the radius of the molten wire depends on the laser beam diameter. Assuming that the radius of the molten wire is proportional to the laser beam radius, we can write $r = Kr_b$, where K is a correction factor, and $r_b$ is the laser beam radius. Therefore, Eq. (1) can be written as

$$PA(1-e_l) = \pi K r_b^2 F \rho \left\{ c_p (T_f - T_i) + L_m \right\} \qquad (2)$$

Eq. (2) is a simple representation of the energy balance. An accurate analysis of the LADRP process will involve a set of nonlinear partial differential equations representing the conservation of mass, momentum and energy [Bird, Stewart and Lightfoot (1962)]. Since these differential equations require a long computational time, it is impractical to implement them for the purpose of obtaining the design rules on-line as the LADRP process progresses in real-time. For this reason, Eq. (2) is useful for digital interface.

Although, the partial differential equations are difficult to implement in practice, they can be solved beforehand to determine the unknown parameters $e_l$, K, $T_f$ and $T_i$ that appear in Eq. (2). In this respect, mathematical modeling is useful because it allows us to

(a) develop a science base for the process,
(b) understand the relationships among various process variables,
(c) determine the important process parameters, and
(d) evaluate important parameters so that a simple equation can be used to obtain the design rules for digital interface.

Also, experimental studies are important because they allow us to

(a) verify the model predictions and the assumptions used in the model, and

(b) check the overall integrity of the process design.

**Thickness Rule**

From Eq. (1), we can write

$$r = \sqrt{\frac{PA(1-e_l)}{\pi F\rho\left\{c_p(T_f - T_i) + L_m\right\}}} \tag{3}$$

For a given material and a fabrication process, the quantities in the denominator is constant. Therefore, Rule 1 can be stated as follows.

**Rule 1:** Given a set of process conditions and a material, the thickness of the layer in the LADRP process is directly proportional to the square root of the product of the laser power and energy utilization factor.

**Minimum Thickness**

Eq. (3) indicates that the minimum thickness of the deposited layer, r would be zero when the wire feed rate, F is infinitely large. However, this minimum value is not useful to set any criterion for checking the feasibility of fabricating any part. For a given material and a set of process parameters, the minimum thickness, r, that can be achieved in practice, must satisfy Eq. (3). Therefore, Eq. (3) will be used in the process planning step to quickly determine the thickness of the part that can be fabricated for the given material and process parameters, and this computed value will be compared with the given dimension of the part in order to check whether the desired part can be fabricated.

**Maximum Thicknes**

The shape of the deposited layer and its thickness need to be obtained by solving the heat balance equation, Navier-Stokes equation and Laplace-Young surface tension formula. These equations are nonlinear partial differential equations which can be solved by using numerical techniques. Although accurate predictions can be made by solving these equations, the computational time for such techniques is so large that it is impractical to implement this kind of elaborate procedure in the process planning step of the LADRP process. For this reason, simple but sufficiently accurate formulae have to be developed for on-line verification of the feasibility of fabrication of a given object. An example to calculate the thickness, $r_d$, of the deposited layer is given below:

Assuming that the layer is deposited in the form of a semi-cylinder on the substrate surface, the mass balance between the feed wire and the deposited layer yields the following relation.

$$r_d = \sqrt{2}\, r \qquad (3)$$

Now $r_d$ will be maximum when r is maximum. According to Eq. (1), r is maximum when $e_l = 0$ and $T_f = T_m$, where $T_m$ is the melting temperature of the material. So the maximum value of r is given by

$$r_{max} = \sqrt{\frac{PA}{\pi F \rho \left\{ c_p (T_m - T_i) + L_m \right\}}} \qquad (4)$$

Combining Eqs. (3) and (4), we find that the maximum thickness, $r_{d.max}$ of the deposited layer is proportional to $\sqrt{P}$.

**Rule 2:** Given a set of process conditions and a material, the maximum thickness of the deposited layer in the LADRP process is directly proportional to the square root of the laser power.

**Typical Process Variables:**

The important process variables that affect the design rules are discussed below:

(a) Laser power - It affects the energy input to the material.
(b) Laser beam radius - It determines the radius of the wire that can be melted. The beam radius depends on the focal length of the lens. The theoretically achievable

smallest beam radius is given by the diffraction-limited spot size. For a given laser machine and beam focusing system, the beam radius can be computed by considering the propagation of the laser beam.

(c) Wire feed rate - It influences the rate of manufacturing the part. This process variable is usually selected on the basis of economic considerations.

## Conclusion:

In this paper, we present an approach to rapidly fabricate three-dimensional parts using a Laser-Aided Direct Rapid Prototyping technique. We have presented our preliminary thoughts on the definition of a digital interface based on a process planning model and understanding of the interrelationship between the geometry and the laser process parameters. We have identified the differences between the design rules used in the VLSI technology and the design rules that need to be used for mechanical design. Dynamic design rules are required for mechanical part design. Based on a simple energy balance model, we have developed some fabrication rules for the LADRP process. We have incorporated into these fabrication rules physical parameters such as absorptivity, density, specific heat, latent heat of fusion, laser power, and the fabrication rate. We also report our preliminary investigation on the development of an integrated design environment for this process using a front-end solid modeling interface. The digital interface will form the basis of exchange of information between the design and the underlying fabrication process.

## References:


Bird, R. B., Stewart, W. E., and Lightfoot, E. N., 1962, Transport Phenomena, (John Wiley and Sons, New York)

Charschan, S. S., Editor, 1993, Guide to Laser Materials Processing, (Laser Institute of America, Orlando, Florida)

Mead, C. and Conway, L., 1980, Introduction to VLSI Systems, (Addison Wesley, New York).

Mukherjee, A. and Hilibrand, J., (Eds.) 1994, New Paradigms for Manufacturing, (NSF Workshop, May 2 -4, Arlington, Virginia)

Timoshenko, S. P. and Goodier, J. N., 1982, Theory of Elasticity, International Student Edition, Third Edition (McGraw-Hill, London,).

# Data Exchange for Solid Freeform Fabrication

**Earlin D. Lutz**

**June 17, 1995**

---

Prepared for the National Science Foundation Workshop on Solid Freeform Fabrication, Carnegie Mellon University, Pittsburgh, PA, June 5-6, 1995.

Earlin Lutz, Bentley Systems, Inc., 690 Pennsylvania Dr., Exton PA 19341
Earlin.Lutz@Bentley.com
610-458-2734

1.

## 1. Introduction

Solid Freeform Fabrication (SFF) refers to manufacturing processes that produce complex objects 'particle-by-particle', usually proceeding in a bottom-to-top sequence of layers (cross sections). The most successful example of this technology is the STL (stereolithography) process for plastic resin parts; however, others, including ones producing metal parts, are in various phases of research.

This paper discusses present and future methods of passing geometry from mechanical CAD systems to the SFF process. Section 2 reviews the current methods, specifically the use of the STL file format for triangulated solids. Section 3 outlines a 'strawman' proposal which has been put forward calling for a new high level standard file format, with the goal of facilitating a manufacturing revolution along the lines of similar standardization of geometric data formats in the VLSI industry 15 years ago. Section 4 considers some difficulties in the strawman scenario.

Section 5 proposes that in place of the focus on *file formats* the SFF community consider using the *procedural interface* already in widespread use by 'third party software developers' who customize the modelers for specific applications.

## 2. Current Practice: Standardization on mid-level file formats

Current industrial practice for providing data for layered manufacturing processes is as follows (Fig. 1):

- Product designers, who may or may not be associated with the final manufacturing organization, use a variety of mechanical CAD systems to design process-independent 3D geometry. The stored forms of the geometry are (a) various proprietary file formats, (b) the IGES and STEP standards.
- Translation software is executed to triangulate the surface of the 3D geometry and write the triangulation to an STL file.
- STL file is passed on to a manufacturing site where process-specific layer geometry is extracted.

The STL file is an unstructured list of triangles, each triangle being given as 3 vertex coordinates. The triangles together define a polyhedral solid that approximates the original geometry.

This process has the following desirable attributes:

- Assuming that a globally-correct triangulation algorithm is already available, the step of producing the STL file is extremely easy in the CAD system.
- Because the STL file is purely geometric, it is not necessary to add the interactive support for process-specific data to the CAD systems. That is, there is no need to query for and subsequently store information such as the number and position of layers or resolution within layers.
- Regardless of the problems noted below, the process 'works', at least in the sense that a once a CAD software house has written the STL output software its users are in fact able to carry out the design and manufacturing steps.

The CAD vendor has no need to track further process development so long as the process software itself continues to use the STL file.

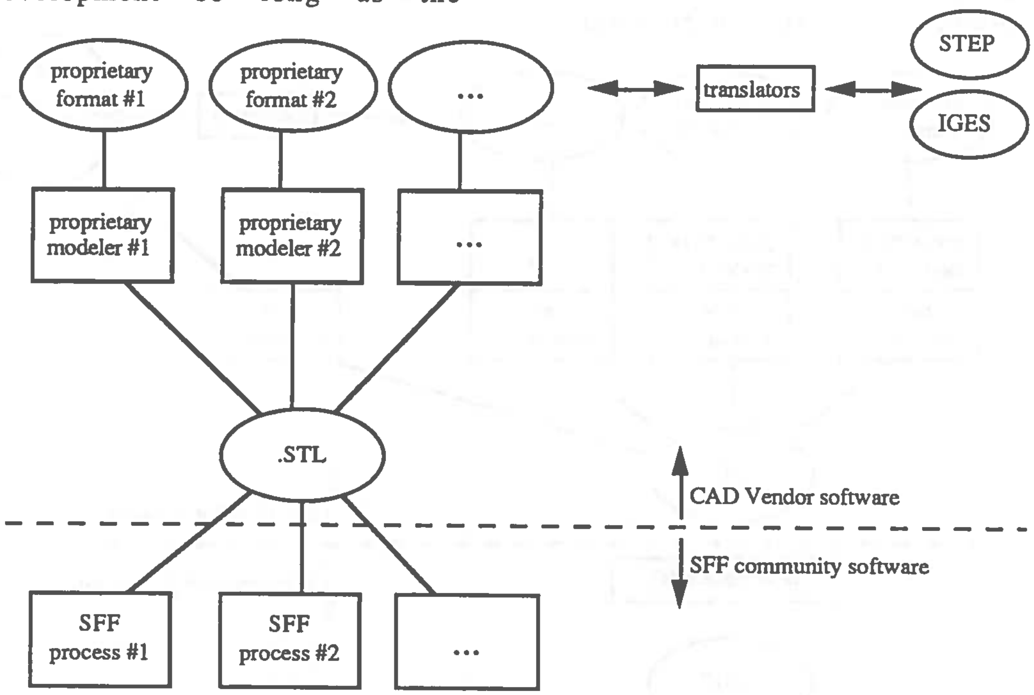



Fig. 1: STL-based geometry transfer

These positive attributes are offset by several negatives:

〈 Because the triangulation is an approximation, it is never possible to generate exact cross sectional data later on in the process.

〈 Triangulation loses design intent. It is a particular snapshot of the design, with no possibility of either altering its approximation tolerances or accommodating the notion of dimensions or parameterization of the overall part.

〈 The STL format contains *coordinates* but not *topology* (connectivity). Downstream software must 'sew' the triangles back together in order to perform typical computations such as validity checks and closed contour extraction.

〈 At the simple syntactic level, the file is extremely verbose. The coordinates of a vertex are repeated on each incident triangle. Since smooth triangulations tend to use each vertex roughly 6 times, the file size is roughly that factor larger than necessary.

〈 Information flow between the CAD and SFF communities is limited. For instance, CAD vendors receive little guidance about how (a) triangle aspect ratios and (b) number of triangles are significant to the process. Preference between these two has a significant effect on how to compute a triangulation. This gives the CAD vendor the awkward choice between (a) arbitrarily selecting one set of parameters (which guarantees that files are 'poor' in the view of at least some users) or (b) presenting the user with interactive options to select among cryptic algorithmic variants that are not really of interest.

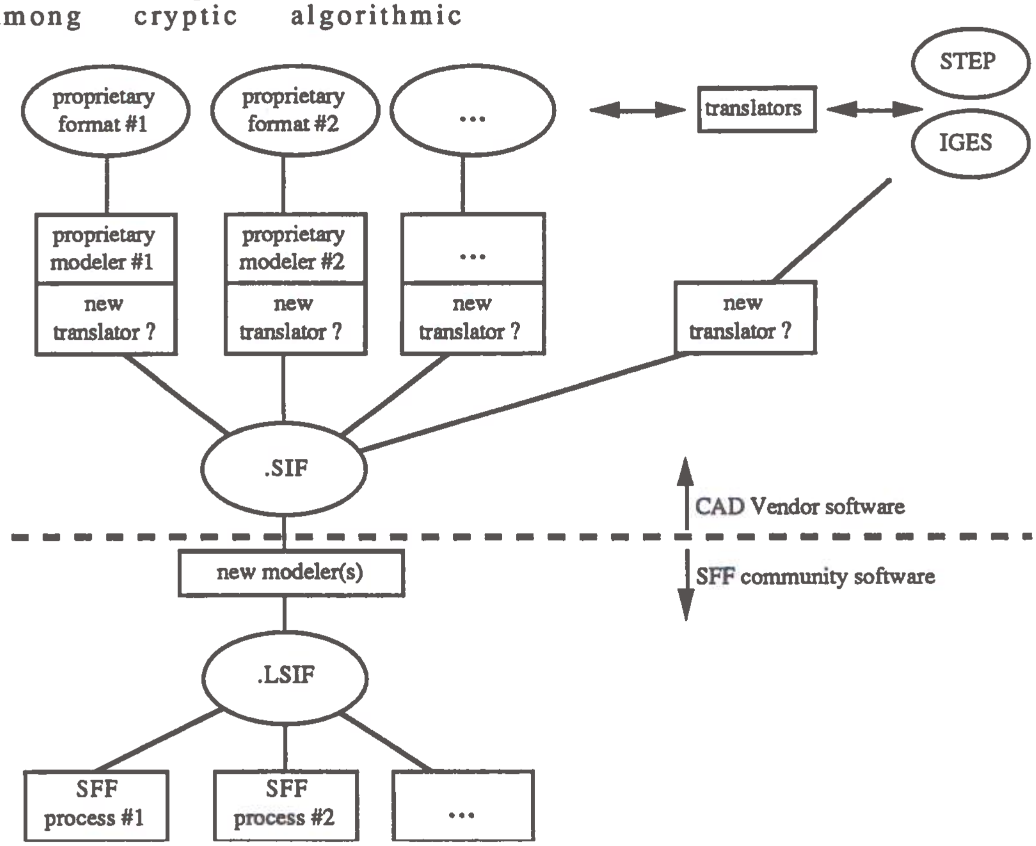



Fig. 2: SIF/LSIF geometry transfer

## 3. Strawman Proposal: Standardization on higher and lower level file formats

A baseline proposal for improving the process is to define two file formats in place of STL: first, a 'higher level' format that captures exact geometry and (if possible) design intent, and, second, a lower-level, single layer format for more direct use by the machine-level software. The high level format is tentatively labeled "SIF", for "solid interchange format", and the lower one "LSIF", for "layered solid interchange format".

The high level file is intended to provide process-independent description of exact geometry. A wishlist of what is desired typically grows to include:

〈 'boolean' operations of constructive solid geometry
〈 boundaries specified in the form of explicit face, edge, and vertex data
〈 commonly occurring analytic surfaces (e.g. quadrics)
〈 a freeform surface capability such as NURBS surfaces
〈 facility for placing predefined objects (variously called cells or symbols) from libraries
〈 simple keyword format conducive to visual inspection and hand editing

Following the (successful) example of VLSI standardization, it is hoped that once the standard format is established the following downstream effects will occur:

- Research sites that wish to extract particular data from such a file will be able to write translators which read the file, expand the geometry to suitable in-memory data structures, and use it as input to computations such as extraction of particular cross-sections of the geometry. These cross sections may be either (a) given directly to the SFF machine for immediate use, or (a) written to LSIF files to be reread by the machines at a later date.
- CAD software vendors will be motivated to write translators so that geometric models created in their (proprietary) systems can be written in the (non-proprietary) SIF format.
- Research and manufacturing sites will access part designs as SIF files.
- There is also some suggestion that the SIF format might be amenable to hand editing to construct text-only part descriptions.

(The lower level LSIF file format has not been described in a form that can be considered here.)

## 4. Difficulties with the strawman proposal

This author is of the opinion that the notion of commercial-modeler-to-SIF and SIF-to-LSIF translators

(a) commits translator writers to producing software which is, for all practical purposes, as complex as any current solid modeler

(b) demands that all existing solid modelers perform a difficult high-level-to-high-level translation

(c) fails to define a voxel-level interface that would directly support data sources such as medical imaging, Minkowski sums, and medial axis which are not readily translated to traditional boundary and CSG forms demanded by the SIF wishlist but are readily converted to voxel-level cross-section data.

(d) fails to take advantage of API protocols which can be used to procedurally access the cross-sectional data directly from existing solid modelers.

The following suggests problems that are likely in the strawman scenario:

- Because both BREP data and trimmed NURB surfaces are present, it can be expected that end users will supply arbitrarily large models with trimmed NURB surfaces. A quick , once-and-for-all implementation of the SIF-to-LSIF translation will not be possible; ongoing effort will be needed to maintain state-of-the art algorithms throughout.
- CSG-to-BREP conversion ("boundary evaluation") is a complex but moderately well understood problem, but BREP-to-CSG conversion is still a bleeding-edge research topic. Therefore, the presence of even *some* BREP and trimmed NURBS data implies that the working internal form must be a BREP, not the otherwise more tractable pure CSG tree. In particular, it is quite probable that the "cell-placement" facility will be implemented by doing a boundary evaluation after cell placement. In the case of massive replication of millimeter-scale surface textures, this will produce truly huge boundary data sets will demand extensive algorithmic attention.
- The assumption that the LSIF file includes *exact contour data*

implies that the SIF-to-LSIF converter will include the computation of exact cross-sections. Even the fact that it is a *planar* section does not change the fact that this involves substantial algorithmic complexity.

〈 Conversion among high level solid representations (e.g. among pure CSG, BREP, voxel, minkowski sum, medial axis forms) is notably more difficult than extraction of lower level forms such as cross-sectional bitmaps. Hence the modeler-to-SIF translator may be quite complex if the full strawman wishlist is supported. The line between software supported by the SFF and CAD vendors will probably be above, rather than below, the translators.

The well-know problem of special-purpose-versus-general-purpose operating systems is relevant here: a system designed for a single, well-posed problem can be simple and fast, but will invariably become more complex (and less satisfactory) as it is applied to a broader problem mix. The strawman proposal hopes that the SIF format can be quite all-inclusive but still be supported by a simple, specialized translator.

There may indeed be selected selected high-level geometries for which a translator can be easily implemented. One example of this is conversion of unevaluated boolean trees directly to voxel models without ever executing boundary evaluation. However, the presence of NURBS, boundary topology, and boolean operations in the strawman wishlist demands a full-featured solid modeler at the SIF-to-LSIF interface.

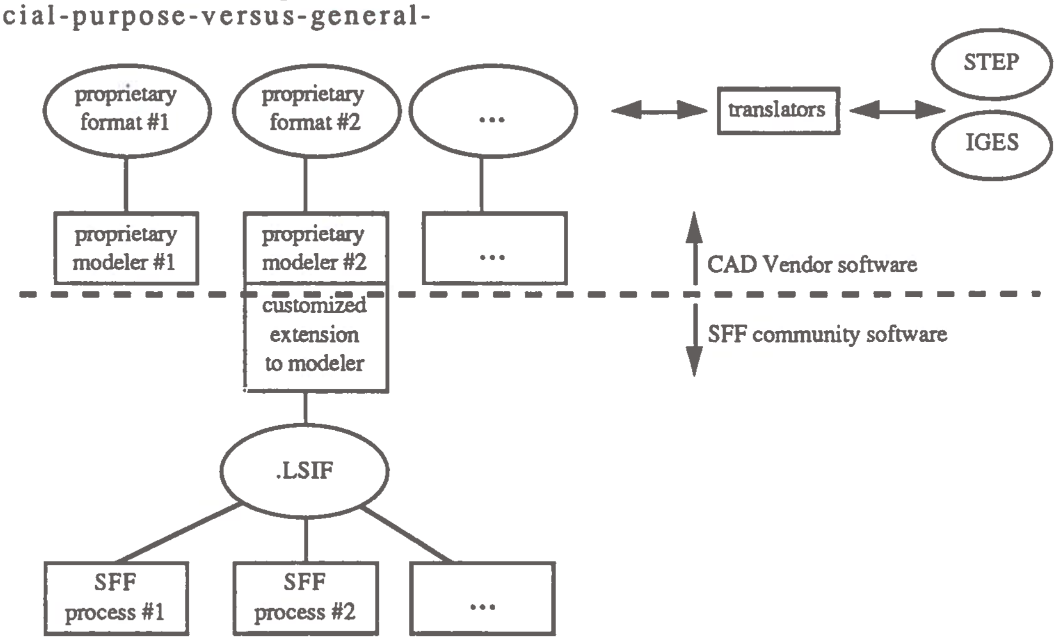



Fig. 3: Direct interface of SFF software to modelers

This author is not directly familiar with the low level details of the computer control of various SFF processes; however, it seems virtually certain that these controls are implemented by *extending and customizing* operating systems developed for other purposes. An SFF-specific operating system was probably never seriously considered. The solid modeling part of the process is analogous: rather than implementing an SFF-specific solid modeler, it may be advantagous to implement an SFF-specific customization of an existing modeler, most likely a commercial product for which the customization protocols are well developed.

## 5. Alternative Proposal: Use procedural access to existing modelers

An alternative to developing an SFF-driven file format and translator is to use existing solid modelers in the manner of sophisticated "third party software developers" to customize the modeler, rather than the more limited interactive user mode.

The necessary libraries and protocols for such development is readily available from the solid modeling vendors. Extraction of cross-sectional contours is always included in such libraries, either as an explicit slicing operation or as a result of the boolean intersection between the object geometry and a plane or a large cube of which one face is the section plane. (Representative procedures for several modelers are noted in the appendix.)

The customization process requires the following steps on the part of the software developer (in this case, the SFF process research group, service bureau, or machine manufacturer):

1. Obtain (from the solid modeler supplier) the technical specs for interfacing to the solid modeler at the *programmatic* level, rather than simply as an interactive user. The interface protocols will typically include (a) procedures to traverse the raw model data structure, (b) procedures to manipulate the geometry, and (c) procedures to prompt for and receive geometric input (e.g. coordinate data and object selections) in the style of the system's native user interface. These '3rd party developer' protocols are a critical part of many successful solid modelers. As with the more limited interactive licenses, academic users can generally obtain them at very significant discounts.
2. Formulate the desired computation and interaction (e.g. cross-section extraction) in terms of the modeler's data structure.
3. Output the results of the computation, either directly to the SFF process or to whatever low level file format becomes appropriate.

A certain up-front investment of time is required to become familiar with the data structures and library protocols. However, the following benefits may be expected:

- The total amount of code required will be far smaller than in the strawman proposal.
- The customized software will be developed in the context of a well developed data structures and interfacing conventions.
- The full interactive environment of the modeler will be available for testing and use of the customized software.
- The full data of the original solid modeler will be available, with none of the losses that occur in high-level-to-high-level translations.
- There is no need to depend on other parties to develop the modeler-to-SIF translators. (Indeed, in the early phases of the strawman proposal the SFF community will probably write basic modeler-to-SIF translators themselves, using the third party developer protocols

exactly as outlined here. Once that step is taken, the SFF community will essentially be in the third-party-developer role already.)

- Although the initial choice of a base modeler may appear to restrict the possible sources of input data to the native format of that modeler, there are already substantial pressures to support good translators among the various modeler formats and standards such as STEP. These mainstream translation tools will be the best available, while the special-purpose SIF translators will to be low-priority for some time.
- All improvements of the still-evolving base modeler, whether in construction tools, user interface, additional translators, or (most importantly) algorithmic improvement in the developer library, will be received.
- The SFF developer will have the option of interfacing directly from the solid modeler to the SFF machine, or sending output to an intermediate data file.

Every modeler that is in widespread use today started out with the same lofty objectives of simplicity and power as those stated by the SFF community. Perhaps the SFF effort may solve some of the various problems in these less-than-perfect systems. On the other hand, there is substantial additional power available in all of them if they are accessed at the programmatic developer level instead of as strictly interactive systems.

## 6. Conclusions

The strawman SIF-to-LSIF plan is commitment to either (a) replicate nearly the full range of software in existing solid modelers, or (b) come up with truly revolutionary solutions to problems that have vexed the solid modeling community for several decades. As an alternative, it is suggested that the SFF community evaluate existing modelers from the viewpoint of a sophisticated, library-level user rather than that of a purely interactive user.

No approach to improving the STL interface method will be problem free. However, the SFF community should be fully aware of the full capabilities of existing modelers before embarking on the SIF file design..

**Acknowledgments:** Vadim Shapiro assisted in soliciting interface specs from other software vendors. The XOX interface interface information was provided by Pradeep Sinha. Arlo Ames assisted with the ACIS information. Nikolay Shulga provided useful comments on the draft.

## Appendix: Obtaining cross sections from representative modelers

The following paragraphs provide a brief summary of the API by which customized software obtains cross sectional data from the the ACIS, Microstation Modeler and SHAPES modelers. An SFF site would need to precede these calls by code to obtain user input identifying the appropriate data and follow it by code to traverse the result geometry and output it to the file or real-time process.

**ACIS:** In a C++ interface to the ACIS solid modeler (Spatial Technologies, Inc, 2425 55th Street, Building A, Boulder Colorado 80301-5704), slicing information is obtained by calling the function

outcome api_intersect( BODY* tool, BODY *blank)

where BODY is the type for the boundary representation of a solid and outcome is a status indicator type. For SFF, the tool would be defined a the section plane (or a cube containing the section plane as its lower face)and the blank is the body to be sectioned.

**MicroStation Modeler:** In the Microstation MDL C environment (Bentley Systems, Inc., 690 Pennsylvania Dr., Exton PA, 19341-1136), the function

int mdlSolid_intersect( BODY *tool, BODY *blank)

is an interface to the ACIS function api_intersect noted above.

**SHAPES:** In the SHAPES modeler (XOX corporation, Two Appletree Square #334, Minneapolis, MN, 55425), slicing information is obtained by calling the function

xGEOM xGmInsct( xGEOM solidToBeSliced, xGEOM slicingPlaneOrSurface )

where xGEOM is the type for pointers to geometry items. The result will in general be a 2-dimensional geom that represents the cross-section.

# Solid Freeform Fabrication User Perspective: Medical Model Fabrication

Nicholas J. Mankovich

*UCLA School of Medicine*
*Department of Radiological Sciences*

## SURGICAL PLANNING AND IMAGING

The challenge of craniofacial surgery is to simultaneously improve the patient's physical condition and appearance. In considering the course of a craniofacial procedure the surgeon requires high-quality information. Today, computed tomography (CT) and magnetic resonance imaging (MR), when coupled with three-dimensional reformatting, reconstruction, and fabricated models provide premier planning tools for craniofacial, maxillofacial, head and neck, and plastic surgery. CT and MRI both produce closely spaced axial slices of patient anatomy that, when rejoined in the appropriate manner, fully describe a volume of tissue.

Each $512^2$ CT slice image is composed of pixels that represent a small volume element (voxel) of patient tissue sampled by the CT scanner – each being as small as .4 x .4 x 1.0 mm. A voxel intensity represents the average tissue x-ray attenuation in that small region of anatomy by assigning a number in the range of 0 to 4095. A typical 3D CT study is made up of 40 to 70 of these slices collected in a 15 to 30 minute CT examination. If the patient remains still during scanning, the slices are in excellent alignment with larger anatomical structures spanning many slices. If the CT slices are collected contiguously we can treat the collection of CT images as representing a 3-D image volume.

| | CT | MRI |
|---|---|---|
| MATRIX SIZE | 512 x 512 | 256 x 256 |
| VOXEL SIZE (mm) | .5 x .5 x 2.0 | .5 x .5 x 1.5 (gap) |
| TYPICAL DATA | 60 images (31 MB) | 120 images (16MB) |
| DENSITY RESOLUTION | 4096 levels (12 bit) | ≈128 levels (16 bit) |
| SIGNAL TO NOISE | high | moderate |
| SEGMENTATION | threshold | complex |
| PROTOCOL | simple (radiation) | complex (benign) |

Table 1. Summary comparison of basic imaging characteristics relevant to three-dimensional reconstruction for CT and MRI.

Recently, magnetic resonance imaging has become increasingly popular for its ability to produce images that show subtle differences in soft-tissue anatomy without the harmful effects of ionizing radiation present in CT and other x-ray examinations. Because of its emphasis on the hydrogen atom, cortical bone appears as image voids (black) in MR images and is thus indistinguishable from air. As an evolving image modality, MRI has quite complex acquisition protocols that offer substantial flexibility in the orientation and spacing of the voxels. Images can be acquired in any plane without respect to patient gantry orientation. Additionally, MR can acquire true 3D image volumes with cubic voxels. Table 1 compares clinical CT and MR imaging for 3D image reconstruction.

If imaged with 3D protocols, both CT and MR can be considered as exhaustive patient tissue samples (3D image volumes). It is the goal of 3D anatomical rendering to extract anatomy of interest and produce a realistic representation of the patient's pathology in relation to other structures of interest.

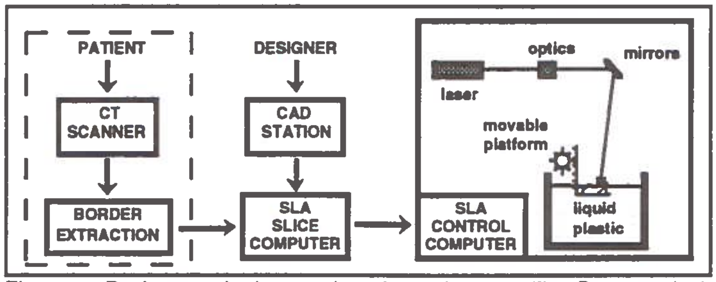


Figure 1. Basic steps in the creation of a patient-specific 3D anatomical model from CT scan data. The CAD design process used in conventional modeling is not used in the medical fabrication.

## CREATING CUSTOM MODELS AND IMPLANTS

Traditional method of fabricating craniofacial implants have relied on cephalometrics (careful measurements of patient cranial anatomy) combined with a meticulous patient examination. In the case of cranial implants, direct patient molds are created by pouring silicone and plaster directly onto the patient so that overall pre-surgical surface shape can be determined and implants designed to correct surface defects. These implants are typically made of a cured plastic (methylmethacrylate) custom-shaped to the patient's cranial defect. This conventional method relies on a combination of surgeon's preoperative skill in making the implant and his or her skill in shaping the sterile plastic piece during surgery. Because of the poor fidelity of the implant, they are generally made considerably larger than necessary and significant operative time is spent in shaping the plastic to fit.

The most recent improvement to this manual process relies on rapid prototyping technology to provide an extremely accurate rendering of anatomical structures. By directly creating a model of the bony defect from a digital imaging modality, the implant taken into surgery fits with little modification. Figure 1 shows the basic elements of the anatomical model fabrication process using stereolithography.

In the fabrication of a patient model the steps include (1) patient scanning with CT or MRI, (2) image processing to enhance edges or create cubic voxels from elongated voxels (optional), (3) segmentation to delineate and extract the anatomy of interest as a solid model, (4) model preprocessing to produce a support framework – often relying on sophisticated ray-tracing algorithms, (5) model slicing of merged model and support structure, and (6) model fabrication. Figure 2 diagrams the steps involved. Based on accuracy studies the process appears most limited by the imaging modality rather than the ability to fabricate from the surface description.

Unlike many rapid prototyping models, patient anatomical models do not rely on computer-aided design for their genesis. The design phase in anatomical modelling resides in the automated extraction of the anatomy of interest directly from CT or MR image volumes. Thus the normal CAD process is replaced by image processing algorithms. Although there are quite sophisticated methods to extract anatomical detail, the most common is a threshold-based segmentation applied to CT for bone extraction. In fabricating models of patient skull anatomy we employ a version of the marching cubes algorithm to construct a voxel-interpolating surface composed of oriented triangular surface patches.

## A BRIEF HISTORY OF INTERFACES

The computer-assisted creation of patient anatomical models began in the early 1980s with the use of bone edge-following algorithms to provide accurate bone/soft tissue boundary information directly from CT scans. The history of model fabrication has been a series incremental technology integrations, paced by the speed of data interchange development.

Initially, CT films were digitized by hand on a tablet and life-size boundary plots were used to guide manual band-saw cutting of thin sheets of plastic. These stacked plastic sheets, each one CT slice thick, were then used as a rough model of the patient's defect. As image interchange software was developed, the CT images were directly transferred to an analysis computer and algorithms were developed to

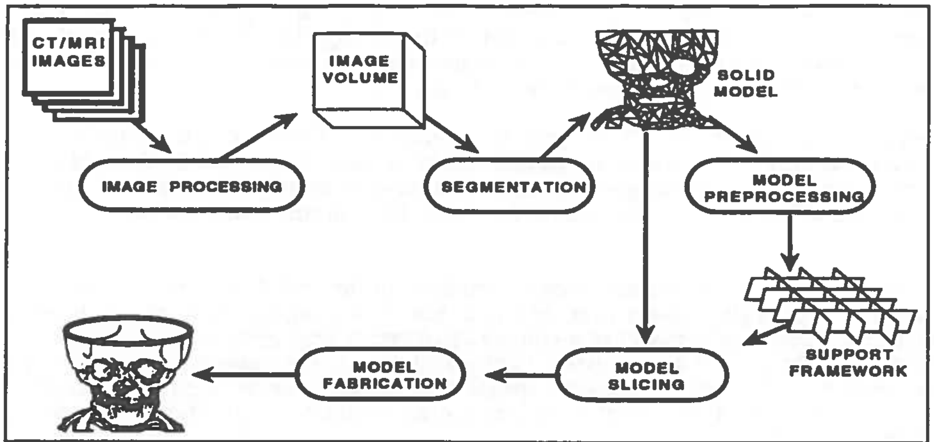



Figure 3. Schematic diagram of the process from patient scan to the fabrication of a plastic life-size model of patient anatomy using stereolithography. The solid model is a mathematical description of the bone surface as represented by 500,000 to 1 million triangles, each < .5 millimeter on a side. Integration with a support structure is necessary to avoid model distortion while in the initial stages of polymerization.

delineate the bone/soft tissue boundary automatically. The data were still plotted to life-size and plastic cut by hand.

As the medical imaging community learned about CNC machine tools, an interface was created to move the edge data directly into a format usable by the machine shop. This further required interfacing the edge description to a CAD package (i.e. AUTOCAD) allowing the slice outline to be modified prior to sheet cutting. The advent of stereolithography brought a new way to build models and the first efforts resulted in low-level interfaces directly to the slice fabrication software used by the stereolithography apparatus.

As computing power increased and as a de facto standard emerged in the STL file format, the anatomical models could be represented as fully tesselated 3D solid models. The STL solid model file provided a level of abstraction that allowed the medical modeler to concentrate on the anatomic extraction without worrying about the parameters of the building process. Of course, these were still important, but the STL format allowed the service bureau to worry about the details.

We are now at the point where we can simply and easily build custom patient anatomical models with stereolithography. In retrospect, much of the time spent in developing the software process from CT scan to model was spent in interface development. Working out the details of proprietary CT image formats, CNC coding, CAD porting, proprietary stereolithography slice format, and SLA format (originally proprietary). Along the way, in order to preview our STL models efficiently, we have also participated in the development of a graphics-display oriented format that allows efficient use of a particular solid model display device (the Silicon Graphics Graphics Engine).

## SFF PROCESS IMPROVEMENTS - MEDICAL MODELING CONSIDERATIONS

For the purposes of the SFF Workshop, I will conclude with some comments on the issues outlined in the call for position papers - from the perspective of a process user and de facto developer.

**Level of abstraction.** Medical modelling is always derived from some sort of slice-oriented image data (tomographic modalities). This does not mean that 2D, slice-oriented abstractions are acceptable in medical modeling. Although the simplest image segmentation algorithms are slice-by-slice these are no longer the accepted method of achieving the most accurate representation of true patient anatomy. All current methods rely on 3D information and, in general, produce a surface or solid model representation.

This allows appropriate algorithmic sophistication to occur in the appropriate data representation domain - either image or solid model. Because the transition from image to solid model discards a tremendous volume of information, it is important that the segmentation be able to take full advantage of image information to produce a best estimate of the true surface.

In considering the properties of 3D designs, the nature of CT/MR voxellated patient images leads naturally to a tessellated representation of the anatomical surface. It is certainly desirable to additionally handle the more general case of surface patches as great economies in representation might be realized in dealing with some of the more regular regions (e.g. smoothly curving bone) the rather smooth surface of bone.

In the Workshop spirit of speculation, medical models go beyond three-dimensional images in their ability to image 3D geometry rapidly over time (e.g. MR flow imaging, ultra-fast CT imaging). Just as the craniofacial surgeon uses a model of a complex fracture to help guide surgical planning decisions, a vascular surgeon might want to see a model of the vasculature as it changes during the cardiac cycle or a pulmonary surgeon might want to see how spatial relationships change during respiration. Although fanciful, physical models of this level of sophistication would have significant research potential in biomedical engineering.

**Type of model.** One of the most important features of medical model fabrication is the one-of-a-kind nature of patient models. There exists a considerable use of SFF in the biomedical engineering for orthopaedics but this prosthetic and appliance creation does not significantly differ from conventional part creation and will not be addressed here. The one-of nature of patient models clearly points to model technologies that are low cost with reasonable fidelity. Compromises in accuracy and precision should be dominated by the imaging modalities, not the fabrication techniques.

Because medical model fabrication is based on real rather than created geometry, the model characteristics in this domain might be classified in a manner similar to living tissue. This would include models of the different types of bony and soft tissue. The useful attributes are those that would provide increasing realism to the surgeon or the implant laboratory. This would include hardness, elasticity, color, texture, etc.

Beyond the ability to produce models of living tissue, if appropriate materials were developed, SFF should allow the direct creation of complex inplantable parts. This biocompatible fabrication could include both soft tissue and bony parts.

**Design tools.** Certainly one of the most useful design tools in medical fabrication is one that compensates for the free-form, organic nature of anatomy. Most especially crucial is the presence of support generation tools that know something about SFF material properties and can produce removable or easily delineated support structures. Because innovative imaging algorithms are often used to produce the solid model data, a proper design critic must be developed to verify the integrity of the model. As we are able to create more sophisticated models with color, texture, and possibly multiple materials, we may want 3D geometry tools to verify or exclude certain 3D adjacencies.

Model visualization would ideally provide rapid rendering of a model representation as a 3D display. This is especially important in avoiding building models with distorted geometry or containing various image artifacts.

Translation tools would allow straightforward movement between tessellated solids and solids based on surface patch descriptions. Smoothing and data reduction tools would allow a complex model to be simplified for data reduction and rapid proto-prototype production.

**Interchange format.** As mentioned above, interchange format is often the most time-consuming and seemingly least productive element of process creation. As a user and reluctant developer I would like to see a concise, efficient representation usable by the most advanced developers of design and fabrication methods. However, there must also exist, a simple entry level format that encourages those engaged in technology transfer into the world of SFF to experiment with new ideas for the use of this exciting technology.

# TOPOLOGY and FEATURES for SOLID FREEFORM FABRICATION

**May 30, 1995**

**T. J. Peters & S. A. Demurjian**
**Department of Computer Science and Engineering, U-155,**
**University of Connecticut,**
**Storrs, CT 06269 - 3155,**
**tpeters@brc.uconn.edu,**
**FAX +1 203-486-4817.**

**WORKSHOP on SOLID FREEFORM FABRICATION**
**CARNEGIE MELLON UNIVERSITY**
**June 5-6, 1995**

## POSITION STATEMENT:

Consider design as an optimization process via iterative refinement of a model. Advances in solid freeform fabrication (SFF) are providing novel opportunities to integrate design and manufacturing iterations.

This paper proposes that topological and feature considerations are fundamental to SFF. Further, it is advocated that the geometrical description appropriate to SFF should be a three-dimensional, feature based model--potentially instantiated from a more abstract, object-oriented model.

## INTRODUCTION and MOTIVATION:

The use of SFF to allow experimental analyses to be included in a timely, cost-effective manner within an iterative design loop providing feedback for modifications to the design geometry is an attractive prospect [SBK94], but prompts the question:

> * Might an edit suggested by investigation of a prototype cause an unintended topological change to the design model?

This topological concern is central because the topology of a model is often one of the most stable aspects of the iterative design process.

## SUPPORTING GEOMETRY:

The increasingly dominant geometric modeling mode relies upon 3D representations. Increasingly, supporting analysis, planning and critique software critically depend upon direct interfaces to 3D models. Having SFF rely upon a specialized 2D representation might be counterproductive to SFF efforts, effectively relegating it to a minor status--particularly if such 2D representations precluded access to contemporary 3D supporting design tools. It appears more desirable to profit from the 3D momentum and have the primary geometry basis for SFF be 3D, with appropriate tools to convert this canonical 3D representation to layered geometry.

## TOPOLOGICAL OVERVIEW:

The ``... de facto solid freeform fabrication industry standard..." [BoWo92] is a data representation consisting of triangulations of the boundary surfaces of a solid. This standard is called the .STL file format. The topological integrity of the data stored in an .STL file is crucial.

Quantifiable topological constraints should be included in integrated, iterative design so that the topological form of the nominal solid is preserved. These topological constraints should adhere to definitions of a strong form of topological equivalence previously proposed within the context of tolerance modeling [BoSt90a,BoSt90b,St93]. This will facilitate integration with those tolerance aspects. Such topological limits have been established for polyhedral geometry [ADPS94,Dor94], and we are participating in ongoing efforts to extend this topological stability theory to freeform geometry.

The intuitive summary relevant to present purposes is that [ADPS94] provides a mathematically verifiable means to locally edit a triangulation while preserving its topological form, including such fundamental properties as its connectivity, genus, embeddedness, homotopy type, etc.

Ensuring topological validity of an .STL file can entail some computational expense [BoWo92]. Appropriate invocation of topological constraints [ADPS94] under local geometric editing would preserve the topological integrity of the .STL file. This provides the opportunity to consider design optimization in a tight loop between experimental analyses upon SFF models and iterative design modifications by local geometric edits to the .STL file.

While the preceding discussion has considered triangulated faces, this dependency upon triangles may decrease in the future. An alternative file format (known as .SLC) now exists for freeform objects, but has not yet been widely adopted [Ja94]. Perhaps efforts to extend the topological constraints to freeform objects will converge with increased adoption of the .SLC format.

Already, there exist interfaces from 3D solid models to the .STL format and it is recommended that future efforts profit from this established base. Even while solids are becoming firmly established, it is further recommended to extend the modeling domain to non-manifold topology (NMT) modeling systems. This is a natural progression and may allow more graceful SFF interfaces, particularly with regard to models having self-intersections or internal voids, which are easily modeled in NMT systems. Such models are often precluded by solid modeling software systems, but may be realizable under SFF.

## FEATURES and ANALOGIES to VLSI:

The expression `the topology of the design' is often used informally, both in mechanical design domains and in VLSI. Hence, topology merits attention as a common unifying abstraction for SFF and VLSI. The commonality is that the topology of the design specifies

* how design elements are inter-connected.

Generally, in the electrical domain, the formal abstractions for design are much more mature than in the mechanical domain. These formalisms permit capture of rules for specifying these inter-connections within reliable software, as has already been successfully achieved for VLSI. Progress is being made in the mechanical domain with features, which are advocated as a critical foundation for formalizing the mechanical design process.

Continuing this abstraction theme, software design can, today, serve as the initial step for capturing conceptual design intent within an integrated design process for traditional engineering disciplines. The resultant software then becomes a `front-end' to a CAD/CAM system for geometric model instantiation, wherein features play a prominent role.

Within our software engineering environment, ADAM [PDTG94], object-oriented features are used to formally model and abstractly represent design intent. The features become fundamental objects within a design hierarchy within this object-oriented modeling environment. The modeling is performed via a graphics user interface, thereby freeing the design engineer from many syntactic details of code development. It may be possible to specify SFF layer based features at this abstract level as an initial graceful path for their inclusion in geometric models. Once the design has been captured, automatic code generators can be invoked to create many of the feature methods within a choice of standard languages. ADAM prompts the designer to supply design justifications, which are captured and serve as the basis for design checks within ADAM. We are currently developing a code generator for a LISP-like proprietary language that serves as the front-end to a commercial CAD/CAM system. This will afford the opportunity to go from conceptual design modeling to geometric instantiation in a CAD/CAM system. The automatic link to manufactured parts can be completed by utilizing features to transform the 3D CAD geometry into a representation supportive of SFF.

ADAM provides for language independent design. It also permits viewing a design at differing levels of abstraction. ADAM facilitates a broad conceptual view, whereas fine scale geometric changes may be more graceful within the accompanying CAD/CAM system. ADAM is quite portable and can easily be adapted to a variety of CAD/CAM systems, merely by creation of another specific code generator.

ADAM also supports propagation, which is a mechanism to specify and update relations between design features. A change in one design feature automatically sends a change notification to its related object, possibly even invoking accompanying software to undertake the responsive changes. These tools of object-oriented propagation have already been incorporated within some CAD modeling prototypes [PDTG94,Pe95]. Propagations could be supportive of a number of design critics to monitor iterative design changes.

As a specific type of design critic, it is often important to anticipate how design changes will effect the manufacturing process. The folklore that `small design changes often cause significant manufacturing changes' has been formally modeled in the domain of injection molded parts [PeRoS94,PeRo94,RoPe92]. Perhaps this formalism could serve as basis for providing manufacturability analysis software for SFF.

The proposed Federated Modeling Architectural [HoJu92] may be of interest as a model of a broad software framework to facilitate integration of many CAD tools for a VLSI-like environment.

## ACKNOWLEDGMENTS:

The authors acknowledge, with appreciation, partial funding for this work received under National Science Foundation Grant Number MII-9308346 and under ARPA Grant Number F29601-93-K-0126. The views expressed herein are of these authors, not of these funding agencies.

The authors express appreciation to L-E. Andersson, T. Bardasz, R. Disa, W. H. Dornfeld, S. M. Dorney, R. Garrett, D. C. Gossard, C. M. Hoffmann, J. W. Kane, E. J. Mintel, R. J. Peters, D. W. Rosen, J. A. Roulier and N. F. Stewart for helpful discussions. Any remaining errors are the authors' responsibility.

## BIBLIOGRAPHY

[ADPS94] Andersson, L-E., Dorney, S. M., Peters, T. J., Stewart, N. F., Polyhedral perturbations that preserve topological form, accepted for publication, CAGD. (This document also exists as University of Connecticut Technical Report CSE-TR 94-9.)

[BoWo92] Bohn, J. H., and Wozny, M. J., A topology-based approach for shell-closure, in IFIP Transactions, Geometric Modeling for Product Realization, eds., P.R. Wilson, M.J. Wozny, and M.J. Pratt, August 1992, pp. 297 - 320.

[BS90a] Boyer, M., Stewart, N. F., Modelling spaces for toleranced objects. International Journal of Robotics Research 10, No. 5, 1991, pp. 570-582.

[BS90b] Boyer, M., Stewart, N. F., Imperfect-form tolerancing on manifold objects: a metric approach. International Journal of Robotics Research 11, No. 5, 1992, pp. 482-490.

[Dor94] Dorney, S. M., Maintaining Equivalent Topological Form in Tolerance Modeling, Doctoral Dissertation, Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, 1994.

[HoJu92] Hoffmann, C. M, and Juan, R., High-level representation for geometric design and analysis, in IFIP Transactions, Geometric Modeling for Product Realization, eds., P.R. Wilson, M.J. Wozny, and M.J. Pratt, August 1992, pp. 129 - 164.

[Ja94] Jacobs, P., personal communication, October 3, 1994.

[Pe92] Peters, T. J., Encoding mechanical design features for recognition via neural nets, Research in Engineering Design, 4:(1), 1992, 67-74.

[Pe95] Peters, T. J., Integration of features and design interdependencies within CAD/CAM, accepted for publication, National

Science Foundation Design and Manufacturing Grantees Conference, January 4 - 6, 1995.

[PDTG94] Peters, T. J., Demurjian, S. A., Ting, T.C., and Glovin, S., Feature-based modeling by object-oriented design with propagation, Proceedings of 1994 International Conference on Data and Knowledge Systems for Manufacturing and Engineering, Hong Kong, May 1994, pp. 101-110.

[PeRo94] Peters, T. J., Rosen, D. W., and Dorney, S. M., The diversity of topological applications within computer aided geometric design, Papers on General Topology and Applications, Eigth Summer Conference at Queens College, (eds.) Andima, S. et al, Annals of the New York Academy of Sciences, Volume 728, pp. 198-209.

[RoPe92] Rosen, D. W., Peters, T. J., Topological properties that model feature-based representation conversions within concurrent engineering, Research in Engineering Design, 1992, No. 4, pp. 147-158.

[SBK94] Steinchen, W., Kramer, B., and Kupfer, G., Photoelasticity cuts part-development costs, Photonics Spectra, May 1994, pp. 157 - 162.

[Stewart93] Stewart, N. F., Sufficient condition for correct topological form in tolerance specification, Computer-Aided Design 25, No. 1, 1993, pp. 39-48.

# Comparison of the Requirements for SFF and VLSI Fabrication

Cesar Pina

This paper will address some similarities and differences between SFF and VLSI prototype fabrication from the viewpoint of someone in an IC prototyping brokerage service. The goal is to try to learn from the "VLSI experience" what might work in the emerging SFF field.

Digital electronic design has some general characteristics that are shared with other forms of design. An extension of the "VLSI experience" to a field such as that of SFF has to take these into account. The main characteristics can be summarized as:

1. **Structural hierarchy.** Views an object as parts composed of subparts in a recursive manner. For example, a television set is composed of groups of parts such as tuners, amplifiers, etc., and these in turn can be further subdivided. The bottom of the hierarchy is reached when all the parts are basic physical components. In VLSI, we have at the lowest level, transistors, then leaf cells, macros of various complexities, etc. depending on the particular design. Can a hierarchy be clearly defined for SFF?

2. **Different or multiple "views".** Each "view" contains an abstraction of the essential artifact. In the case of a VLSI circuit the Physical view is the collection of polygons which describe the geometry of the circuit on the chip. Structurally, the "view" is that of a collection of logic gates. Behaviorally, the "view" is a set of operational restrictions in some hardware description language. What are the equivalent "views" in SFF?

3. **Connectivity.** All electronic components have wires connecting them to other components. Every component is therefore interconnected with all others through some path. The collection of paths through the circuit is its topology. This is not always present in mechanical artifacts such as aircraft or buildings, which have optional connectivity and often contain totally unrelated components (Such as plumbing and structural support members).

4. **Dimensionality.** All design disciplines have a spatial dimensionality in which they are significant. In VLSI design, the manufacturing processes control the third spatial dimension (Junction depth, oxide thickness, - metal thickness, etc.) and these are apparently transparent to the designer, who is laying out the circuit in two dimensions. The designer, however, cannot ignore the third dimension, whose effects are manifested as capacitances, sheet resistances. The designer may also be implementing primitive cells, in which case the work is being done in layers, (diffusion, polysilicon, metal, etc.). If he/she is working at a higher level of abstraction, the CAD system is generating the layers when cells are instantiated. This is obviously very important in SFF.

The status of SFF fabrication and CAD tools today, is somewhat reminiscent of the status of VLSI design and fabrication in the 1978 - 1982 period. At that time, there were a large number of available technologies to choose from (and no clear winner): NMOS, PMOS, CMOS, SOS, IIL, BJT. Feature sizes were in the 5um region and design tools and methodologies were very primitive. During this period, a number of developments took place that may be applicable to SFF processes today. Among these, was the development of geometrical description languages and interchange formats, (some of which are still in use today) which made it significantly easier for designers to access the fabrication lines. Initially, designers were forced to work at the lowest level of abstraction, e.g., at the transistor level. As the CAD tools improved and both designers and the CAD manufacturers developed libraries which enabled the designers to work at higher levels of abstraction. The IC CAD systems have to produce an output which can be used by the manufacturing equipment the designer needs to access. However, there are a large number of different manufacturing and test systems, each with its own particular set of requirements and input/output formats. These requirements are often the result of proprietary designs which the manufacturer may not wish to share with designers at large. This led to the development of interchange formats, to provide the connection between the CAD system and the manufacturing equipment. A brief description of the most important IC interchange formats developed during this period is given below. All are still in use today. What features of these formats can prove useful in SFF today?

**CIF (Caltech Intermediate Form).** This format consists of a set of textual commands to set layers. It is concise and is human and machine readable. It describes only the geometry on IC photomasks in terms of rectangles, polygons, wires and circular pads. No higher level graphics such as text is provided. It has hierarchy, and is therefore easy to generate and can be used to build complex graphics. The language is structured, therefore allowing for multiple CIF files to be aggregated. It is an interchange format, and is not readable by the mask manufacturing equipment. CALMA GDSII. This intermediate format is older than CIF, and because the of the profusion of CALMA design systems, it became a defacto standard in the industry. GDSII is the complete database representation for CALMA CAD systems, and not just an output spec for mask making equipment. This makes it a more complicated format than CIF, and is therefore harder to use. It is binary, hence not human readable.

**EDIF. (Electronic Design Interface Format).** More recent than CIF or GDSII. It is a textual format, resembling LISP. Each statement is enclosed in parentheses and has a keyword first, followed by its parameters. The entire file is structured, so that all statements are parameters to other statements. EDIF files can be hierarchical, and can contain any form of information, including topology, geometry, and behavior.

Use of one of these interchange formats, serves as the common denominator in the description of different projects. Regardless of the form of the original input, the designs can be translated into one of the interchange formats before being translated again into a variety of formats required for different purposes. This concept is one that can be profitably applied to SFF prototyping, the development of an interface format, such as CIF or EDIF. The three interchange formats above cannot by themselves be used to drive the fabrication or test equipment used for manufacture and test of ICs. As a comparison, the format used to drive the E-beam mask generating equipment is briefly described below.

**EBES (Electron Beam Exposure System Format).** Developed by Bell labs, it's directly readable by mask making equipment. It allows no hierarchy, only three different types of figures: rectangles, parallelograms, and trapezoids. The geometry must be sorted so it's presented to the equipment in spatial order. It is clipped into 1k x 32k stripes (a "segment") and up to 255 segments can be extended in the y direction, and is arbitrary in the x direction. Rounded figures must be approximated by polygons and complex polygons are broken into four sided figures.

The advent of the intermediate format descriptions, made it possible for the designer to generate a fabrication independent geometrical representation of the circuit in CIF or CALMA GDS II formats and to submit these descriptions to a "broker" or IC "foundry." The broker or foundry then converted the designer's geometrical representation into aformat that could be used by the mask making equipment to generate the required phototooling. The conversion from CIF to the machine usable format to actually manufacture the masks, was transparent to the designer, who was left to concentrate on the intricacies of his design. This approach, involving the development of a geometric representation, with a textual format, that could be independent of the process being used, has been very successful in the case of VLSI. The use of CIF, has enabled the ARPA/NSF research community to make relatively simple transitions for designs in different technologies such as NMOS, CMOS, Bipolar, GaAs circuits through the generation of different layers. This approach appears to be one that should be developed for use in SFF. A geometric description format, such as CIF, describes objects in terms of "boxes", "polygons", "layers", in essentially two dimensions, Nothing, however, prevents us today from adding additional dimensions to the arrays used in CIF to describe two dimensional objects. Instead of the CIF description of a box as [ length width xpos ypos xrot yrot ] we could define a layer as [x1 x2 x3 f(x1) f(x1,x2) ....].

# CAD SUPPORT FOR SOLID FREEFORM FABRICATION

Herbert Voelker

## Abstract

The families of solid freeform fabrication (SFF) technologies now emerging from research exhibit several distinctive characteristics.

C1 - All offer rapid prototyping (direct 'art to part'), but this is not unique to SFF processes.

C2 - Some offer topological novelty, i.e. the ability to produce parts not conventionally manufacturable, such as parts totally contained within other parts.

C3 - Some offer anisotropic material properties, including opportunities to embed other objects and materials, that are more flexible and controllable than are attainable with convenional processes.

C4 - Most require process command and control information that is spatially partitioned, e.g. layered, and in some cases, directional within layers.

Most modern CAD systems are based on principles that guarantee 'geometric completeness' through an underlying mathematical model of solids dubbed 'r-sets'. One of the axioms underlying the r-set formalism is material (set interior) homogeneity, and thus r-set-based systems ostensibly cannot support C3 and other characteristics governed by C3 — usually C4, and sometimes C1.

This talk will review the known properties of complete rep schemes for solids through an extended version of Requicha's 1980 taxonomy, with the aim of identifying rep schemes that can be extended easily to support SFF technologies. The candidates that emerge are schemes that offer explicit representations of a solid's interior — notably spatial enumerations, cell decompositions, ray representations, and two function-based schemes. Interestingly, this set does not include the scheme most widely used in contemporary CAD systems — boundary representations — or its primary competitor, CSG (constructive solid geometry).

The talk will also raise a doctrinal issue: should definitions (of parts) that REQUIRE processing information for completeness be admitted as archival definitions, or should the current doctrine, which prohibits process specifications in part definitions, be retained?

## *A Taxonomy of Complete Representation Schemes for Solids*

< HBV: 9/93; rev. 2/94, 2/95, 6/95>

| *PARADIGM* | *FAMILY NAME* | *VARIANT* | *MATHEMATICAL MECHANISM* | *ACCESSIBLE ATTRIBUTES* | *SPECIAL PROPERTIES* |
|---|---|---|---|---|---|
| Instantiation | Templates[1] | | Special-Case Formulae | | |
| Cell Composition (Gluing) | Spatial Enumeration[2] | Linear subdivision of space ⇒ rectilinear grids of equal-size cells | Union of Quasi-Disjoint Solid Cells<br>Solid = $\cup C_i$<br>where<br>$C_i \cap_h C_j = \varnothing$ , $i \neq j$ | Interior, Discretized into Cells | Spatial Addressability |
| | | Recursive subdivision ⇒ hierarchically graded cells (quadtrees, octrees) | | | |
| | Cell Decomposition | Simplicial or Cell Complexes; elements produced by object partitioning (e.g. triangulation) | | | |
| | Boundary Represenations (B-reps) | Exact boundaries | Union of Boundary Cells<br>$\partial$ Solid = $\cup$ Face$_i$<br>$\partial$ Face = $\cup$ Edge$_j$<br>(Solids thru boundary determinism) | Boundary;<br>Surface 'Features' | |
| | | Linear approximations ('tilings', 'facets') | | | |
| Boolean Composition | Constructive Solid Geometry (CSG) | Halfspace primitives | Regularized boolean composition | Solid primitives; syntactic structure | Recursive partitioning |
| | | Bounded solid primitives | | | |
| Generation | Sweep Representations | Simple sweeps: S constant, M a single rotation or translation | 'Infinite Union':<br>Sweep(S, M(t)) =<br>$\cup S(t_i) @ M(t_i)$, $\forall t_i \in$ [interval] | Generator Elements (S, M) | (Limited domains) |
| | | General sweeps | | | |
| | Medial Axis Transforms (MATs)[3] | | $\cup$ Spine(p) $\oplus$ Sphere(R(p)) | Spine (skeleton); Radius function | |
| Spatial Sampling | Ray Representations[3] | | Induced cell decompositions | Ray-sampled interior | Directional sampling |
| | Parametric Sample Sets[3] | | Interpolation | Sampled boundary | |
| Real and Vector Functions | Parametric-function Representations[3] | | Homeomorphic mapping over finite domains | Interior | (Limited domains) |
| | Real-function Representations[3] | Differentiable Indicator Functions | $f(p) > 0 \Rightarrow p \in iS$<br>$f(p) = 0 \Rightarrow p \in \partial S$<br>$f(p) < 0 \Rightarrow p \in cS$ | Point classifications | Boolean composition through arithmetic operations |

[1] Called 'Pure Primitive Instancing' in Requicha's paper in *ACM Computing Surveys*, December 1980.
[2] Called 'Spatial Occupancy Enumeration', without linear and hierarchical distinctions, in Requicha's 1980 paper.
[3] Not acknowledged in Requicha's 1980 paper.

## Shape Deposition Processing

Lee Weiss
Carnegie Mellon University
and
Fritz Prinz
Stanford University

**Position Statement**

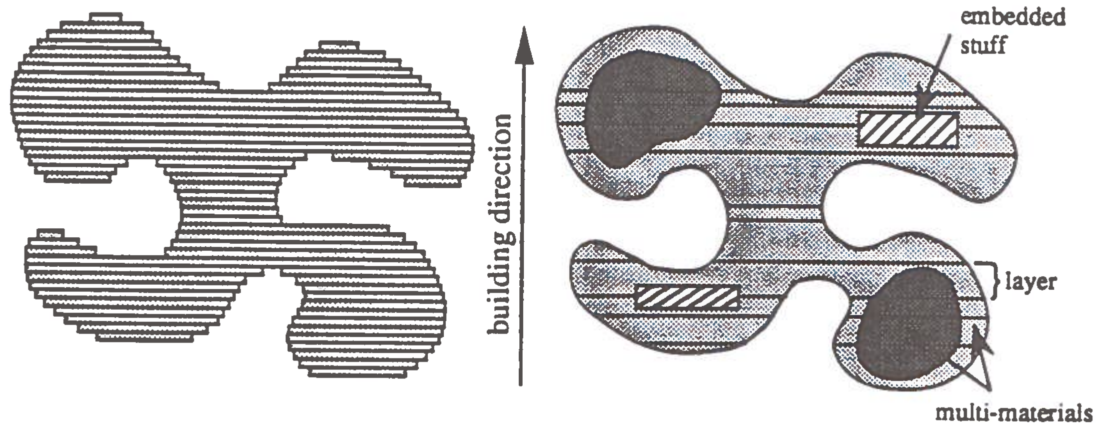
"Solid freeform fabrication is the production of freeform solid objects directly from a computer model without part-specific tooling or human intervention."[1] The planning and execution approach underlying the majority of SFF processes in use today is to first decompose a 3D CAD model of the object into cross-sectional layers, followed by the use of material deposition techniques to physically build up these layers to form the object. Sacrificial supporting layers may also be simultaneously built-up to fixture the object as required.

The layer topology and material composition vary with different SFF processes. For one example, Fig. 1A depicts an SFF object decomposed into 2-1/2 dimensional layers (i.e., each layer is represented by a planar cross-section with an associated uniform thickness). It is a homogeneous structure consisting of a single material. One advantage of this type of structure is that the process to build it is rapidly planned and executed *independent of the object's geometry*. There is a trade-off, however, between layer thickness and surface quality.

As a more general example, Fig. 1B depicts the same shape decomposed into three dimensional (3D) cross-sectional layers of varying thicknesses (i.e., the outer surface of each layer maintains the 3D geometry of the original model). It is a heterogeneous structure composed of multi-material deposits, as well as pre-formed objects which can be embedded into the growing shape. This structure exhibits two important properties. First, such heterogeneous shapes may not be practical or even impossible to fabricate with conventional forming techniques. Second, the surface does not exhibit the 'stair-step' appearance as in Fig. 1A. These advantages, however, come with a price; ***the process planner requires the full 3D geometric modeling information throughout the planning steps, and the model will require attributes such as material composition, microstructures and representations of embedded components. While the final building plan will be dependent on the part's***

[1] D.L. Bourell, J.J. Beaman, H.L. Marcus, J.W. Barlow, "Solid Freeform Fabrication: An Advanced Manufacturing Approach," in proc. of The Solid Freeform Fabrication Symposium, ed. J.J. Beaman, H.L. Marcus, D.L. Bourell and J.W. Barlow, Univ. of Texas at Austin, Aug., 1990, pp. 1-7

*geometry, proper shape decomposition will assure that a successful building plan can be automatically synthesized. Representations which label features, such as fillets, would simplify the requirements for automatic extraction of features such as such as principal curvatures. Design critics could also provide useful feedback during the design process about the manufacturability of alternative designs.*




**A.** 2-1/2D layered, homogenous **B.** 3D layers, heterogeneous structure

**Figure 1.** SFF structures (cut-away views ) based upon layered shape decompositions .

Examples of heterogeneous structures include multi-material tooling, embedded electronics, and scaffolding for human tissue engineering. CMU and Stanford are investigating SFF of embedded components and multi material structures using Shape Deposition Manufacturing (SDM) processing.

## SDM Process Description

SDM is an SFF approach for building heterogeneous, 3D layered structures. Each layer in an SDM structure consists of primary material(s) (i.e., the material(s) forming the part being created) and complementary shaped sacrificial support structure material which is removed when the entire part is completed.

SDM integrates several processing stations which operate on each layer (refer to Fig. 2). The two primary stations are for deposition and for shaping. Individual layer segments are deposited as near-net shapes and then machined to net-shape before depositing more material. As will be described below, shapes are adaptively decomposed into layer segments, or 'compacts', such that undercut features are not machined, but formed by previously shaped segments.

The growing structures may also be transferred to other processing stations where additional operations are performed on each layer. For example, shot peening can be used to control the build-up of internal stresses induced by thermal deposition processes like welding.

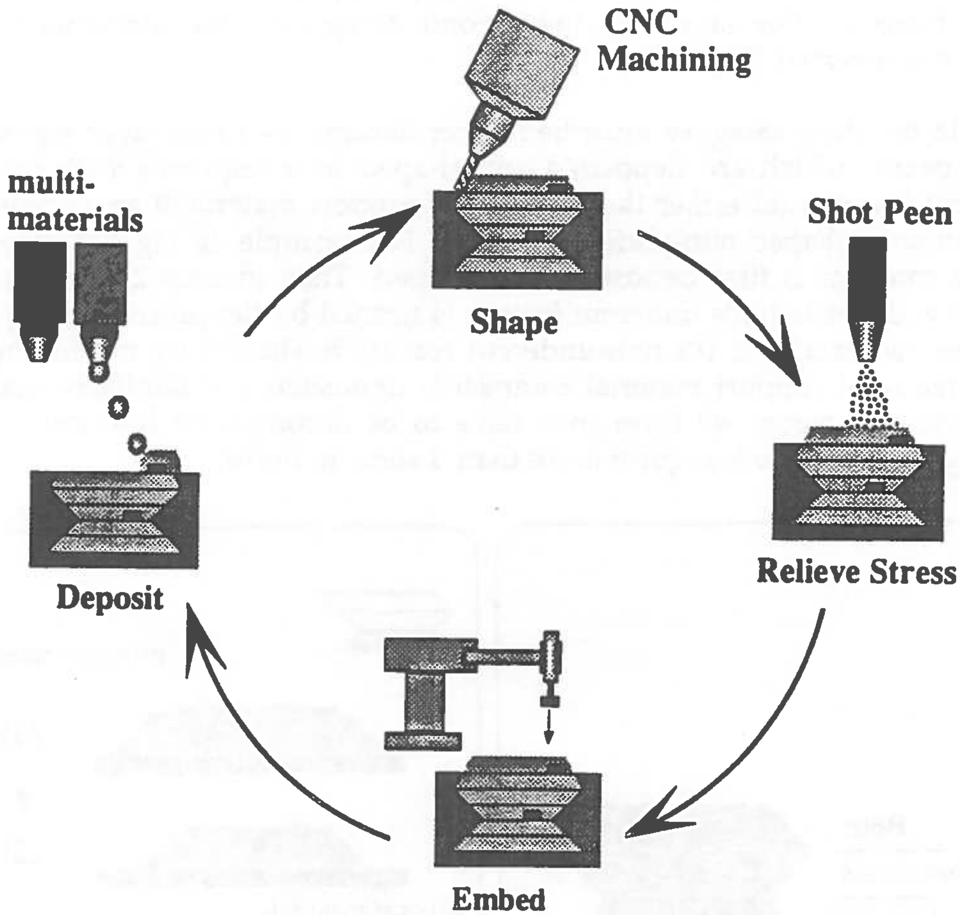



**Figure 2.** Shape Deposition Manufacturing

**Adaptive Shape Decomposition**

The basic sequence for building a single-material part is described below (see Fig. 3). In general, any shape can be decomposed into layers of primary material which can be characterized by one of three categories (see Fig. 3A):

- Category 1 - The layer has no under-cut features (relative to the intended building direction), or
- Category 2 - the layer only has under-cut features, or
- Category 3 - the layer has both under-cut and non-undercut features.

Straight-wall features are considered either as under-cut or non-undercut features depending upon subtle processing steps.

The thickness of each layer will therefore vary and the sequence for depositing and shaping the primary and support materials in each layer also vary based upon part geometry. For layers in the first category, the primary material is deposited first (Figure 3B, step 1) and then machined (step 2). The support material is then deposited (step 3), then the entire layer surface is planed (step 4). For layers in the second category, the aforementioned sequence is reversed (Fig. 3C).

Layers in the third category must be further decomposed into layer segments, or 'compacts', which are deposited and shaped in a sequence such that all under-cut features (of either the primary or support materials) are formed by the previously shaped non-undercut feature. For example, in Fig. 3D, step 1) a support compact is first deposited and shaped. Then in step 2, the primary material is deposited; it's undercut feature is formed by the preceding support structure compact, and it's non-undercut feature is shaped by machining. In step 3, the final support material compact is deposited and similarly shaped. In general, a category #3 layer may have to be decomposed into more than three compacts and will require more than 3 steps to build.

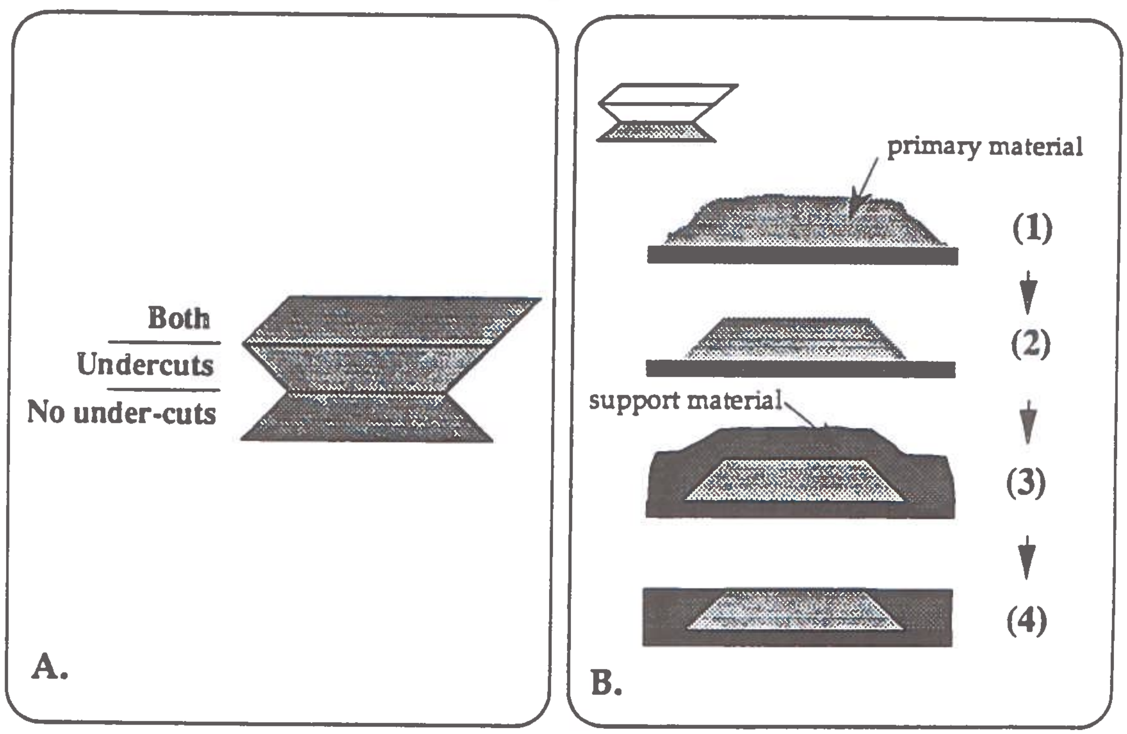



**Figure 3.** Basic sequence for depositing and shaping layers (continued on next page).

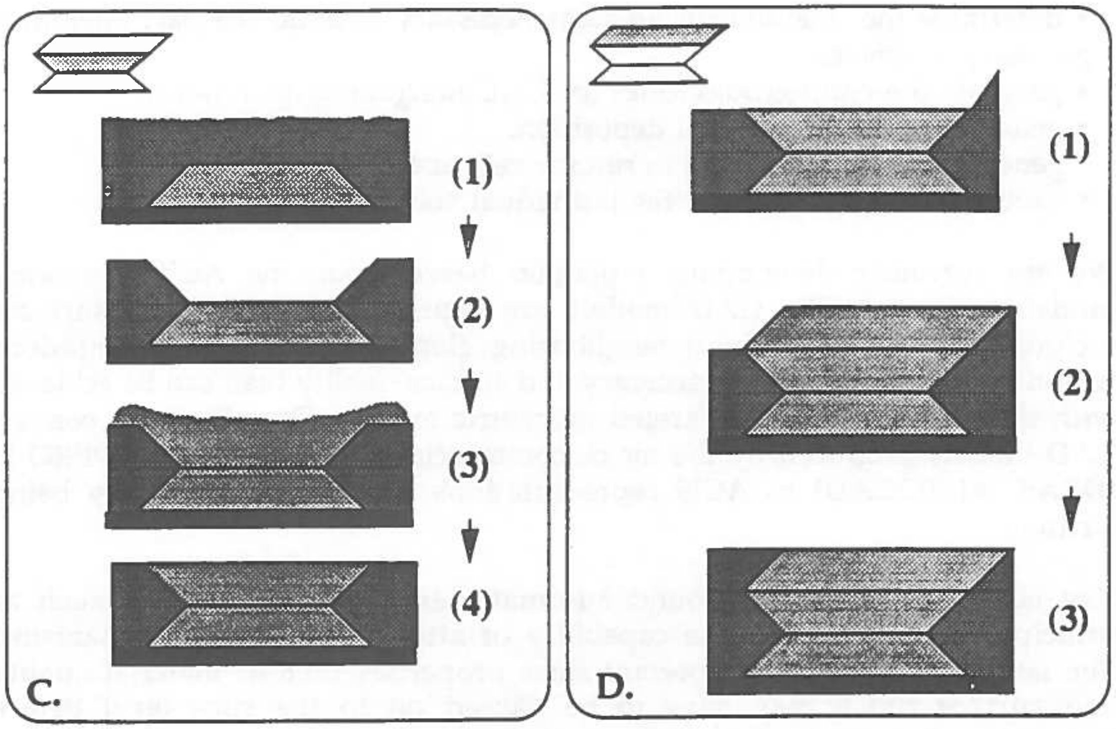



**Figure 3 (continued).** Basic sequence for depositing and shaping layers .

For parts which include layers with more than one material, such as depicted in Fig. 4, each of these layers are simply built according to the category #3 sequence described above.

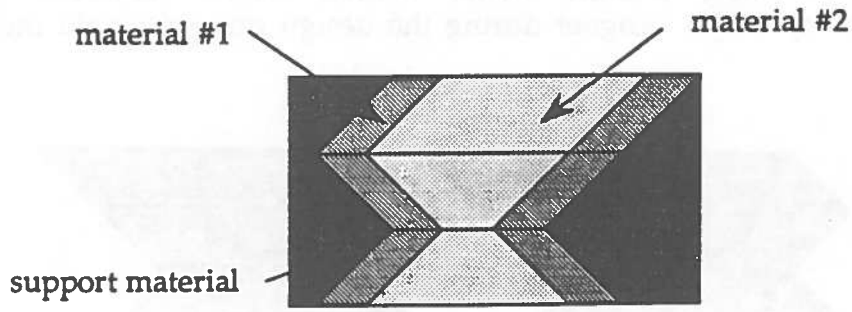



**Fig. 4.** A multi-material structure

**SDM Process Planner Requirements**

Given a CAD model of the desired part, a CAD/CAM planning and control system is required for the SDM process to automatically:

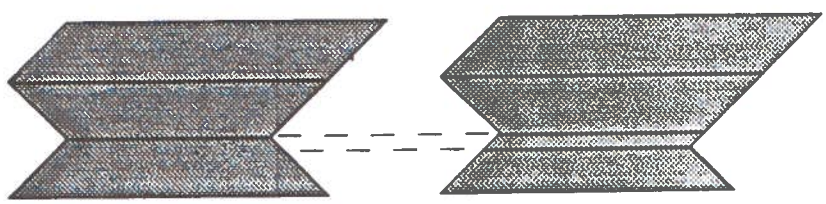
- adaptively slice the parts ,

- determine the manufacturing steps necessary to build the part (including process parameters),
- generate the cutting trajectories for CNC machining operations,
- generate paths for material deposition,
- generate the code required to run the cell, and
- execute the commands on the individual stations.

We are currently developing a planner based upon the ACIS geometric modeling kernel. The CAD models are represented as NURBS surfaces including information about neighboring elements. These representations will ultimately lead to better accuracy and surface quality than can be achieved with the traditional planar faceted geometric models. Translators to convert CAD models produced by the major commercial CAD systems (e.g., PRO-E, IDEAS, AUTOCAD) to ACIS representations exist or are currently being written.

Key issues are centered around automatic extraction of features such as principal curvatures and the capability of attribute handling mechanisms. The latter is particularly important since properties such as material quality and surface finish may have to be passed on to the slice level before generating all necessary processing commands.

Another issue is that small changes in local geometry may have significant effects on the manufacturing process. For one example, while the structure in Fig. 5a. can be built with 3 layers, a small change in it's geometry will require an additional layer as shown in Fig 5b. The 4 layer structure will take longer to build. Providing design critics which could feedback about manufacturability to the designer during the design process would therefore be useful.



**A.** 3 layers **B.** 4 layers

**Figure 5.** Small changes in design have significant effects on manufacturing.