**Proceedings of DETC98/CIE:
18th Computers in Engineering Conference
September 13–16, 1998 — Atlanta, Georgia**

DETC98/CIE-5527

TRADEWIND: A PROTOTYPE INTERNET MARKETPLACE FOR SOLID FREE-FORM FABRICATION

Sian Tan
Center for Design Research
Stanford University
560 Panama Street
Stanford, CA 94305-2232
Email: tan@cdr.stanford.edu

Alan Rothkopf
Computer Science Department
Stanford University
Gates Building
Stanford CA 94305-9045
Email: ar@cs.stanford.edu

Miguel Pinilla
Rapid Prototyping Lab
Stanford University
Bldg.530, Room 128
Stanford, CA94305-3030
Email: jmp@stanford.edu

ABSTRACT

TRADEWIND is a prototype electronic marketplace for SFF processes. This marketplace facilitates the interaction between designers and manufacturers from the initial request for bids to build a design up to transaction closure. The system is built with three distinct layers of functionality corresponding to the market functions of presence and accreditation, bid resolution and transaction closure. The bid resolution process collects bids from a number of manufacturing centers and performs a manufacturability analysis to compare a proposed design with the manufacturing capabilities of each of them. To support this manufacturability analysis we present a prototype parametric language to represent the parts to be built and the manufacturing constraints called “ShopTalk”.

INTRODUCTION

Solid Free-form Fabrication (SFF) processes fabricate mechanical parts by near-net shape incremental deposition of material. A particular instance of this kind of process is Shape-Deposition Manufacturing (SDM) [Merz 1994], briefly described later in this paper. We consider SFF processes to have an advantage over more conventional manufacturing techniques. In the general field of mechanical part design, manufacturability analysis and design interfaces have proven to be very hard problems to solve. The application of SFF offers an opportunity to reduce the problem complexity by decomposing the part to be built into simpler geometrical entities, such as 2-1/2 D layers or 3D compacts (See below). The simplification allows for a formalization of the manufacturing process and the design description. A successful example of the benefits of such simplification is the VLSI industry, where a simple decomposition scheme (masks composed of polygons) is able to represent very complex designs. It is this possibility of automating the analysis and information transfer that opens the opportunity to create an electronic marketplace for SFF.

TRADEWIND is an exploration of issues involved with the marriage of Internet commerce with Agent-based Design and Manufacturing in the SFF domain. When automated agents enable interactions between designers and manufacturing centers, these agents need a framework in which to conduct business with one another. TRADEWIND attempts to define architecture for such a framework. It is our belief that this work can be leveraged for use with methods of automated manufacturing other than SFF.

The type of networked infrastructure represented by TRADEWIND is essential in order to integrate operations between geographically separated, independent parties involved in the creation of artifacts via Solid, Free-Form processes. In addition to the expected features of a virtual marketplace, such as presence and accreditation of participants, and a transaction/bidding protocol, TRADEWIND also allows for the negotiation of manufacturability issues.

PROBLEM DESCRIPTION

The need for a marketplace

In the current structure of the rapid prototyping industry, service bureaus provide part-building services to customers who create the designs. They perform their tasks in an artisan-like fashion with heavy interaction between designers and manufacturers. This interaction promotes longer-term relationships between designers and manufacturers. As a result, competition among service bureaus tends to be based more on these relationships than on the value of the service provided. As the design/manufacturing interface becomes more streamlined, there will be less of a need for engineers to interact so closely, resulting in a more agile market for these services. Such a market cannot exist, however, without an effective trading infrastructure.

Marketplace Characteristics/Roles

A market infrastructure has to provide the following key elements:

a) Presence & Accreditation of Participants

In order to function effectively in a marketplace, participants must be able to communicate with each other. It is particularly important that participants know at some point with whom they are communicating and whether the other parties can be trusted to follow-through on contractual arrangements.

b) Bidding Protocol

The key activity in any marketplace is bidding. The key role of a marketplace is to join together those who wish to buy and those who wish to sell. An added feature of the market part-building services is the need for interaction between designer and manufacturer in order to resolve issues of manufacturability. There are 3 possible approaches to this problem: Finding a solution that satisfies a set of constraints, finding an optimal exchange, or allowing the participants to find a negotiated solution by themselves. TRADEWIND uses a combination of the first two approaches under the assumption that the SFF interface between the designer and the manufacturer will be rich enough to obviate the need for a complex protocol of proposals and counter-proposals.

c) Transaction Closure

When participants in a marketplace commit to a transaction, they must be assured that other participants have committed equally to that transaction under mutually agreed terms. It is a goal of TRADEWIND to provide an environment in which transactions are trusted. We see the key elements of trust being atomicity, non-repudiability and security.

Summary of SDM

The Shape Deposition Manufacturing (SDM) Process has been designed to directly produce functional prototypes from CAD models. It utilizes the benefits of layered manufacturing methodologies used in Solid Freeform Fabrication, the superior material quality of welded or cast structures and the accuracy of CNC machining processes. To manufacture a part the CAD model is decomposed into slices (layers). Each slice is manufactured in alternating deposition and shaping steps.

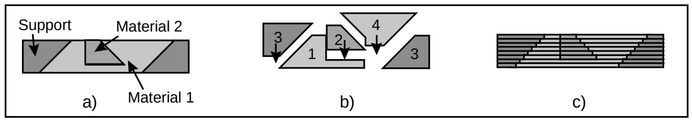

Figure 1: A two-material part embedded in support structure (a) can be represented with compacts [Merz 1994] (b) as decomposed features for SFF processes with 3D compacts, or (c) as layers for processes with 2½D layers.

While most currently available SFF processes use uniform 2½D layers, processes under development, like the Shape Deposition Manufacturing (SDM) process at CMU and Stanford [Merz 1994], use 3D layers. Therefore, a 3D representation of decomposed features is required for the general SFF class. As an example, we are pursuing a description of decomposed features based on “compacts,” which are single-material object segments that maintain the 3D geometric information of the outer surface of the object and have no directly-machined undercut surfaces [Merz 1994]. Figure 1a illustrates a simple two-material part (or perhaps a horizontal slice of a larger part) with support structure. Figure 1b illustrates the decomposition into SDM compacts. Figure 1c shows the decomposition into a layered representation required by commercial SFF processes. Constraints can be specified for each of the compacts and on local combinations of compacts to ensure manufacturability.

SDM provides the possibility for complete automation of manufacturability analysis, process planning and part building. There is an ongoing effort to provide a design description interface for SFF processes [Rajagopalan et. al. 1998]. TRADEWIND aims to leverage these characteristics to enable access to SDM services in an electronic commerce environment.

ARCHITECTURE

The system we have developed is a 3-tier system as shown in Figure 2. The three tiers of the architecture roughly correspond to the three major roles of a marketplace as described above. The system uses Java RMI objects and a specification language that we call ShopTalk. At the lowest level, we have the clients at the Transaction Layer, which comprise the Design & Manufacturing centers. In the middle level, we have a system of brokers that act as intermediaries between parties at the lower tier (the Transaction Layer). At the highest level, we have a Registry and a Directory Database, which comprise the Registration and Directory Layer.

TRADEWIND – Phases of Development

With current globalization, and communication facilities, it is natural to address the creation of such a marketplace electronically. What we propose is an Internet based marketplace that supports the functionality outlined above while being simple, easy to access and scalable.

Phase I:

The aim of this phase is to provide a computational infrastructure that will allow for presence and local accreditation of participants. It will also provide a non-secure communication infrastructure to enable the basic bidding protocol. Finally, Phase I will implement a form of bid resolution limited to constraint checking.

Phase II:

This phase will build upon the work done in Phase I, and will include the ability to express Utility functions of participants. The expression of these Utility functions is a matter of continuing research at Stanford. The system will be further enhanced to improve security, reliability and scalability.

In this paper, we focus on the work done in Phase I.

Registration & Directory Layer:

The Registry is responsible for system agent registration and authorization, corresponding to the marketplace's role of providing presence and accreditation of participants. It issues a unique system identity to each participating party and it keeps an internal list of all agents within the system which it stores persistently in the Directory Database.

The Registry informs the brokers at the Trading Layer of sign-ins and sign-offs of authorized agents at the Design & Manufacturing centers, for the purposes of processing job requests (from Design Centers) and bid requests (from Manufacturing Centers).

The Registry is responsible for providing agents at the Transaction Layer a list of all brokers in the system that those agents can use to choose brokers. If the Registry is temporarily unavailable for some reason, agents in the Transaction Layer can obtain information about brokers directly from the Directory Database.

Trading Layer:

The Trading Layer implements the bidding protocol for the marketplace. It is responsible for receiving job requests from the Design Centers. In addition, it provides the Design Center which originates the request with a list of Manufacturing Centers capable of accepting the job (based on the requested terms with regard to time, surface finish and other constraints). If no manufacturers are capable of accepting the job, the broker passes on information about why the job was unacceptable to specific manufacturers. In some cases, this information could allow a designer to revise a job sufficiently to be acceptable to some or all manufacturers.

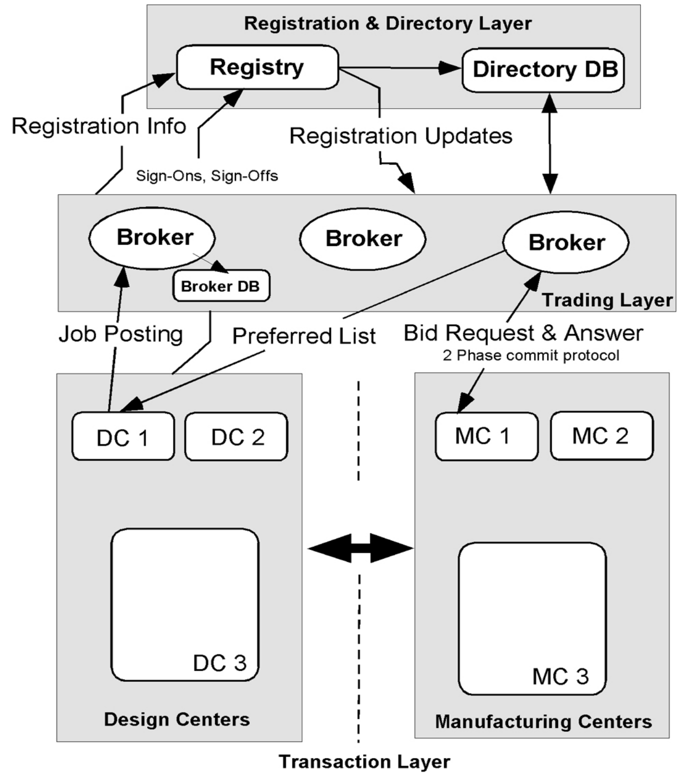

Figure 2: Architecture Schematic

A constraint resolution process accomplishes the selection of suitable manufacturers for a given design. This process is based on defining the part to be manufactured in terms of a number of parameters, which will be determined from the design of the part. While this description lacks geometrical detail, it allows for an automated treatment of the manufacturability analysis of the part in question.

In such a representation, the parameters define a design space in which the part to be built is represented by a point. The parameters evaluated from the part are expressed in 4 basic types: real numbers, integer numbers, sets of real numbers and sets of integer numbers. We impose a further restriction on these sets to be finite so the problem remains tractable. For enumerated values, including binary decision parameters, they are converted into integers.

With the use of parameters to describe part characteristics, one can formulate the problem of manufacturability, using the evaluated properties, as one of finding whether the design submitted lies inside the parametric region that represents a particular manufacturer's capabilities. Once the design is represented in parametric form, the designer may express alternative designs by submitting not only a point in the design space, but a set of constraints that represent the desired design region. This form of expression allows a designer to indicate constraints such as "material hardness should be between 20 and 40 RC" or "Delivery time needs to be between 2 and 5 days".

These two constraint sets, the manufacturer capabilities and the designer requirements, are two hyper-regions in the parameter space. As such, they can be intersected. If they do indeed intersect, the design is manufacturable, and the intersection region is called the "Mutually Acceptable Region" (MAR).

For the implementation in Phase I of this effort, the system is restricted to work with Mixed Integer Linear Constraints, resulting in polyhedral hyper-regions that can be intersected numerically.

For Solid, Free-form Fabrication (SFF) processes, two levels of parametric evaluation are possible: one for the whole part (e.g. max dimensions) and another for the sub-part (see description of Shape Deposition Manufacturing below) level. Manufacturability analysis of each of the elements that make up a part can reveal most of the manufacturability problems that may arise (e.g. features that are too small, unavailable material, etc.) for a part. Our process of parameter evaluation leverages the simplification of geometry by the decomposition of a part into simpler components – in our case, into compacts.

To provide this functionality, a broker will poll the Manufacturing Centers at the Transaction Layer with the job request and will compile, process and finally return bid information from the Manufacturing Centers to the Design Centers in a secure manner. At this juncture, once a Design Center has selected a Manufacturing Center from the list of choices, control of the transaction is offloaded to the Design Center originating the request and the Manufacturing Center selected to perform the task.

Transaction Layer:

Communication in this Layer provides the means of transaction closure within the marketplace. Closure takes place via a one-to-one interaction once the bid has been resolved and the Design Center (the client) has chosen a Manufacturing Center (the supplier). This direct interaction between a Design Center and a Manufacturing Center is based on a preliminary agreement on the part to be built, its cost and delivery terms. In an idealized environment the only remaining step would be to build the part, ship it and bill the designer center. In reality, in engineering interchanges, there is a need for details to be clarified and further negotiated between the designer and the manufacturer.

The protocol for the Transaction Layer has a set of primitives that describe the nature of the job in detail. It includes geometry of the part to be produced, tolerancing, additional engineering information and administrative information regarding product versions, authors, dates, and so on.

TRADEWIND uses two passes for this transaction layer. The first or "coarse" one, where the protocol defines a mutually acceptable region for attributes that are not specific to the manufacturing process used. These include the ranges of cost values, time constraints, dimensions, material types and material volumes. The "coarse" phase represents the kind of interaction that might occur when a designer talks on a telephone to a manufacturer to get a preliminary idea as to whether a job is feasible. The second or "fine" pass is largely process-specific and narrows down cost and time parameter ranges to single "points", thus forming a competitive bid which can be evaluated on the basis of a utility function of cost and time.

THE EXPERIMENTAL TESTBED

We have implemented a testbed that demonstrates the feasibility of complex designer/manufacturer interactions within the infrastructure proposed for Phase I. For this testbed, we chose the Shape Deposition Manufacturing Process at Stanford University due to its 3D capability and availability. Moreover, it offers the potential for fully automating manufacturability analysis and process planning as discussed above.

In our demonstration scenario, in addition to the Registry and Directory Database, there are 5 agent programs running on various machines. There is one designer agent, one broker agent, and 3 manufacturer agents. The designer agent and each manufacturer agent are given via the command-line the pathname of one local file that contains an encoding for a ShopTalk object to be used by that agent. The designer's file contains the encoding of a job posting specification which includes all negotiable parameters of the design and the pathname for a local CAD file. Each manufacturer's file contains an encoding of the range of manufacturing parameters with which it can operate.

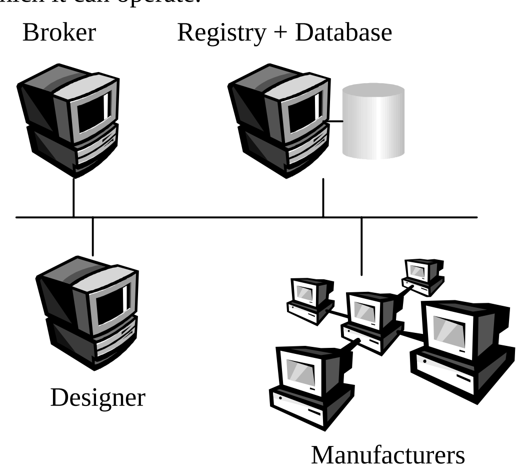

Figure 3: Testbed Schematic

When the designer has read its encoded job posting, it selects a broker (in this case, there is only one to choose from) at random, and it sends that broker information about the job. The broker then forwards that information to each manufacturer. Upon receiving that information, each manufacturer compares the received job information with its own parameters of operation. The manufacturer then returns to the broker a revision containing the intersection of the manufacturer's parameters of operation with the designer's job parameters. If specific parameters do not match between the job posting and the manufacturer's ranges of operation, the information about that mismatch is returned as part of the job revision to the designer via the broker.

In our Phase I implementation, various distributed JAVA objects are passed between a designer agent and multiple manufacturer agents via a broker agent. Currently implemented JAVA objects include JobPosting, JobSpace, CompactSpace, JobSpec and CompactSpec. The JobPosting object represents a request for bids. It contains a JobSpace object representing a set of parameter ranges that the designer views as acceptable. These parameters include utility-related properties such as cost and time, as well as manufacturability-related properties such as dimensions and material types and volumes. A JobSpace object can contain multiple CompactSpace objects, which represent similar parameter ranges for each compact in the job. While JobSpace and CompactSpace represent hyper-volumes in multi-dimensional spaces, JobSpec and CompactSpec objects represent single "points" within those spaces.

The JAVA objects defined above can be represented linguistically for purposes of communication and persistent storage in a language we call ShopTalk. Currently, when one of our agents reads in a ShopTalk object, it identifies what kind of ShopTalk object it is and decides whether or not to use it. Our DesignerAgent implementation specifically looks for a JobPosting clause, whereas the ManufacturerAgent looks for a JobSpace clause specifying its range of possible operation.

In our testbed scenario, our designer agent reads in a ShopTalk file containing a JobPosting clause. It forwards the JobPosting to its broker agent, which forwards the JobPosting object to multiple manufacturer agents. Each of these manufacturing agents has read in a ShopTalk file that contains a JobSpace object representing that manufacturer's range of operational parameters. Each manufacturer agent then compares the JobSpace from the designer agent with its own JobSpace and creates a new JobSpace object representing the mutually acceptable hyper-region (MAR) for the job. The manufacturers wrap these MAR JobSpace objects in a revised version of the original JobPosting object along with updated information about the history of the job posting and, if necessary, textual enumeration of any violations of the manufacturer's operational parameters. They then send these revised JobPosting objects back to the broker agent, which then forwards them to the designer agent.

CURRENT AND FUTURE WORK

Two areas of work lie ahead. On one side, there is the task of upgrading the communication and network structure to make it more reliable and improve its transactional capabilities. Secure communications will be added. On the other is the enhancement of the job representation to capture more accurately the interactions between designers and manufacturers. Figure 4 shows the planned protocol scenario.

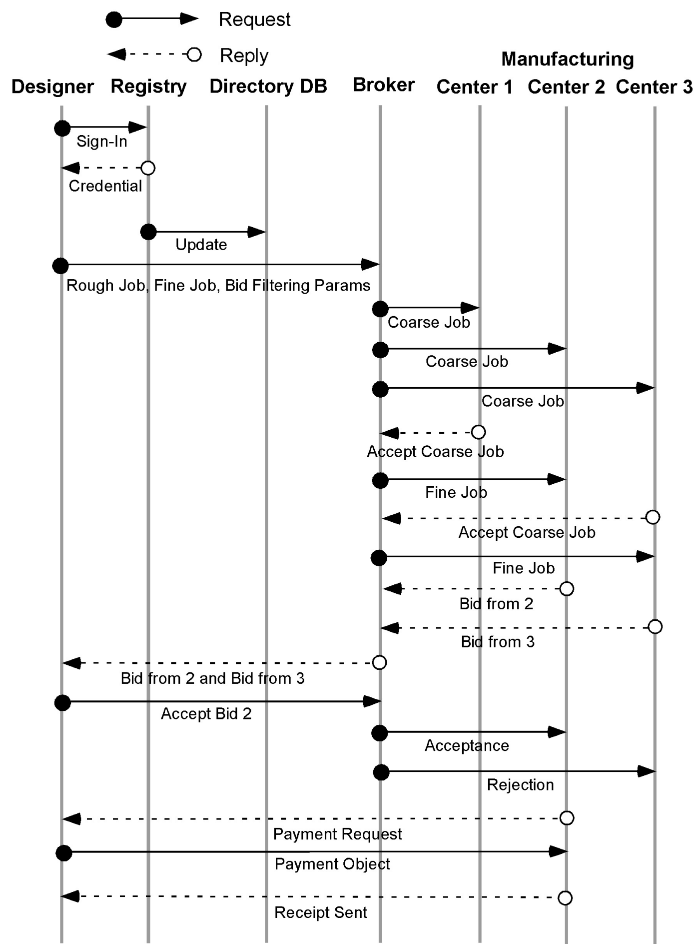

Figure 4: Typical Scenario for Improved Protocol

While the network infrastructure already has its main elements in place, it is lacking some important aspects to make it usable in a real-world scenario. The infrastructure has to be provided with security, enhanced reliability and robust transactional behavior.

Regarding the semantics of the ShopTalk language, it has to be enhanced to capture more fully the concepts handled by Designers and manufacturers, and to express more complex constraints like the ones that relate more than one parameter. Also, the very parameters used in the language have to be validated with manufacturing experience to confirm that they represent accurately the manufacturability issues a part may have.

Finally, the bid resolution process will need to incorporate concepts about preferences of designers and manufacturers within the MAR. These preferences will be expressed as utility functions, which the broker infrastructure will try to maximize.

CONCLUSIONS

The need for an agile marketplace in Rapid Prototyping Services is rising in a fast-growing industry [Wohlers 1998]. Industry is also moving towards more sophisticated designs and parts that are more functional than current look-and-feel prototypes. This requires closer interactions between designers and manufacturers. On the other hand, the specialization of the RP techniques has led to increased outsourcing. Add to these facts the increasing trend towards globalized geographical distribution of manufacturing shops, and it becomes clear that designers and manufacturers would benefit from an agile marketplace and design interchange infrastructure. We have designed and demonstrated a prototype of what such infrastructure could look like. The design is based on independent agents, networked over the Internet, which share a common protocol to establish their presence, resolve bids and close transactions.

An important part of the results has been the ability to resolve bids by using a parametric representation of the parts to be built. This bid resolution is based on satisfying constraints coming from the design space and the manufacturer capabilities into a Mutually Acceptable Region, which represents the range designs that can be effectively manufactured by a particular manufacturer. In case a design is not manufacturable, our formalism is able to identify the violated constraints coming from the manufacturer capabilities, and if can therefore help pinpoint the design parameters that originate the problem.

The current prototype is applied to the SFF class of processes as reflected in the primitives of the protocol. Nevertheless, the basic infrastructure and organization of the system is amenable to be adapted for other manufacturing applications where a complex interaction between the end players (Designers and Manufacturers) is needed.

APPENDIX: THE SHOPTALK LANGUAGE

A language is needed to express the JobPosting clauses that players in the scenario interchange. This language has to convey information about the parameter values and constraints representing the designed part and/or the manufacturing center capabilities. In the current implementation, the expressions are restricted to parameter values and parameter ranges as constraints.

We have designed a language, which we call ShopTalk, to specify various objects relevant to negotiation between designers and manufacturers. These objects convey part parameters and command information that structures the dialog between system participants. Currently, commands are only partially implemented, but a flexible ShopTalk parser has been coded with future expansion of the language in mind.

A ShopTalk clause has the following format:

```
<KEYWORD NAME
[VARNAME=VALUE][VARNAME=VALUE]...>
```

KEYWORD identifies the type of the object, which can be one of the following: JobPosting, JobSpace, JobSpec, CompactSpace, CompactSpec. Future planned implementations include objects such as: Compact Adjacency Graph [Pinilla 97], Posting, Revision, Transaction, Commit, and Acknowledgement. NAME & VARNAME parameters are Strings. A VALUE parameter can be either a number e.g. 3.1415, a range of numbers e.g. (1.0,3.5)(5.7,9.7), or a nested clause.

A hypothetical example file that displays the use of this language in a Job posting from a design center is shown below:

```
<JobPosting Sample Job
[desc=Christmas Rush Job][revisions=0][jobID=1]
[fileName=Attach.stl][dirPath=d:\temp]
[proposed=12/12/1997 at 12:48:55:666 PST]
[revised=12/12/1997 at 12:48:55:666 PST]
[JobSpace=<JobSpace Reindeer Powered Sled
[TotalTime=(10,12)]
[LengthX=(10,10)][LengthY=(10,10)][LengthZ=(10,
10)]
[ArtifactVol=(1000,1000)]
[Cost=(0,200)]
[Copper=(0,1000)][Steel=(0,1000)][Stainless=(0,
0)][Invar=(0,0)]
[RedWax=(0,0)][BlueWax=(0,0)][SolderMask=(0,0)]
[ADD=<CompactSpace Piece1
[Machinability=(3,3)][MinFeatureSize=(0.1,100)]
[Complexity=(5,5)][Removability=(8,8)]
[EstProcTime=(6,7)][MaterialVol=(100,200)][Mate
rialType=(0,1)]>]
[ADD=<CompactSpace Piece2
[Machinability=(5,5)][MinFeatureSize=(0.1,100)]
[Complexity=(8,8)][Removability=(8,8)]
[EstProcTime=(4,6)][MaterialVol=(800,900)][Mate
rialType=(0,1)]>]>]>
```

The object represented in the example above posts a Job called “Sample Job”, which is described as “Christmas Rush Job”. It refers to an STL file called Attach.stl that will contain the part geometry. It then provides various pieces of information about the Job and the posting itself. For the posting it tells creation date and possible subsequent revisions (in this example none). Then, in the nested JobSpace clause, it provides the designer constraints for the job, e.g. dimensions, Cost range, Materials, etc… It also shows the use of an SFF-specific object, the compact. Two compacts are defined in the example, each of them with its associated properties, among them, the material, which is defined by reference to a shared convention within the system. For example, 0 = Copper, 1=Steel. Notice that the example design shown above allows for each compact to be built using either copper or steel.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the NSF under contracts MIP-9617994 and MIP-9618050, as well as the Future Professors of Manufacturing program at Stanford University. In addition, the authors are grateful for the contributions of Prof. Mark Cutkosky, Prof. Fritz Prinz, and others at the Center for Design Research, Rapid Prototyping Lab who were involved in the authoring of the original proposals on which this paper is based.

REFERENCES

R. Merz, F.B. Prinz, K. Ramaswami, M. Terk, L. Weiss, "**Shape Deposition Manufacturing**", Proceedings of the Solid Freeform Fabrication Symposium, University of Texas at Austin, August 8-10, 1994, 1-8.

J.M. Pinilla, Ju-Hsien Kao, Mike Binnard, Fritz B. Prinz, "**The Compact Graph Format"** NIST Rapid Prototyping Workshop, Oct 1997

K. Ramaswami, Y. Yamaguchi and F.B. Prinz, "**Spatial Partitioning of solids for Solid Freeform Fabrication** ", Proceedings of the Fourth ACM/SIGGRAPH Symposium on Solid Modeling and Applications , May,1997.

ITU recommendation M3010 "**Principles for a Telecommunication Management Network**"

S. Rajagopalan et. al. **"Integrated Design and Rapid Manufacturing over the Internet",** Proceedings of DETC98/CIE: 18th Computers in Engineering Conference September 13, 1998 — Atlanta, Georgia

T. Wohlers "**Rapid Prototyping & Tooling State of the Industry: 1998 Worldwide Progress Report**" -
http://lamar.colostate.edu/~wohlers/#State of the Industry