Using the iFAB Architecture to Execute Rapid Response Manufacturing

MBE Summit 2013
Gaithersburg, MD

Presented by:
Mark Traband, PhD
mtt1@arl.psu.edu
iFAB - instant Foundry Adaptive through Bits

iFAB Foundry - a flexible, programmable, distributed production capability able to accommodate a wide range of systems and variants with extremely rapid reconfiguration timescales.

* A network of Information, Material, and Agreements
Designers → Submissions → Design Data Package → Cost, Lead Time → META → Design Completion → Tech Data Package → Foundry Configuration → Bid → DARPA → Status

Automated, Manufacturability Assessment:
- Purchased parts
- Manufactured parts
- Assembly operations
- Shipping cost/time
- Resource requirements
- Build schedule

Distributed Manufacturing Enterprise:
Component Acquisition and Fabrication

Product → Test and Evaluation → As-Built TDP → Final Assembly

Component Model Library

Component Data Package

DARPA

Foundry Configuration

Schedule, Work Instructions, Machine Code

Tech Data Package

Design Completion

META

Designers

Submissions
iFAB Foundry Manufacturability Assessment

VehicleForge Manufacturability Feedback Designer Interface

- aPriori
- CNC-ANA / CNC-RP
- CastView / Cast-ANA

Assembly Planning

Custom Applications
- Wire Harnesses
- Hydraulic Assemblies
- Armor
- Structural Composites

Fixturing Feasibility

Conceptual Manufacturability Analysis

- Build
- Material
- Total
- Cost
- Time
- Parts

Fixturing Feasibility
iFAB Foundry Manufacturability Assessment

**Analysis Workflow**

1. **CAD Design (Creo)**
2. **Design Submission and Analysis (iFAB Execution Site)**
3. **Manufacturing Data Specification (TDP Editor)**
4. **Manufacturing Analysis**
iFAB Foundry Configuration

Technical Data Package

Generate Foundry Configurations

Final Foundry Configuration

Generate Detailed Build Information

Build

Rule-Based Assembly Planning

Human Work Instructions

Automatic CNC Code Generation

Collision-Free Path Planning

Fixturing Analysis

Resource-Constrained Schedule

Detailed Foundry Configuration View

Penn State ARL
Summary/Overview

- Automated process planning and NC code generation within MasterCAM software
- Input: CAD model, material, critical tolerance
- Output: Setup Sheet, NC code

- Analysis / NC code Generation ~4 minutes
- Machining ~15 hours
iFAB Foundry Configuration

NC Code Generation

OVERVIEW

• **Purpose:** Generate CNC code for classes of machined parts using geometric reasoning algorithm for tool and toolpath selection; machine code generation using MasterCAM software through an application programming interface

• **Program development tasks**
  - Volume decomposition, tool selection, setup analysis (PSU/ARL)
  - Fixturing analysis (PARC)
  - Feeds and speeds – milling (MLI)
  - Toolpath and overall code generation (PSU/ARL)

• **MasterCAM functionality**
  - **Subroutines:** divided into drilling, facing, contour and pocket milling operations; each operation has extensive parameter definitions; and tooling library functions for modifying/managing/creating tools
  - **MasterCAM API:** able to generate tool paths using scripted point-by-point specification of tool paths; control of machining linking parameters and tool/operation settings; generation of G-code.
Automated CNC Code Generation w/MasterCAM (PSU)

Setup Analysis

Test case
- Blue represents Post-form
- Yellow represents Removal volume

Preform | Visible volume | Post-form

SETUP0

SETUP1

SETUP2
G-code generation

- **XML file**: includes feature geometry, tool parameters, machine selection and overall sequence of operations

- **C-hook dll**: automated, script-based MasterCAM generation of G-code from XML file input

```
[!--- This file is written for BlockWithThreeHoles.step -->
[!--- Note: The a component of point in drill is the top of the hole -->
[!--- Note: The dimensions are units (for now) - Mastercam defaults are used -->

<operation type="drill">
  <diameter>0.25</diameter>
  <depth>0.5</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="drill">
  <diameter>0.3</diameter>
  <depth>1.25</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="drill">
  <diameter>0.125</diameter>
  <depth>1.25</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="pocket">
  <depth>0.25</depth>
  <point x="0.5" y="0.5" z="0.0" />
  <point x="0.5" y="3.5" z="0.0" />
  <point x="3.5" y="3.5" z="0.0" />
  <point x="3.5" y="0.5" z="0.0" />
</operation>

[!--- This operation is commented out

<operation type="contour">
  <depth>0.01</depth>
  <point x="0.5" y="0.5" z="0.0" />
  <point x="0.5" y="3.5" z="0.0" />
  <point x="3.5" y="3.5" z="0.0" />
  <point x="3.5" y="0.5" z="0.0" />
</operation> -->
```

**XML FILE INPUT**

```xml
[!--- This file is written for BlockWithThreeHoles.step -->
[!--- Note: The a component of point in drill is the top of the hole -->
[!--- Note: The dimensions are units (for now) - Mastercam defaults are used -->

<operation type="drill">
  <diameter>0.25</diameter>
  <depth>0.5</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="drill">
  <diameter>0.3</diameter>
  <depth>1.25</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="drill">
  <diameter>0.125</diameter>
  <depth>1.25</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="pocket">
  <depth>0.25</depth>
  <point x="0.5" y="0.5" z="0.0" />
  <point x="0.5" y="3.5" z="0.0" />
  <point x="3.5" y="3.5" z="0.0" />
  <point x="3.5" y="0.5" z="0.0" />
</operation>

[!--- This operation is commented out

<operation type="contour">
  <depth>0.01</depth>
  <point x="0.5" y="0.5" z="0.0" />
  <point x="0.5" y="3.5" z="0.0" />
  <point x="3.5" y="3.5" z="0.0" />
  <point x="3.5" y="0.5" z="0.0" />
</operation> -->
```

**G CODE OUTPUT**

```c
G00 X0 Y0 Z50
G01 X1 Y0 F0.1
G03 X0 Y0 Z0 I0 J0 K0
G01 X2 Y0 F0.5
G01 X0 Y2 F0.5
G01 X0 Y0 Z0 F0.5
G00 X0 Y0 Z50
```

```c
; XML FILE INPUT
<operation type="drill">
  <diameter>0.25</diameter>
  <depth>0.5</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="drill">
  <diameter>0.3</diameter>
  <depth>1.25</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="drill">
  <diameter>0.125</diameter>
  <depth>1.25</depth>
  <point x="1.0" y="2.0" z="0.0" />
</operation>

<operation type="pocket">
  <depth>0.25</depth>
  <point x="0.5" y="0.5" z="0.0" />
  <point x="0.5" y="3.5" z="0.0" />
  <point x="3.5" y="3.5" z="0.0" />
  <point x="3.5" y="0.5" z="0.0" />
</operation>

[!--- This operation is commented out

<operation type="contour">
  <depth>0.01</depth>
  <point x="0.5" y="0.5" z="0.0" />
  <point x="0.5" y="3.5" z="0.0" />
  <point x="3.5" y="3.5" z="0.0" />
  <point x="3.5" y="0.5" z="0.0" />
</operation> -->
```

Manufactured Part

KiFAB Foundry Configuration
Automated Human Work Instructions

Part Geometry

Operation List
1. Material Stock
2. Laser Cut
3. Painting
4. Symbol
5. Part Inspection

Reference Materials
1. Machine Setup Sheet
2. NC Code

Part List
1. Carbon Steel - A-36 36" x 240" x 0.5"

Required Tools
1. ML 3018-40 CFX
2. Sinks Paint Booth

Personal Protective Equipment

Inspection
Operation 1. Inspect quality of raw stock
Sign: Date: / / 
Operation 2. Check dimensions and tolerances of features
Sign: Date: / / 
Operation 5. Inspect finished part
Sign: Date: / /

Assembly

Part Geometry

Operation List
1. Lift Y000781_060000 for assembly
2. Position Y000781_060000 for assembly
3. Apply consumables as needed
4. Bolt blind holes x 12
5. Bolt through hole

Personal Protective Equipment
RIA Building 299 facilitization requirements are minimized by exploiting the resources of the distributed foundry.

“iFAB Foundry represents a novel kind of manufacturing capability, linked by a common model-based information architecture, that can be distributed geographically and across corporate and contractual boundaries.”

Building 299 iFAB Area = 138,000 ft² Industrial Area
FANG 1 Manufacturing Partners = 3,400,000 ft²
The iFoundry
A Distributed Network of Information, Goods and Agreements

The FFM (FANG#1) Upper Power Pack (Cooling) Example

Assembled at Rock Island Arsenal, Building 299

Demmer Corporation
East Lansing, MI
Cooling Module Bracket

Detroit Flex
Troy, MI
Custom Pipe Assemblies

PIF ARDEC
Picatinny, NJ
Transmission Oil Cooler Tray

Mercury Electronics
Seven Valleys, PA
Hydraulic Harness, Cooling Harness and Can Link Assembly

RG Group
York, PA
Cooling Module

RG Group
York, PA
Hydraulics Assembly

GENCO Infrastructure Solutions
Pittsburgh PA /Rock Island IL
Fasteners, hoses, fittings, clamps reservoirs, etc

PIF ARDEC
Picatinny, NJ
Hydraulic Pump Bracket
Dressed Transmission
Cool Pack
Dressed Engine
FUPP Final Assembly – Engine Transmission Mating
FUPP Final Assembly – Cool pack Installation
FANG Hull
FANG Hull
FANG Hull
The requirement to generate technical data packages sufficient for manufacture is a critical issue:

- Lack of data reduces confidence in cost and lead time estimation
- Detail comes last – suffers from lack of time
- Traditional detailed design data is not machine interpretable
- Design expertise required is high
- Forces designer to down-select early, reducing ability to conduct trades at abstract level

**Level of Effort to Complete Design**


- Concept Design: 20 work-days
- Preliminary Design: 300 work-days
- **Detail Design: 60,000 work-days**
FANG 1 Build
Increasing Complexity of Manufacturability Analyses

FANG 1 Winning Design
15 Apr 2013

• 4/15 – 44 EBOM Line Items

FANG for Manufacturing (FFM)
20 August 2013

• 8/20 – 1704 EBO Line Items
• 7/3 – 1398 EBOM Line Items
• 5/10 – 407 EBOM Line Items
Solution: develop Design For Manufacturability (DFM) Assist Tools - DATs

- Deploy in Design Tool Chain
- Embed in CAD
- Helps discover manufacturability issues up-front
- Constrains user early to manufacturable designs
- Guides user in specifying all needed detail
- Use design automation to bridge gap to detail
- Control data format to ensure machine interpretable
Enabling Manufacturable Designs

Goal: Eliminate the effort spent in detailing designs.

Including DAT tools in the Design environment will ensure that concept designs can be rapidly detailed, and will guarantee that designs are manufacturable.
HuDAT – automatically generates as-cut plate and detailed weld geometry for ballistic joints from concept hull design.

TDP Editor – enables designers to specify detailed part and assembly requirements, saving in a machine interpretable format (XML).

Tolerance Assistant – assigns tolerance callouts to CML component interfaces, guides designers to assign tolerance schemes for hull components.
• **TDP Editing** is design completion process for AVM
  
  • Provides sufficient amount of design data for iFAB manufacturability analysis
  
  • Machine readable format
  
  • Supports manufacturability analysis and HWI generation
  
  • Manufacturing data augmentation being integrated with the design process (and tools)
Design Assist Tools

Hull Design for Manufacturability Assist Tool (HuDAT)

• Guide and constrain the user in order to enable them to create a viable hull design that is manufacturable by the iFAB Foundry
  – Guide: offering valid options for materials, and welds
  – Constrain: rejecting non-iFAB-able designs
  – Let designer spend time innovating
  – Contains: Weld information and Detailed plate and structure information

• Generate a design that may pass qualification of ballistic requirements associated with U.S. military ground vehicles (i.e. ballistic plate joined by ballistic joints)
Hull Design for Manufacturability Assist Tool (HuDAT)

HuDAT CREO Plug-in

Complex Joint Preparation and Welding Rules

Concept Plates → Edge Preparation → Solid Weld Objects

Design Assist Tools
HuDAT - Ballistic Weld Rules for Aluminum

Steel Weld Configurations

Higher plate rests on lower plates

Fillet-Fillet Weld Joint

Joint Geometry Based on Thickness

Joint Geometry Based on Build Sequence

Aluminum Weld Configurations

Criteria – 60 Degree Min Included Angle
Example: Total Flat Area of .56"-.63" For 1.50" Thk Plate

Gravity
Build Sequence

3 Degree Change TO Enable Use of Standard Tooling
Recalced In. 0K Gap

8
7
6
5

4
3
2
1

+Z
Summary

• iFAB Tools and Architecture are being demonstrated on a substantive build application
  – 8 month total build span (includes long lead items – hull, engine, waterjets)
  – 4 month final assembly at Rock Island
  – Excellent data captured for forensics and process improvement – problem reporting, deviations, ECRs
  – Cross-organizational, distributed team

• Primary Contributions:
  – *Automated* cost and lead time feedback to designers
  – *Automatic* assembly sequence, work instruction, and machine code generation

• Path forward
  – Improved capture of manufacturing knowledge in design stage
  – Improved assembly planning tools
  – Incorporation of tolerance allocation for assembly design/analysis