Sight Machine

Common Data Models for Manufacturing
Background and Presentation Overview

- Sight Machine analyzes production by streaming plant floor data from all systems and sources in real time.
- Data is streamed into 4 Common Data Models. These models represent automated production in all industries.
- Streams continuously generate Data Foundation. Data Foundation is then graphed into representations of machines, lines and plants, which are further visualized and analyzed in KPIs, analytics and AI/ML.
- Sight Machine’s architecture is modular, transparent, and configurable at each level. Clients and partners can access and modify: (a) raw data, (b) configuration, and (c) Transformed Data via API and SDK layers.
- This presentation reviews Common Data Models and graphing methods, and highlights a few out of hundreds of analytics currently generated through web services. After several introductory slides, it is mostly pictures.
- Connectivity and pre-processing, and streaming and transformation are addressed elsewhere.
- A summary architecture is provided last.
Things we want to understand in manufacturing

- Machine
- Machine Type
- Facility
- Cycle
- Downtime
- Part
- Batch
- Defect
- Line
- KPI
Problem to be solved: when analyzing plant floor production, customary modeling approaches don’t work

- Most manufacturing activities besides production itself are successfully analyzed with batch processing and traditional modeling techniques: Product design and engineering (PLM), business processes (ERP), physics-based simulation, Supply Chain (SCM)

- Analysis of actual plant floor production is different. Despite substantial investment in the last decade by both technology and industrial companies, traditional modeling techniques have not worked for plant floors

- Among the many challenges, one of the more fundamental is around representational units

- Because machines are often the representational unit we think of first, when analyzing production we usually model machines. Other representational units we might use include single sensors, larger data sources (e.g., historians, quality systems) lines, plants, and parts

- But manufacturing has many thousands of kinds of machines and hundreds of systems, and relationships among data sources are complex. This complexity is one reason traditional approaches fail

- For plant floors, we need a simpler approach with only a few elemental units that can be continuously generated and “built up” to represent all machines, lines, and plants
What’s needed are common building blocks

- We need to use building blocks that are common across every machine and process

- We can then assemble those building blocks into higher-level conceptual templates. This way, we can relate every machine, process, and part with the same underlying, standardized elements

- These building blocks are Common Data Models for Production. With these models, mapped through stream processing, we can represent almost every activity in manufacturing with the same Data Foundation

- This approach is essential for scale. It enables analysis not just from the asset to enterprise level, but also across industries and Value Chains
Manufacturing Common Data Models

To represent infinite machines, parts, processes, and plants, only a few Common Data Models are needed

- Building blocks are represented as rows in data tables
- Columnar values are parameters of interest, generated through real-time data transformations
- Columnar data combines and transforms data from any source:
  - Machines (hundreds of sensors each, thousands of different machines per enterprise)
  - Operator data (Excel, entries, digitized reports)
  - Adjacent systems (historians, ERP, MES, quality, etc.)
  - Environment (temperature, humidity)
  - Value chain (raw material, upstream supplier)
- The resulting information is now **Data Foundation**. It is continuously generated and standardized, and can be associated, combined, compared, and analyzed with all other standardized units of information
- Data Foundation enables broad application of analytical techniques from visualization and KPIs to data science and AI/ML. Data Foundation can also be joined with other information (finance, energy, logistics) in manufacturing enterprises
- **Data Foundation supports both real-time operational analysis and firm, or industry-level analytics**
Common Data Models
Cycle characterizes discrete periods of machine activity. A cycle represents a unit of work by a machine.

Each row in a Sight Machine data table describes a single recurring unit of work by any machine. Examples of repetitive work done by machines are limitless and include pressing, cutting, filling, painting, and heating.

Cycle creates a common thread across machines, lines, process areas, facilities, and industries.
**Unit of work**
- Primarily identified by Machine + Time (sometimes Machine + Counter/Unique Identifier)
- Contains a shift, production day, and output

**Commonly includes:**
- Machine telemetry
- Running status
- Product/material information
- Energy use
- Quality/inspection data
Example Cycle Data Table

<table>
<thead>
<tr>
<th>#</th>
<th>Machine</th>
<th>Cycle Start Time</th>
<th>Cycle End Time</th>
<th>Production Day</th>
<th>Cycle Time (Net) Seconds</th>
<th>Cycle Time (Gross) Seconds</th>
<th>Shift</th>
<th>Output</th>
<th>Aluminum Staging - Total Time</th>
<th>Aluminum Temperature °C</th>
<th>Cylinders</th>
<th>Die Temperature °C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:39:55</td>
<td>2021-03-06 09:40:27</td>
<td>2021-03-06</td>
<td>32</td>
<td>32</td>
<td>Shift 2</td>
<td>1</td>
<td>5000</td>
<td>650.9</td>
<td>4</td>
<td>283.7</td>
</tr>
<tr>
<td>2</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:39:12</td>
<td>2021-03-06 09:40:17</td>
<td>2021-03-06</td>
<td>65</td>
<td>65</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>653.1</td>
<td>4</td>
<td>288.7</td>
</tr>
<tr>
<td>3</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:38:20</td>
<td>2021-03-06 09:39:34</td>
<td>2021-03-06</td>
<td>65</td>
<td>65</td>
<td>Shift 2</td>
<td>1</td>
<td>5000</td>
<td>658.0</td>
<td>4</td>
<td>284.8</td>
</tr>
<tr>
<td>4</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:37:59</td>
<td>2021-03-06 09:38:59</td>
<td>2021-03-06</td>
<td>30</td>
<td>30</td>
<td>Shift 2</td>
<td>1</td>
<td>5000</td>
<td>657.7</td>
<td>4</td>
<td>289.0</td>
</tr>
<tr>
<td>5</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:37:25</td>
<td>2021-03-06 09:38:25</td>
<td>2021-03-06</td>
<td>34</td>
<td>34</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>656.0</td>
<td>4</td>
<td>291.3</td>
</tr>
<tr>
<td>6</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:36:52</td>
<td>2021-03-06 09:37:52</td>
<td>2021-03-06</td>
<td>33</td>
<td>33</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>658.5</td>
<td>6</td>
<td>288.0</td>
</tr>
<tr>
<td>7</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:36:21</td>
<td>2021-03-06 09:37:21</td>
<td>2021-03-06</td>
<td>31</td>
<td>31</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>659.4</td>
<td>6</td>
<td>285.8</td>
</tr>
<tr>
<td>8</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:35:56</td>
<td>2021-03-06 09:36:56</td>
<td>2021-03-06</td>
<td>25</td>
<td>25</td>
<td>Shift 2</td>
<td>1</td>
<td>5000</td>
<td>657.1</td>
<td>6</td>
<td>303.5</td>
</tr>
<tr>
<td>9</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:35:12</td>
<td>2021-03-06 09:36:12</td>
<td>2021-03-06</td>
<td>65</td>
<td>65</td>
<td>Shift 2</td>
<td>1</td>
<td>5000</td>
<td>669.6</td>
<td>6</td>
<td>294.7</td>
</tr>
<tr>
<td>10</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:34:39</td>
<td>2021-03-06 09:35:39</td>
<td>2021-03-06</td>
<td>33</td>
<td>33</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>663.6</td>
<td>4</td>
<td>289.6</td>
</tr>
<tr>
<td>11</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:34:05</td>
<td>2021-03-06 09:35:05</td>
<td>2021-03-06</td>
<td>65</td>
<td>65</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>654.3</td>
<td>4</td>
<td>290.1</td>
</tr>
<tr>
<td>12</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:33:24</td>
<td>2021-03-06 09:34:24</td>
<td>2021-03-06</td>
<td>31</td>
<td>31</td>
<td>Shift 2</td>
<td>1</td>
<td>5000</td>
<td>655.5</td>
<td>4</td>
<td>283.9</td>
</tr>
<tr>
<td>13</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:32:48</td>
<td>2021-03-06 09:33:48</td>
<td>2021-03-06</td>
<td>31</td>
<td>31</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>654.6</td>
<td>4</td>
<td>284.6</td>
</tr>
<tr>
<td>14</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:32:09</td>
<td>2021-03-06 09:33:09</td>
<td>2021-03-06</td>
<td>34</td>
<td>34</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>654.2</td>
<td>4</td>
<td>288.2</td>
</tr>
<tr>
<td>15</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:31:39</td>
<td>2021-03-06 09:32:39</td>
<td>2021-03-06</td>
<td>65</td>
<td>65</td>
<td>Shift 2</td>
<td>1</td>
<td>5000</td>
<td>659.1</td>
<td>4</td>
<td>292.2</td>
</tr>
<tr>
<td>16</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:30:50</td>
<td>2021-03-06 09:31:50</td>
<td>2021-03-06</td>
<td>23</td>
<td>23</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>655.0</td>
<td>4</td>
<td>291.8</td>
</tr>
<tr>
<td>17</td>
<td>Busan - Diecast 1</td>
<td>2021-03-06 09:29:50</td>
<td>2021-03-06 09:30:50</td>
<td>2021-03-06</td>
<td>26</td>
<td>26</td>
<td>Shift 2</td>
<td>1</td>
<td>6000</td>
<td>659.8</td>
<td>4</td>
<td>288.6</td>
</tr>
</tbody>
</table>
Downtime describes instances and durations of non-productive, idle, or stop time for a machine.
Downtime Data Model

**Track when a machine is down**
- Start and end of downtime
- Reason code for downtime

**Notes**
- Split on shift, day boundaries, and reason code changes
- Commonly contains type, category, and reason code

Other reason code may be on Cycles

Primary reason code on Downtime
**Part and Batch**

**Part** provides traceability across production processes, facilities, and supply chains, and associates the process data with quality outcomes.

**Batch** represent raw material, output grouping, and summary data, all of which are associated with multiple parts.
**Part and Batch Data Model**

**Represents a Part/Batch as it moves down the line**
- Used for product traceability, identified by serial, or sometimes synthetic timestamp “serial”
- Will have multiple machines’ data

**Contains**
- Cycles for all machines a part will/could touch
- Start time is from the first machine, end time is the most recent machine — across many facilities as needed
Defect represents non-conformant production output, in both single parts and batches.
Defect Data Model

Track defect-specific information, associated with parts or batches
- Defect test time
- Category, type, and quantity
Configuring Common Data Models Into Plants, Lines, Machines, KPIs
Facility

Facility defines the location, time zone, and shift schedule unique to each facility to determine when machines should be running, overall facility-wide KPIs, and shift performance analysis.
Example Facility

- Facility location (TZ-aware)
- Shift schedule
- Machine assignment

```json
Configuration
14  "factory_id": "Discrete_PZ_FZ",
15  "factory_provider": "Discrete_PZ",
16  "factory_location": "Discrete_FZ",
17  "factory_location_clean": "Ann Arbor Facility",
18  "geo_location": {
19    "lat": 3.799231,
20    "lng": -122.4819282,
21  },
22  "place_name": "221 N Main St, Ann Arbor, MI 48104, USA",
23  "shift_events": [
24    {
25      "start_time": "2000-01-01 07:00:00.000000",
26      "end_time": "2000-01-01 19:00:00.000000",
27      "freq": "WEEKLY",
28      "byday": [
29        "SU",
30        "MO",
31        "TU",
32        "WE",
33        "TH",
34        "FR",
35        "SA"
36      ],
37    },
38    {
39      "exdates": [],
40      "shiftid": "shift_1558055951922",
41    }
42  ]
```
Machine and Machine Type

**Machine** is an instantiation of a machine type mapped to a facility location and shift schedule.

**Machine Type** represents several of the exact same machine that can be modeled with the same schema (same sensors, etc.).
Example Machine Type

Machine Type

- Data schema and tag metadata:
  - Internal name
  - UI naming
  - Function used to create
  - Output format
Example: Linking Machines to Facilities and shifts

Link individual asset on the plant floor to the Machine Type
- Assign Machine to Facility and shift schedule
**Line** defines the layout and sequence of a series of machines involved in the production process and allows for functionality like bottleneck detection, overall process OEE, traceability, and cross-asset analysis.
Example Line

Configured line topology

Time-based offsets to analyze machine interactions
KPI represents key performance indicators unique to each facility's process and goals. KPIs are formulas calculated dynamically at runtime.
Example KPI

Formula configuration

- Dynamic run-time calculation

```json
{
  "obj": {
    "dependencies": [
      {
        "aggregate": "sum",
        "name": "gross_cycle_time",
      },
      {
        "aggregate": "sum",
        "name": "net_cycle_time",
      }
    ],
    "display_name": "Availability",
    "formula": "{(net_cycle_time)/gross_cycle_time)*100 if gross_cycle_time>0 else None}"
  },
  "id": "f86e11dab1a08112444ce2e",
  "model": "line",
  "name": "availability"
},
"type": "KPI"
}
```
Real-time Applications
To drive outcomes, manufacturers require real-time insight

For real-time analysis, stream-processing data into Common Data Models is the ideal approach

- Sight Machine’s **Pipeline as a Service** is a configurable stream-processing product that continuously transforms operational data into Data Foundation. The screen at the right is one of our browser-based tools for configuring pipelines.

- Factories frequently add and change data. In this environment, robust pipeline management tools are essential. Sight Machine has management tools for modifying data sources, systematically changing configurations across pipelines, dynamically provisioning cloud services, and generating new analytics.

- With a standardized Data Foundation, a wide array of analytics can be applied from visualization and KPIs to data science and AI/ML.

- Data Foundation can be supplied to other systems (through API and SDK layers) such as Supply Chain Control Towers, CMMS, S&OP, etc.
Visibility and KPIs

Data Foundation supports continuous visibility into operations. Data can be visualized and analyzed through Sight Machine and other leading applications such as Power BI, Tableau, Looker.

An example of a series of visualizations and analytics used to run a plant.
**Example: Line Productivity**  This “bottleneck detector” shows by SKU where problems arise in multiple lines. Takt time for each step is updated and shown in a heatmap of blocked and starved steps. The analytic uses machine and line models to continuously optimize production. The client achieved a 15% efficiency lift at one of its best plants.
Dynamic Recipes and Operator Co-Pilot: Continuous guidance for optimization

- **Dynamic Recipes** are another example of advanced analysis. This analytic sets optimal settings for complex arrays of assets by analyzing all previous production runs and optimizing based on goals set by the process engineer. The **Operator Co-Pilot** advises the engineer of any variations from ideal settings.

- **Example: Energy and Emissions** The screens below show a real-time Dynamic Recipe for minimizing SO₂ emissions in a chemicals process at a given temperature, and a Co-Pilot advising of variations that need to be corrected.
Example: Dynamic Optimization of Assets

Milk supplies and product mix are ever-changing in dairies. How best to use critical assets, like pasteurizers? This model uses genetic algorithms to schedule optimal asset use. The model adjusts every 30 minutes to reflect dynamic changes in supply and demand.

Sight Machine has applied many AI/ML techniques to Data Foundation

Manufacturing Data Foundation provides abundant opportunities for AI/ML models
Representative AI/ML use cases are listed here
Architecture
Sight Machine Architecture

Each layer is open and modular: Transform directly supports 3rd party products and services

**Connect**
- Securely ingest all plant data

**Transform**
- Automate data transformation
  - Pipeline as a Service
  - Common Data Models
  - Stateful Transforms

**Analyze**
- Analyze transformed data with 1st or 3rd party apps

**Build**
- Share transformed data with other applications

**Results**
- Make data useful for all stakeholders

- Operations Leadership
- Process Engineers
- Operators
- IT Managers
- Data Scientists
- Supply Chain
- Sustainability
- Connected Machines
- Sales Teams
- Planning

Controls/Automation
MES/ERP
Images
Serialization
Historians
Quality
Edge Analytics

Manufacturing-specific data transformations (~50 per stream)

OT Data is now transformed and IT-ready

WMS
SCM
MRP
Production Planning

S&OP
ML Ops
CMMS
CRM

Pipeline as a Service
Common Data Models
Stateful Transforms

WMS
SCM
MRP
Production Planning

S&OP
ML Ops
CMMS
CRM

Controls/Automation
MES/ERP
Images
Serialization
Historians
Quality
Edge Analytics
Thank you