MTConnect Architecture and Research Overview

NAMRC
UNC Charlotte
June 9, 2015

VIMANA | by System Insights
Berkeley, CA | Chennai, India

William Sobel
Chief Strategy Officer
System Insights

MTConnect Chief Architect

Me

- Chief Strategy Officer – System Insights
- Chief Architect of the MTConnect Standard for the last 8 years
- Chair of the MTConnect Technical Steering Committee
- Original Author of the Standard

- My Background
  - Financial Systems Architecture
  - Real-Time Analytics
  - Distributed Architecture & Fault Tolerance

Warning

- Lots of material – too much for time allotted
  - I will be moving quickly
  - This is being recorded and will be posted online

- As well… had to cut back material…
  - There will be followup webinars to for in-depth review
  - Data analytics tools will be covered in followup online training

MTConnect Overview

What is MTConnect?

- A domain model for manufacturing equipment
- Free, **NO** IP hindrances – You won't get sued for using it
- Read-only – Safe
- Shoulders of giants – use most prevalent standards as foundation (HTTP/XML)
- Never create where we can borrow /collaborate
- Most requests and verification can be done using a browser

What MTConnect is NOT

- 100% Complete – Did not boil the ocean
- Command and control
- Optimized
- Proprietary
- Complex
- Analytical
- An application

Taxonomy of Standards

- Procedures and Best Practices
  - ISO 9001
- Protocols, Compression, and Encryption
  - Syntax
  - Transport
  - HTTP, SSL, OpenDDS, …
- Information Models
  - Semantics
  - Vocabulary
  - ISA-95, STEP, …

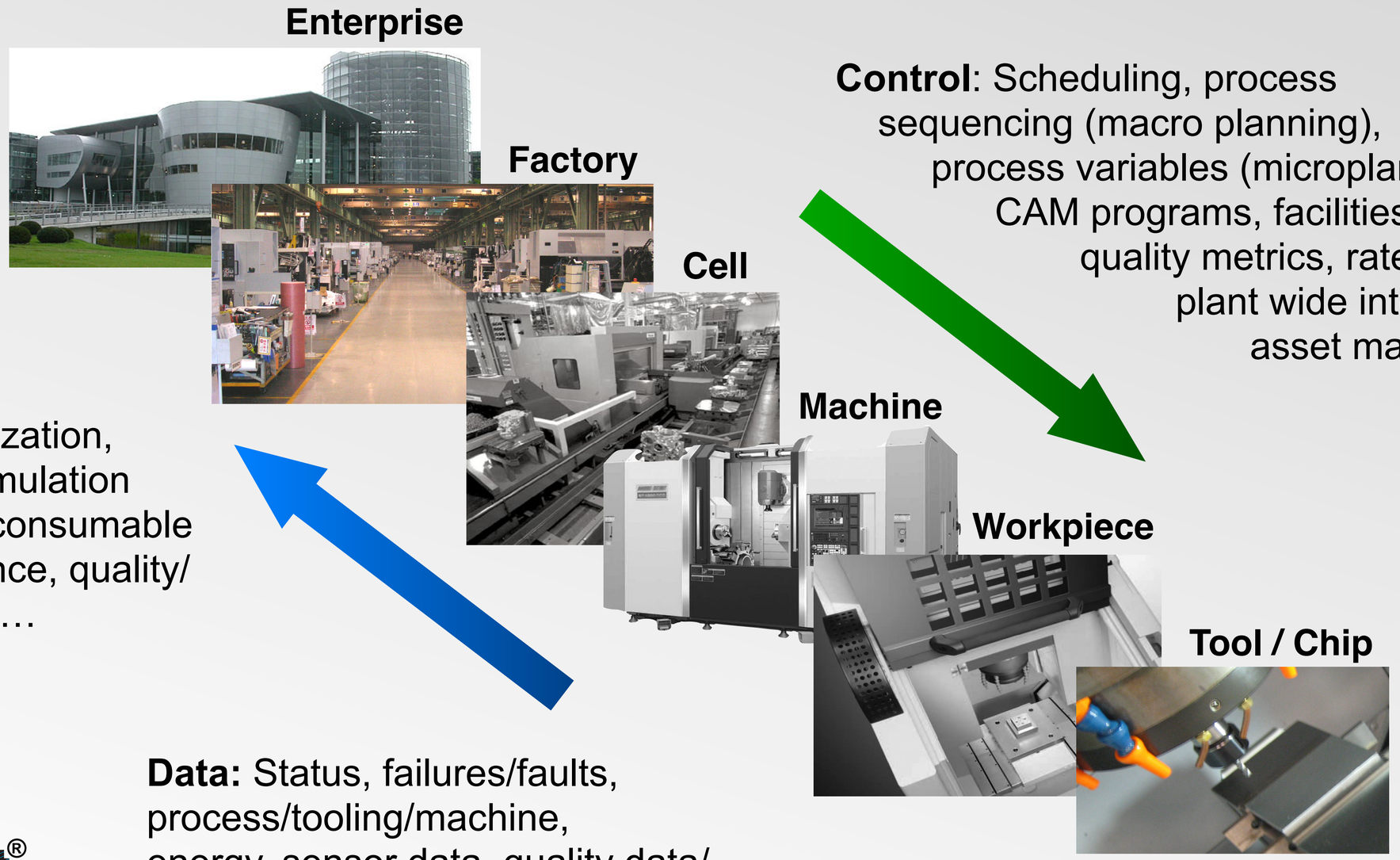
MTConnect specifies: Protocol and Semantics

MTConnect defines the communications protocols

and

the vocabulary and semantics of the information models

What Standards Enables

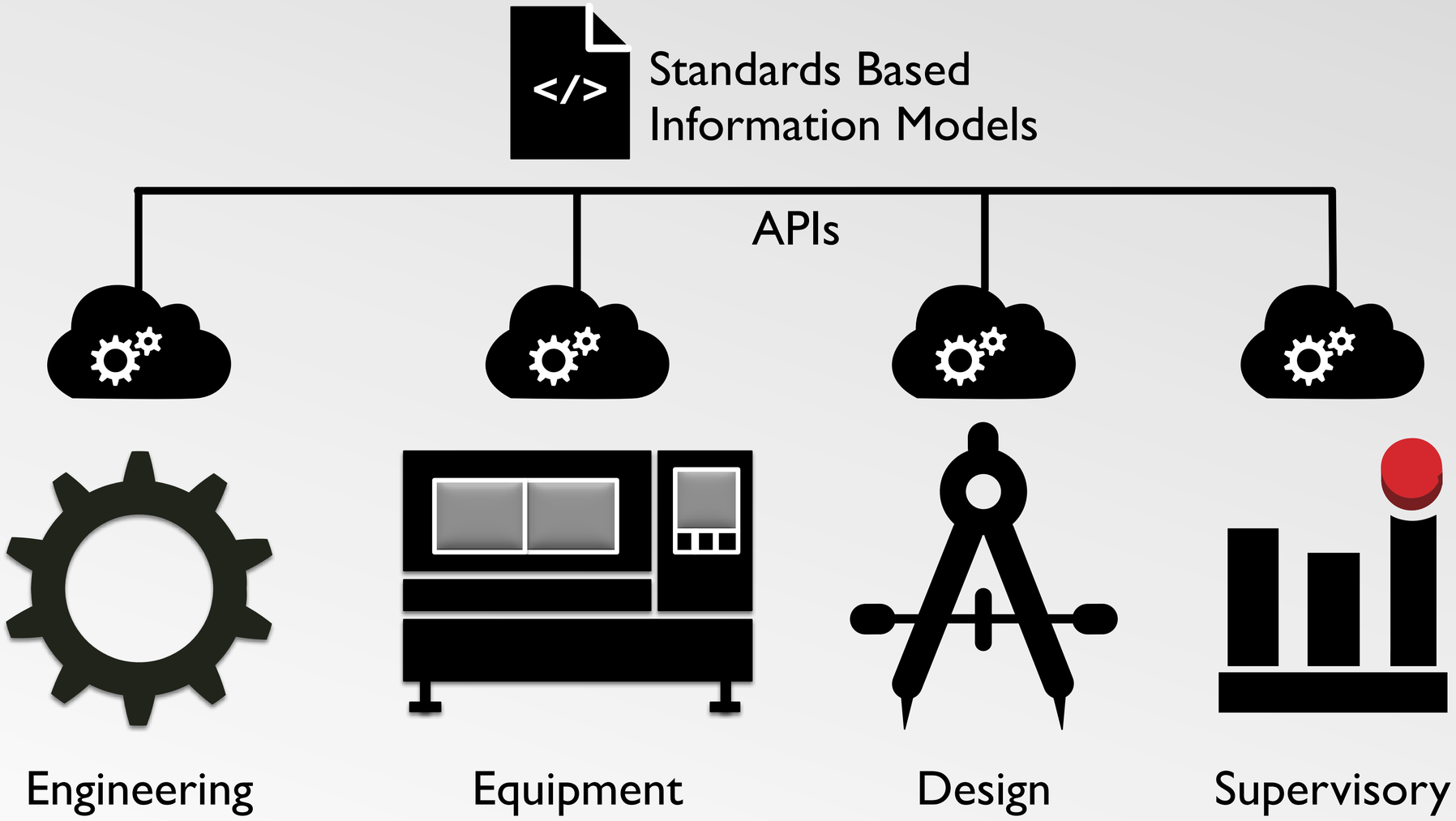


Control: Scheduling, process sequencing (macro planning), process variables (microplanning), CAM programs, facilities, tooling, quality metrics, rates, cost, plant wide integration, asset management…

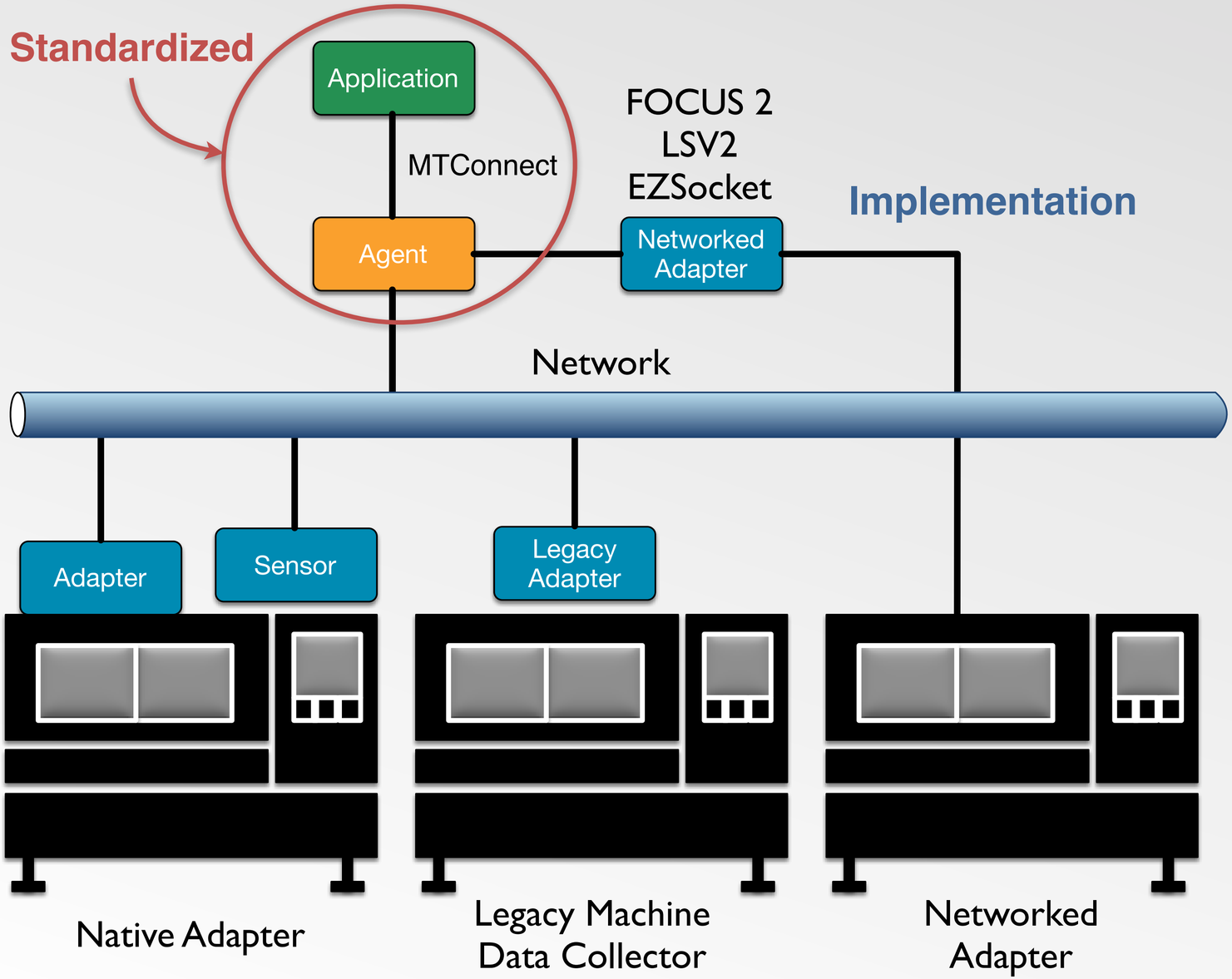
Information: Utilization, OEE, verifying simulation with REAL data, consumable needs, maintenance, quality/ inspection results…

Data: Status, failures/faults, process/tooling/machine, energy, sensor data, quality data/ inspection output…

Services

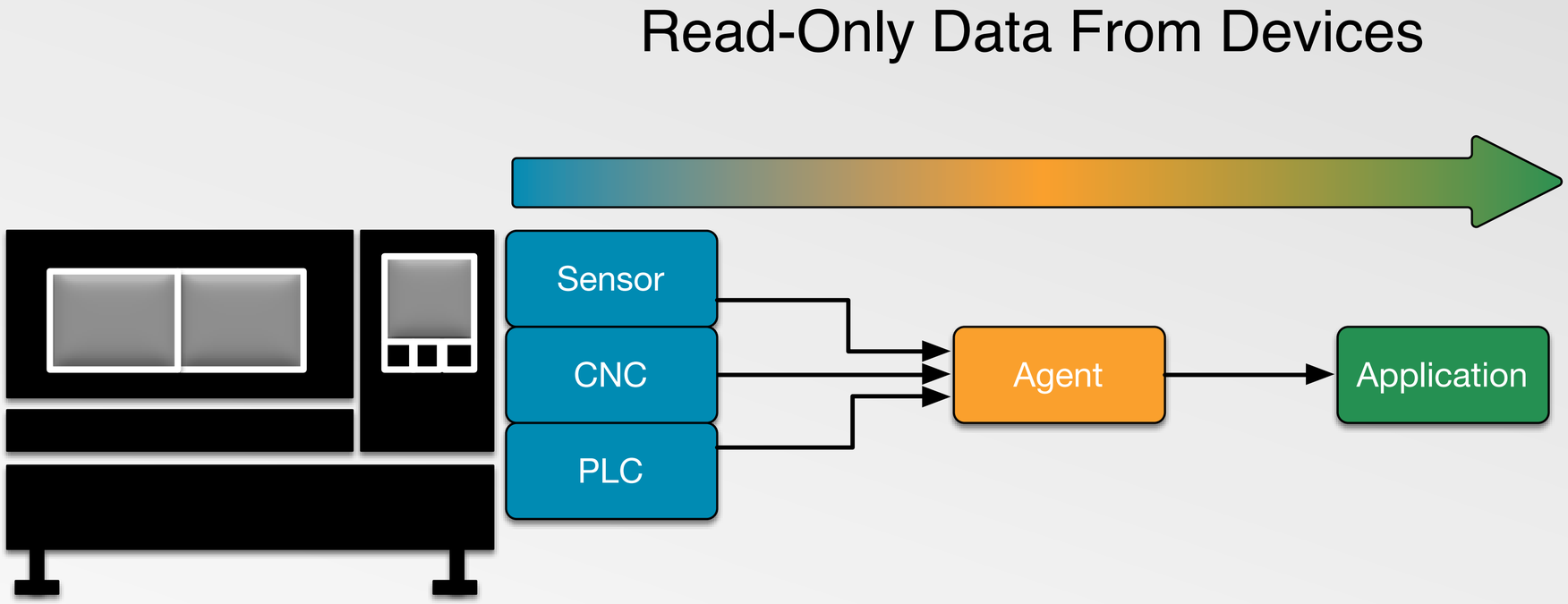


MTConnect Architecture



- An agent can support multiple machines
- Adapters can be provided by the manufacturer of the machine or developed by third parties
- Adapter can reside:
  - On the machine's controller
  - Connected to the PLC wires
  - Remotely if the machine communicates using a networked protocol
- Applications only communicate with the MTConnect Agent

Data Flow



The Role of the Adapter and Agent

Meaning

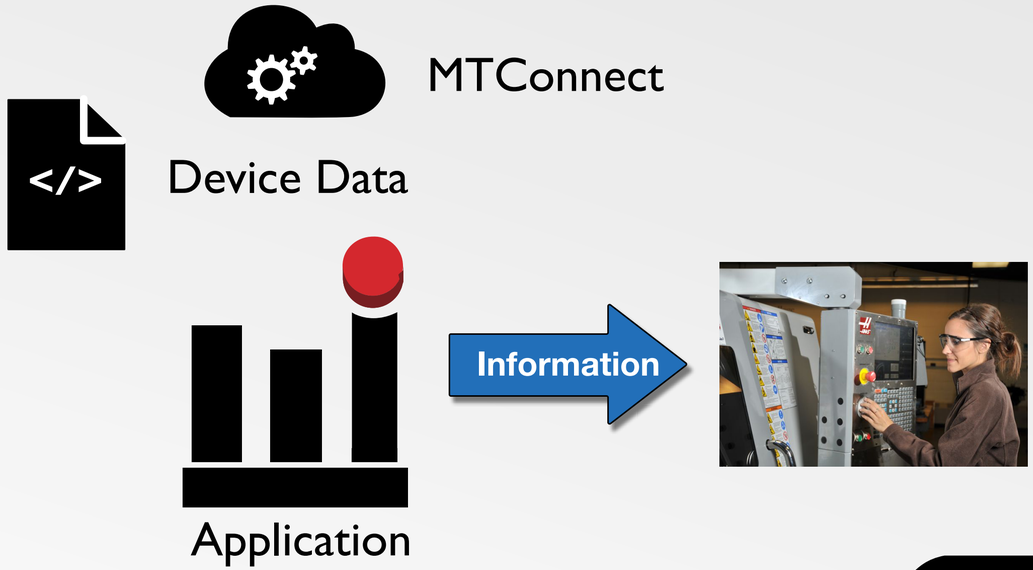
Raw | Syntax | Semantics

010100100110 — Serial — a = 5000, b = 12.43, c = 87.22, d = 2 — Network

Device
- Linear X
  - Position: 196.54mm
  - Load: 12.43%
- Rotary C
  - Rotary Velocity: 87.22 RPM
- Controller
  - Execution: ACTIVE

Value

Semantic Data Transformation Pipeline



Information Models

Four Top Level MTConnect Document Types

- MTConnectDevices – Meta data about the device
- MTConnectStreams – The time-series values
- MTConnectAssets – Key/Value store of information
- MTConnectError – Protocol related errors

Prerequisites – Just enough XML

- XML – Extensible Markup Language
- Provides open standards based grammar for data representations
- Human Readable → Verbose! → Ease of Use
- E**X**tensible – types can be extended using namespaces
- Verifiable
  - Two Levels: *Well Formed* and *Valid*
- *XML Schema* – Used to *valid*ate the grammar and vocabulary
  - MTConnect uses XML Schema to validate the documents
- *XPath – Used to select individual components or data items*

XML Example

Process Instruction

Open Element

CDATA

Element w/ No Content

```
<?xml version="1.0" encoding="UTF-8"?>
<Element attribute="value">
    <SubElement baz="VALUE">
        CDATA
    </SubElement>
    <!-- Comment -->
    <Author id="x213455">
        <FirstName>Lewis</FirstName>
        <LastName>Carroll</LastName>
        <Address>
            <Street>Weyside Rd</Street>
            <City>Guilford, Surrey</City>
            <Country>United Kingdom</Country>
            <PostalCode country="UK">GU1 1HZ</PostalCode>
        </Address>
        <!-- Document -->
        <MixedContent>
            Some text
            <a href="http://example.com">A value</a>
            Some more text
        </MixedContent>
        <SimpleElement attribute="value"/>
    </Author>
</Element>
```

Documents MUST have ONE top level element

Attributes

Attribute "id" is guaranteed to be unique within the XML document

Close Element

Element Order Matters
Whitespace Does NOT Matter
Attribute Order Does NOT Matter

MTConnect Style Guide

- No abbreviations *(except id)*
- Elements
  - CamelCase - first letter of Each Word is capitalized and spaces removed
- Attributes
  - name - snakeCase - first Letter of second+ words are capitalized
  - value – for controlled vocabularies, UPPER_CASE with _ separating words
- Container elements are pluralized and have no attributes
  - Used for grouping (Devices, Components, DataItems, Events, …)

Some Basic XPath – W3C Standard

```
<?xml version="1.0" encoding="UTF-8"?>
<Element attribute="value">
    <SubElement attribute="value">
        CDATA
    </SubElement>
    <!-- Comment -->
    <Author id="x213455" living="NO" birthday="1832-01-27"  famous="YES">
        <FirstName>Lewis</FirstName>
        <LastName>Carroll</LastName>
        <Address>
            <Street>Weyside Rd</Street>
            <City>Guilford, Surrey</City>
            <Country>United Kingdom</Country>
            <PostalCode country="UK">GU1 1HZ</PostalCode>
        </Address>
        <!-- Document -->
        <MixedContent>
            Some text
            <a href="http://example.com">A value</a>
            Some more text
        </MixedContent>
    </Author>
</Element>
```

All Authors
//Author

Find all famous dead authors
//Author[@living="NO" and @famous="YES"]

All Elements with attribute country "UK":
//*[@country="UK"]

Explicit Path
/Element/MixedContent

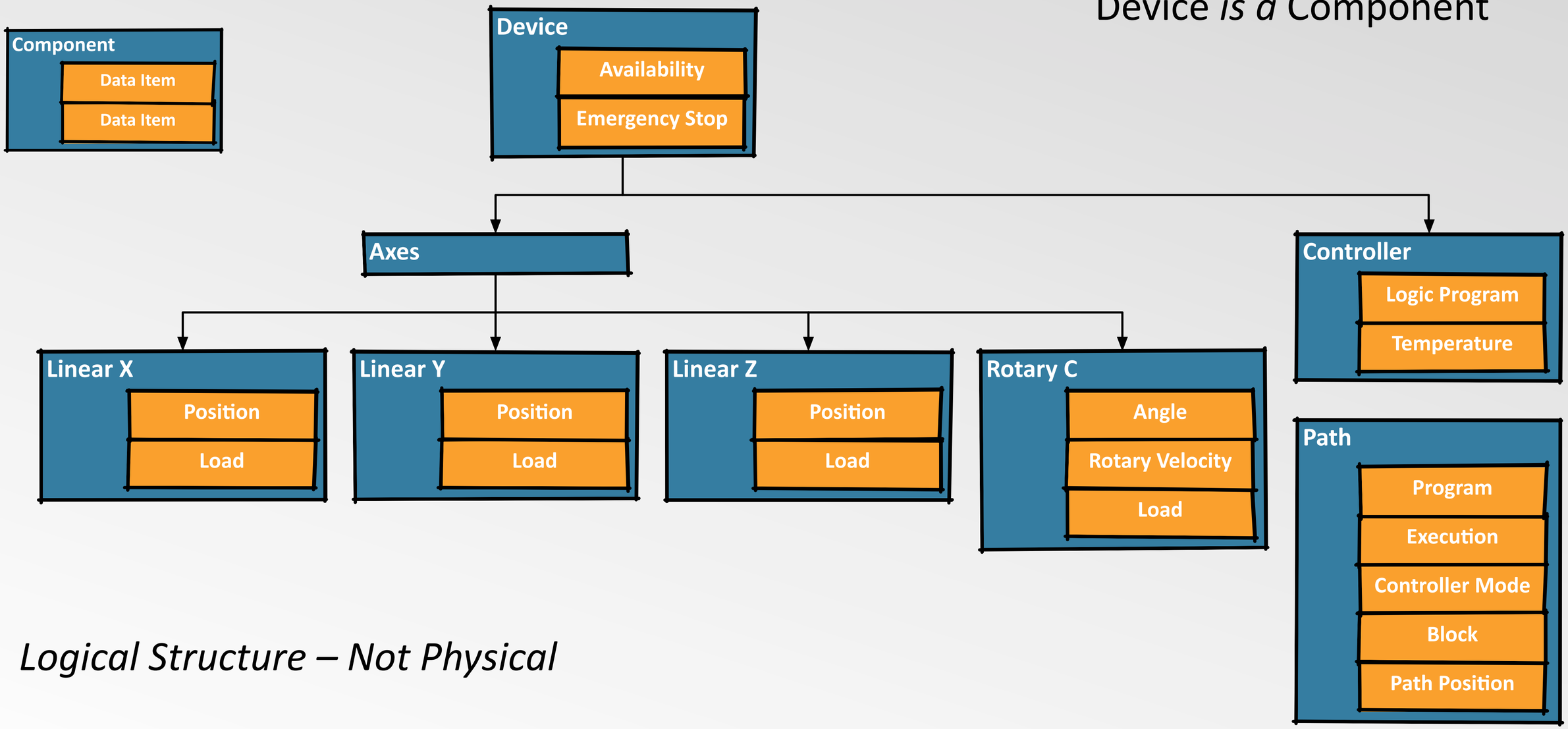
MTConnect uses XPath to select specific components and data items with current and sample request

MTConnectDevices (probe request)

- Provides logical device structure
- Breaks the device down into Components and Data Items, the data the components are capable of providing
- Components are typed by the element name – consistent across all manufactures
  - Beyond simple data typing (int, float, string) – syntax
- All data item are semantically and contextually identified – a *sample* of type *temperature* will always be reported in centigrade, positions in millimeters, etc…
- Components can be nested to provide additional context
- Only the **Availability** event is required
- **Only Metadata**
  - **Describes the data – structure and meaning – think database schema**
  - **No Values**

Devices, Components and Data Items

Device *is a* Component



Logical Structure – Not Physical

Components

- Attributes
  - id *
  - name
  - uuid
- Element name is component type:
  - Controller
  - Path
  - Rotary
  - Linear
  - …

* required

```
<?xml version='1.0' encoding='UTF-8'?>
<MTConnectDevices xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
        xmlns='urn:mtconnect.org:MTConnectDevices:1.3'
        xmlns:m='urn:mtconnect.org:MTConnectDevices:1.3'
       xsi:schemaLocation='urn:mtconnect.org:MTConnectDevices:1.3 /schemas/
MTConnectDevices_1.3.xsd'>
 <Header creationTime='2015-02-10T10:04:55Z' assetBufferSize='1024' sender='localhost'
           assetCount='0' version='1.3' instanceId='0' bufferSize='524288'/>
 <Devices>
  <Device name='INDEX-200' uuid='ee44f744-0a4a-11e5-85ff-28cfe91a82ef' id='INDEX-200'>
   <Description model='C200' manufacturer='Index'>Index C200 - Machine #200</Description>
   <DataItems>
    <DataItem type='AVAILABILITY' category='EVENT' id='dtop_5001' name='avail'/>
    <DataItem type='EMERGENCY_STOP' category='EVENT' id='dtop_5002' name='estop'/>
    <DataItem type='SYSTEM' category='CONDITION' id='dtop_5003' name='alarm'/>
   </DataItems>
   <Components>
    <Axes name='axes' id='axes_5004'>
     <Rotary name='C4' nativeName='S4' id='C4_5007'>
      <DataItems>
       <DataItem type='x:MOTION' category='EVENT' id='C4_5008' name='s4_motion'
                 subType='ACTUAL'/>
       <DataItem category="SAMPLE" id="c2" name="Sspeed" nativeUnits="REVOLUTION/
MINUTE" subType="ACTUAL" type="SPINDLE_SPEED" units="REVOLUTION/MINUTE"/>
      </DataItems>
     </Rotary>
```

Top level element

- Most XML documents will have the following attributes:

- xmlns:<name> – Declares a namespace to allow for combination and extensibility
- xmlns – Default namespace
- xsi:schemaLocation – The url or relative location of the XML Schema

- All MTConncect XML will have these attributes at the top level

```
<MTConnectStreams
  xmlns:m="urn:mtconnect.org:MTConnectStreams:1.3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mtconnect.org:MTConnectStreams:1.3"
  xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.3
  ../../MTConnectStreams_1.3.xsd">
```

Data Item

- Attributes
  - id *
  - category *
  - type *
  - subType
  - name
  - units †
  - nativeUnits
  - nativeScale
  - coordinateSystem
  - representation
  - statistic
  - significantDigits
  - sampleRate
- Sub-Elements
  - Source
    - componentId
    - dataItemId
  - Constraints
    - Minimum
    - Maximum
    - Filter

* required

† required for samples

```
<DataItems>
  <DataItem type='AVAILABILITY' category='EVENT'
            id='dtop_5001' name='avail'/>
  <DataItem type='EMERGENCY_STOP' category='EVENT'
            id='dtop_5002' name='estop'/>
  <DataItem type='SYSTEM' category='CONDITION'
            id='dtop_5003' name='alarm'/>
 </DataItems>
 <Components>
  <Axes name='axes' id='axes_5004'>
   <Components>
    <Rotary name='C4' nativeName='S4' id='C4_5007'>
     <DataItems>
      <DataItem type='x:MOTION' category='EVENT'
                id='C4_5008' name='s4_motion'
                subType='ACTUAL'/>
       <DataItem category="SAMPLE" id="c2"
name="Sspeed" nativeUnits="REVOLUTION/MINUTE"
subType="ACTUAL" type="SPINDLE_SPEED"
units="REVOLUTION/MINUTE"/>
     </DataItems>
    </Rotary>
```

*Data Items **SHOULD** be unique by:*
***Category** – **Type** – **subType** within a Component*

Data Item Categories

- Data Item Types are broken down into three Categories:
  - *Event*
  - *Sample*
  - *Condition*

Data Items - Events

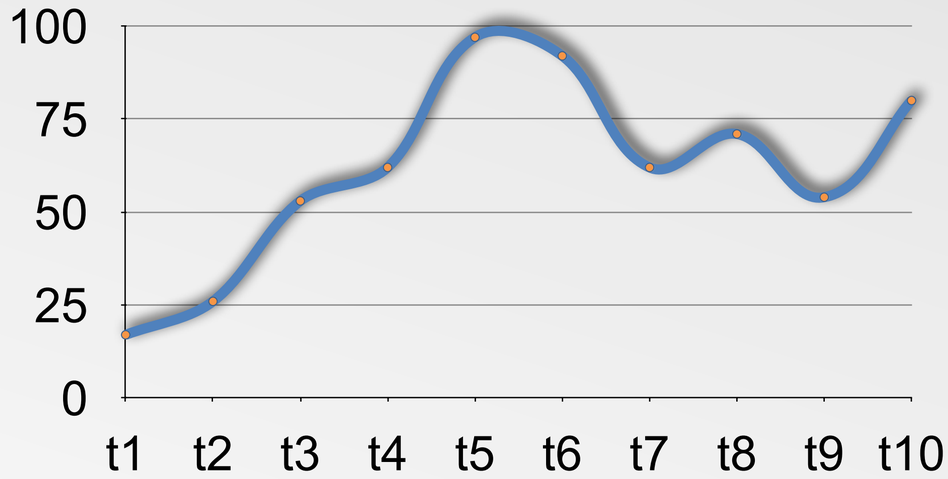
- Events represent the state of a particular data item or a discrete message/count
  - State
    - *State based event will only update when it changes. Most events are states.*
  - Discrete
    - *Discrete events (with representation="DISCRETE") will publish each time the event occurs. These are messages and counts where the count represents a change as in repeating a count of 1 for each part. There are only two types that are applicable: COUNT and MESSAGE.*

```
<DataItem category="EVENT" id="mode" name="mode" type="CONTROLLER_MODE"/>

<DataItem category="EVENT" id="pc1" name="pc" type="PART_COUNT"
          representation="DISCRETE"/>
```

Data Items - Samples

- A sample represents a continuous variable that is sampled at a given rate
- Samples are always floating point numeric values
- Samples **MUST** have units – for each data type units are specified
  - Temperature is Celsius, Position is Millimeters, etc…
- Samples are only reported when the values change
- Representation can be set to TIME_SERIES – multiple values can be given per update
  - Have tested 8khz Audio – See adapter labs…

```
<DataItem category="SAMPLE" coordinateSystem="WORK" id="pf"
          name="Fact" nativeUnits="FOOT/MINUTE"
          subType="ACTUAL" type="PATH_FEEDRATE"
          units="MILLIMETER/SECOND"/>

<DataItem category="SAMPLE" id="audio" units="DECIBEL"
          representation="TIME_SERIES"
          type="DISPLACEMENT" sampleRate="44100" />
```



Data Items – Condition

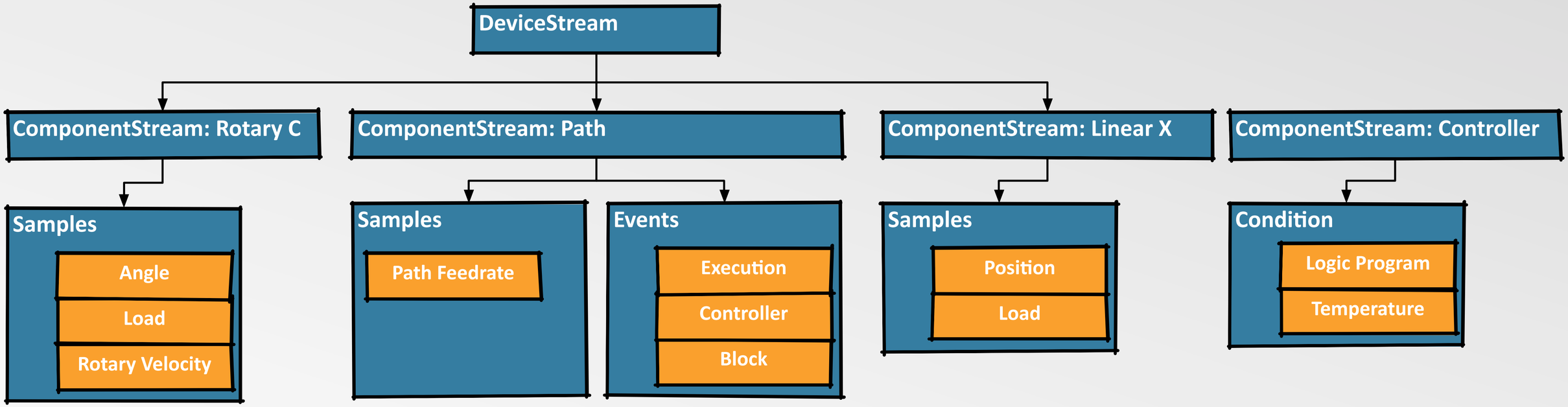
- Conditions represent the health of a component
- A condition has a type – for example a TEMPERATURE condition will report when a component is over or under temperature
- Conditions have three states: Normal, Warning, and Fault
- We will go over the meaning of the values when we discuss the streams document
- Types indicate what the condition is triggered on
- Source can reference the data item that provides the underlying trigger – like a spindle temperature

```
<DataItem category="CONDITION" id="c1sc" name="spt_cond" type="TEMPERATURE">
   <Source dataItemId="c1t" />
</DataItem>
…

<DataItem category="SAMPLE" id="c1t" name="c_temp" type="TEMPERATURE" units="CELSUIS"/>
```

MTConnectStreams

- The actual ***Values*** for each data item
- Flattened hierarchy to reduce protocol overhead
- Allows for complete history of changes to the data items
- Protocol header contains necessary information to have continuous data flow
- Refers back to MTConnectDevices document for additional information
- Balance between sufficient reference information and overhead
- ***Every Event, Sample, & Condition has a timestamp***

DeviceStream, ComponentStream, and Data



Preview… Sample and Current Request

- More on that in protocol – *Short Overview:*
  - Current provides the most recent value for each data item
  - Sample provides a stream of changes from a certain point in time
  - Both return the MTConnectStreams document
  - Current + Sample === Initial State + All Deltas (no missed data)
  - Use timestamps to align data and contextualize

MTConnect Streams Document

```
<MTConnectStreams xmlns:m="urn:mtconnect.org:MTConnectStreams:1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mtconnect.org:MTConnectStreams:1.3" xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.3 ../../MTConnectStreams_1.3.xsd">
  <Header creationTime="2010-03-04T18:58:50+00:00" sender="localhost" instanceId="1267728234" bufferSize="131072" version="1.1"
nextSequence="183806" firstSequence="52734" lastSequence="183805" />
  <Streams>
    <DeviceStream name="VMC-3Axis" uuid="000">
      <ComponentStream component="Rotary" name="C" componentId="c1">
        <Samples>
          <SpindleSpeed dataItemId="c2" name="Sspeed" sequence="135230" subType="ACTUAL" timestamp="2010-03-04T18:54:11.677212">3400.0000000000</
SpindleSpeed>
        </Samples>
        <Events>
          <RotaryMode dataItemId="cm" name="Cmode" sequence="2" timestamp="2010-03-04T18:43:54.178023">SPINDLE</RotaryMode>
        </Events>
        <Condition>
          <Normal dataItemId="Cload" sequence="62" timestamp="2010-03-04T18:44:24.187733" type="LOAD" />
        </Condition>
      </ComponentStream>
      <ComponentStream component="Path" name="path" componentId="pth">
        <Samples>
          <PathFeedrate dataItemId="Fovr" sequence="55" timestamp="2010-03-04T18:44:24.187733">100.0000000000</PathFeedrate>
          <PathFeedrate dataItemId="Frt" sequence="183801" timestamp="2010-03-04T18:58:50.749116">0.4</PathFeedrate>
          <PathPosition dataItemId="Ppos" sequence="3" subType="ACTUAL" timestamp="2010-03-04T18:43:54.178023">UNAVAILABLE</PathPosition>
        </Samples>
        <Events>
          <Block dataItemId="cn2" name="block" sequence="183791" timestamp="2010-03-04T18:58:50.689112">X0.327005 Y-0.359532</Block>
          <ControllerMode dataItemId="cn3" name="mode" sequence="53" timestamp="2010-03-04T18:44:24.187733">AUTOMATIC</ControllerMode>
          <Line dataItemId="cn4" name="line" sequence="183786" timestamp="2010-03-04T18:58:50.689112">764</Line>
          <Program dataItemId="cn5" name="program" sequence="52" timestamp="2010-03-04T18:44:24.187733">FLANGE_CAM.NGC</Program>
          <Execution dataItemId="cn6" name="execution" sequence="178023" timestamp="2010-03-04T18:58:27.319781">ACTIVE</Execution>
        </Events>
      </ComponentStream>
      …
```

Header

- All MTConnect XML Documents contain a Header
- Contains important protocol information
  - We will get into detail on the MTConnect protocol in the next section…
- Sequence numbers allow for continuous stream processing of data
- Instance Id indicates when the sequence number resets (usually the agent restarted)
- Version is the version of the MTConnect standard currently in use
- Sender is the machine name and creationTime says when the document was created

```
<Header creationTime="2010-03-04T18:58:50+00:00"
        sender="localhost"
        instanceId="1267728234"
        bufferSize="131072"
        version="1.3.0"
        nextSequence="183806"
        firstSequence="52734"
        lastSequence="183805" />
```

Streams

- Streams can have multiple **DeviceStream** elements – one for each device we are receiving data from
- Each request can give results for multiple devices
- For each device, each component of that device is given as a **ComponentStream** with the name, id, and component type (the element name in the MTConnectDevices doc)

```
<Streams>
  <DeviceStream name="VMC-3Axis" uuid="000">
    <ComponentStream component="Rotary" name="C" componentId="c1">
```

Component Stream

- Three Sections – Maps to Category
  - Events
  - Samples
  - Condition
- For samples and events, there can only be one value at any given timestamp
- Each Event, Sample, or Condition will have a unique sequence number
- Sequence numbers are unique for a given instance of an agent
- The sequence number monotonically increases for every piece of data that arrives at the agent
- Sequence numbers are **NOT** the same as timestamp
  - Timestamp is set when the data was observed
  - Sequence number is set when the data arrives at the agent
  - Sequence numbers are used for the protocol

Component Stream

- MTConnectDevices Data Item types are translated to the MTConnectStreams element names with the following rule:
  - Capitalize each word and remove underscores
- Values are grouped by category
- SubType and name are repeated for convenience – **dataItemId should be used as reference**

```
<DataItem type="CONTROLLER_MODE" category="EVENT" id="m1" name="mode"/>

<ControllerMode dataItemId="m1" name="mode" sequence="53"
                timestamp="2010-03-04T20:00:24.155470">AUTOMATIC</ControllerMode>

<DataItem type="EXECUTION" category="EVENT" id="e1" name="exec"/>

<Execution dataItemId="e1" name="exec" sequence="53"
                timestamp="2010-03-04T20:00:24.155470">ACTIVE</Execution>
```

Events

- Refresher – State Events and Discrete Events
- Values of most state events are “Controller Vocabularies” – consistent semantics
- State Events

```
<DataItem category="EVENT" id="mode" name="mode" type="CONTROLLER_MODE"/>
```

- Adjacent values **MUST** be different – All data is delta compressed (*except discrete…*)

```
<ControllerMode dataItemId="cn3" name="mode" sequence="53" timestamp="2010-03-04T20:00:24.155470">AUTOMATIC</ControllerMode>
<ControllerMode dataItemId="cn3" name="mode" sequence="64" timestamp="2010-03-04T20:00:25.435553">AUTOMATIC</ControllerMode>
```

- Correct

```
<ControllerMode dataItemId="cn3" name="mode" sequence="53" timestamp="2010-03-04T20:00:24.155470">AUTOMATIC</ControllerMode>
<ControllerMode dataItemId="cn3" name="mode" sequence="64" timestamp="2010-03-04T20:00:25.435553">MANUAL</ControllerMode>
<ControllerMode dataItemId="cn3" name="mode" sequence="78" timestamp="2010-03-04T20:00:26.435553">MANUAL_DATA_INPUT</ControllerMode>
```

Discrete Events

- Message and Counts can be Discrete

```
<DataItem category="EVENT" id="ct1" name="count" type="PART_COUNT" representation="DISCRETE"/>
<DataItem category="EVENT" id="m1" name="msg" type="MESSAGE" representation="DISCRETE"/>
```

- Values are allowed to Repeat
  - **Note: Discrete is appended to type**

```
<PartCountDiscrete dataItemId="ct1" name="count" sequence="30777"
        timestamp="2010-03-04T20:02:03.664934">1</PartCountDiscrete>
<PartCountDiscrete dataItemId="ct1" name="count" sequence="30785"
        timestamp="2010-03-04T20:02:04.664934">1</PartCountDiscrete>
<PartCountDiscrete dataItemId="ct1" name="count" sequence="30792"
        timestamp="2010-03-04T20:02:05.664934">1</PartCountDiscrete>
```

- Meaning – Three parts were made
- Message Example – Door open message occurred twice

```
<MessageDiscrete dataItemId="m1" name="msg" sequence="30777"
        timestamp="2010-03-04T20:02:03.664934">Door Open</MessageDiscrete>
<MessageDiscrete dataItemId="m1" name="msg" sequence="30785"
        timestamp="2010-03-04T20:02:04.664934">Door Open</MessageDiscrete>
```

Samples

- Timestamp is always set to the time the value is collected

```
<DataItem category="SAMPLE" id="Frt" nativeUnits="MILLIMETER/SECOND" type="PATH_FEEDRATE" units="MILLIMETER/SECOND">
```

- The units are constant for any standardized data item type

```
<Samples>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.670137" sequence="1843805305">0.2986444</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.682588" sequence="1843805311">0.2974917</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.694910" sequence="1843805317">0.296339</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.707282" sequence="1843805322">0.3716662</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.719648" sequence="1843805327">0.4</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.843346" sequence="1843805368">0.364326</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.855673" sequence="1843805373">0.2996606</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.868173" sequence="1843805379">0.2995241</PathFeedrate>
 <PathFeedrate dataItemId="Frt" timestamp="2015-06-06T05:35:43.880672" sequence="1843805385">0.3358733</PathFeedrate>
</Samples>
```

- Like with events – values will not be reported unless they have changed
- If the timestamp is in the past, it means the value has not changed since that time

Time Series Samples – High Frequency Sensor Data

- Multiple values at a time
- For time series, the timestamp is **ALWAYS** the exact time of the **LAST** observation

```
<DataItem category="SAMPLE" id="audio" units="DECIBEL" representation="TIME_SERIES" type="DISPLACEMENT"
sampleRate="44100" />
```

- This is 44khz audio sampling with 2048 samples – sample rate is in hz

```
<DisplacementTimeSeries dataItemId="audio" timestamp="2011-03-07T22:57:39.6766722Z" sequence="1032"
sampleCount="2048">0.00154704 0.00104351 0.000870681 0.000980529 0.00133584 0.00193649 0.00270566 0.00336045
0.00355552 0.00317654 0.00233925 0.00136575 0.000626581 0.000177587 -5.82915e-05 1.82085e-06 0.000472774 0.00125045
0.00203212 0.00259833 0.00277265 0.00244583 0.00186203 0.00141206 0.00122931 0.00116704 0.00111445 0.00118448
0.00149118 0.00203722 0.00270667 0.00313162 0.00300439 0.00242456 0.00162871 0.000781631 -0.000107176 -0.00100444
-0.00159512 -0.00158403 -0.000924871 0.000168987 0.0013498 0.00234963 0.00302059 0.00346951 0.00384501 0.00404805
0.00405175 0.00386467 0.00339448 0.00279493 0.00224661 0.00197666 0.00235257 0.00333088 0.00447948 0.00533796
0.00572051 0.00573359 0.00568443 0.0057683 0.00578263 0.00540111 0.00447592 0.00333043 0.00258744 0.00270065
0.00364354 0.00480498 0.00551655 0.00546966 … </DisplacementTimeSeries>
```

TimeSeries is appended to type

T10:07:22.120	T10:07:22.130	T10:07:22.140	T10:07:22.150
12.314	10.222	17.234	6.889

Time-stamp of Sample

Time of first sample is timestamp - sampleCount / sampleRate (hz)

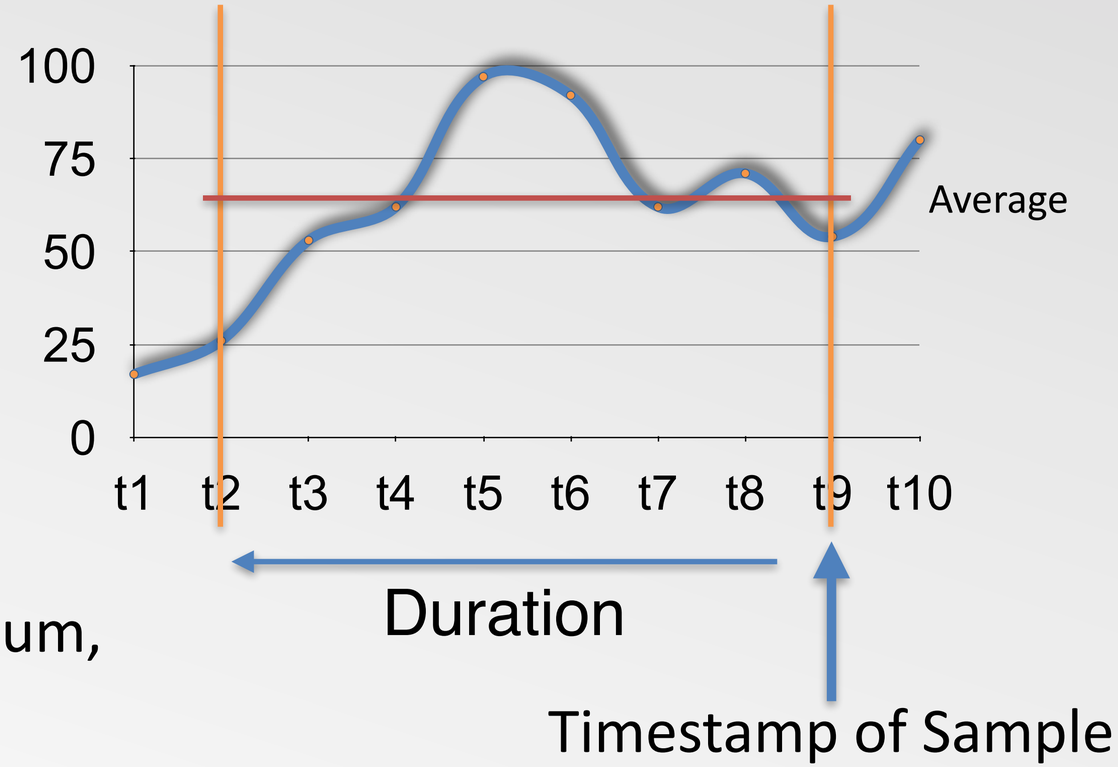
Sample with Statistics

- Statistics provide the ability to express an simple analytical method over a period of time

```
<DataItem category="SAMPLE" id="audio_avg" units="DECIBEL" type="DISPLACEMENT" statistic="AVERAGE" />
```

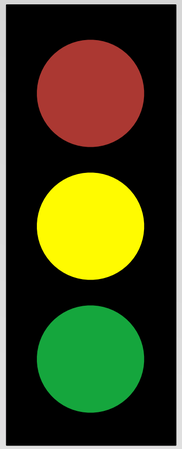
- The stream value carries the attribute to distinguish it and a duration
- To compute the beginning of the interval, you must subtract the duration from the timestamp

```
<Displacement dataItemId="audio_avg"
  timestamp="2011-03-07T22:57:39.6766722Z"
  sequence="1035" statistic="AVERAGE"
  duration="0.0464399093">-2.544695307E-05</Displacement>
```



- Current set of statistics in the standard:
  - Average, Kurtosis, Maximum, Median, Minimum, Mode, Range, Root Mean Square, and Standard Deviation

Conditions



- States:

Normal	The component is within normal operating parameters
Warning	The component is approaching failure, but can self correct without intervention
Fault	The component has failed and manual intervention is required to resolve the issue

- A simple model was chosen because the definitions are defensible
- Conditions are naturally contextualized with their component
  - For example: Rotary C Axis (Spindle)

```
<Normal dataItemId="clp" sequence="66" timestamp="2010-03-04T20:00:24.157257" type="LOAD" />
<Warning dataItemId="clp" sequence="66" timestamp="2010-03-04T20:00:24.157257" type="LOAD"
         nativeCode="SPL-19234" qualifier="HIGH">Load is high</Warning>
<Fault dataItemId="clp" sequence="66" timestamp="2010-03-04T20:00:24.157257" type="LOAD"
       nativeCode="SPL-20084"  qualifier="HIGH" nativeSeverity="1000">Overload</Fault>
```

Multiple Conditions

- Conditions are special since there can be multiple conditions present at the same time

```
<Condition>
  <Fault sequence="7754" timestamp="2015-06-05T11:32:56.55343Z" dataItemId="c2128" type="MOTION_PROGRAM"
         nativeCode="BRX13-1167" name="logic">Syntax Error on line 1167</Fault>
  <Fault sequence="7754" timestamp="2015-06-05T11:32:56.55343Z" dataItemId="c2128" type="MOTION_PROGRAM"
         nativeCode="BRX13-1170" name="logic">Syntax Error on line 1170</Fault>
  <Fault sequence="7754" timestamp="2015-06-05T11:32:56.55343Z" dataItemId="c2128" type="MOTION_PROGRAM"
         nativeCode="BRX13-1169" name="logic">Syntax Error on line 1169</Fault>
</Condition>
```

- In the current implementation, the conditions must be unique by nativeCode
- A Normal **without** a nativeCode all conditions of that type
- A Normal **with** a nativeCode clears only that one condition

```
<Condition>
  <Fault sequence="7754" timestamp="2015-06-05T11:32:56.55343Z" dataItemId="c2128" type="MOTION_PROGRAM"
         nativeCode="BRX13-1167" name="logic">Syntax Error on line 1167</Fault>
  <Fault sequence="7754" timestamp="2015-06-05T11:32:56.55343Z" dataItemId="c2128" type="MOTION_PROGRAM"
         nativeCode="BRX13-1170" name="logic">Syntax Error on line 1170</Fault>
  <Normal sequence="7754" timestamp="2015-06-05T11:32:56.55343Z" dataItemId="c2128" type="MOTION_PROGRAM"
         nativeCode="BRX13-1169" name="logic"/>
</Condition>
```

Data availability

- MTConnect has one required data item: Availability
  - It can have two values: AVAILABLE or UNAVAILABLE
  - **Remember: all other data items are optional**

```
<Availability dataItemId="avail" timestamp="…" sequence="…">AVAILABLE</Availability>
```

- Any data item can be UNAVAILABLE if its value is unknown or data is not available
  - In this case there is no Z axis load reporting from the machine

```
<Load dataItemId="z4" timestamp="…" name="Zload" sequence="46">UNAVAILABLE</Load>
```

- All Samples, Events, and Conditions must support UNAVAILABLE
  - Condition example

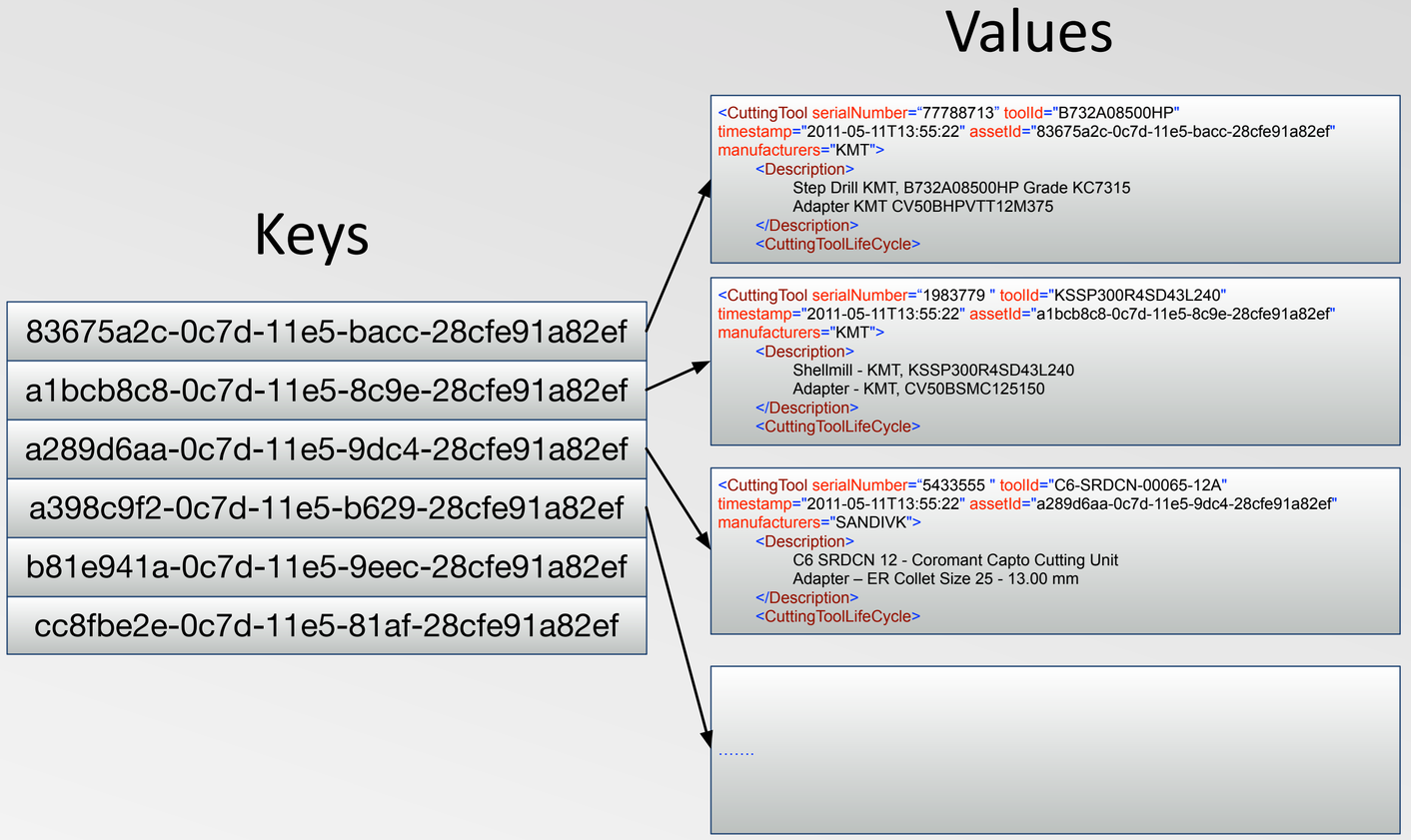
```
<Unavailable dataItemId="Zsystem" timestamp="…" sequence="11" type="SYSTEM"/>
```

Assets

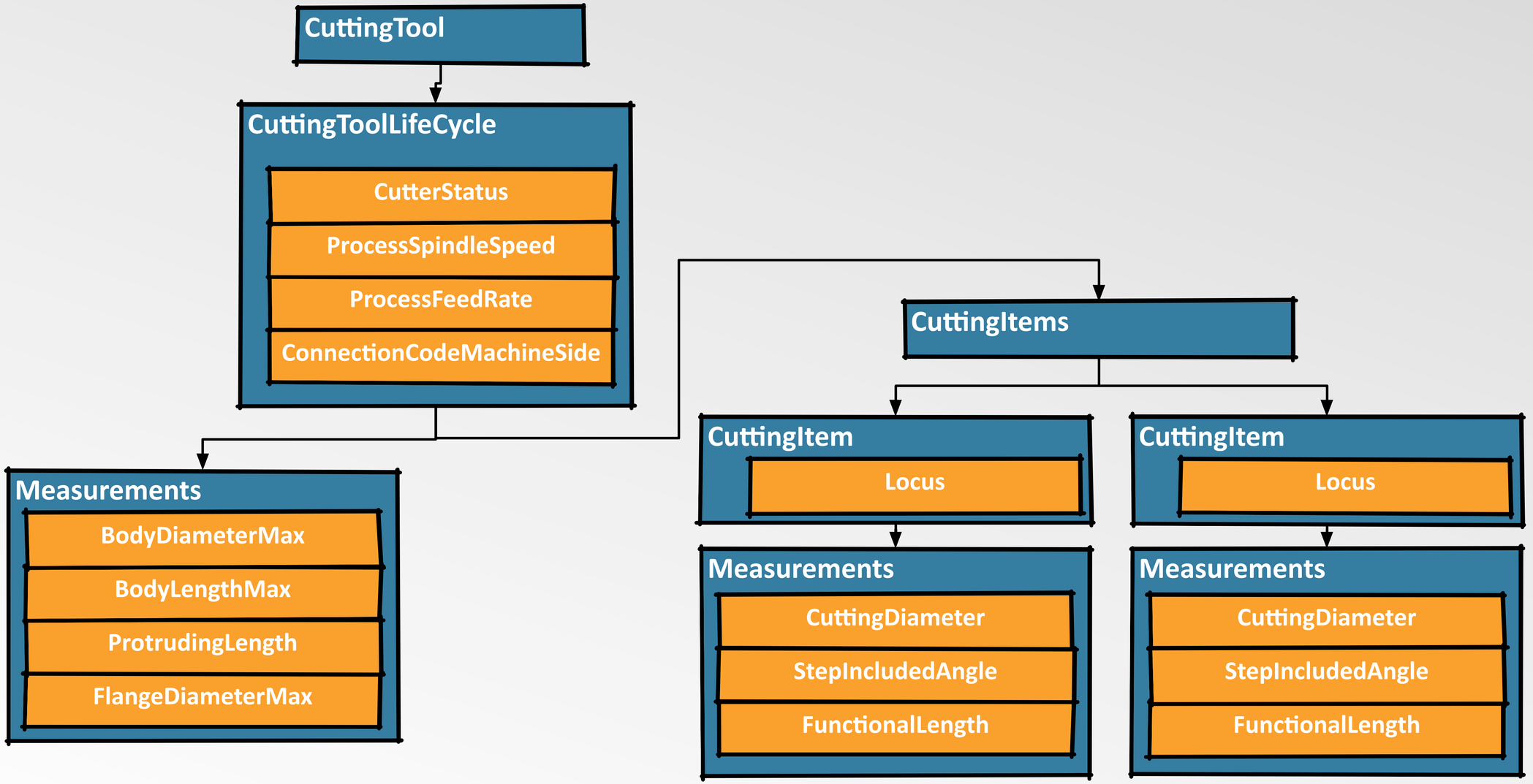
- MTConnect breaks manufacturing domain models down by their content type and their rates of change
  - The meta data (MTConnectDevices) changes very infrequently
  - The real-time (MTConnectStreams) data changes very frequently
  - The assets (MTConnectAssets) change fairly frequently
- Assets provide a place to store and communicate rich domain models, some examples:
  - Cutting Tools
  - Parts
  - Inspection and Verification
  - Workholding
  - Coordinate System Transformations – Offsets
- Currently we support Cutting Tools, Parts and Inspection are being develop and will be released towards the end of 2015

Key / Value Storage

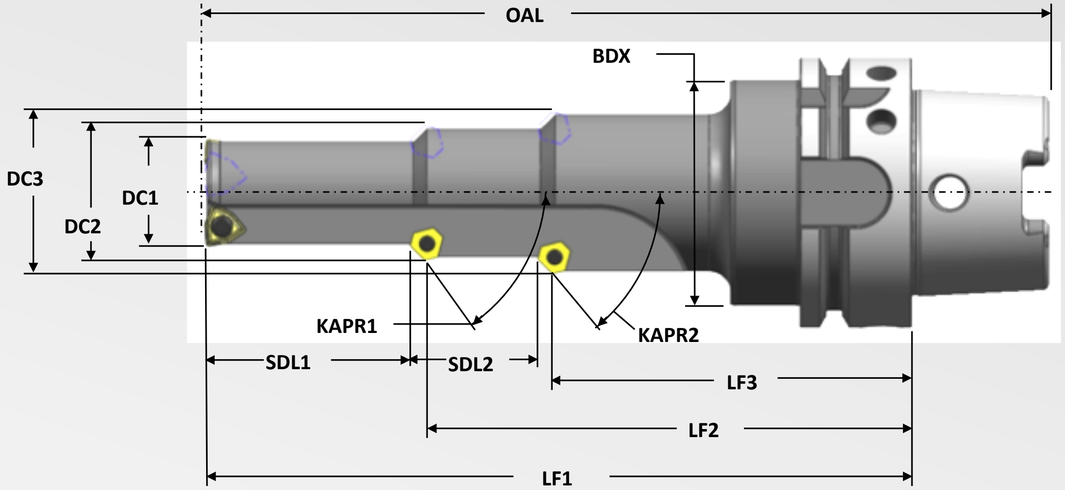
- Same mechanism used in most NoSQL databases – MongoDB, Cassandra, HFS, etc…
- Keys are globally unique and are associated with the asset for its lifetime
- Notifications on additions, updates, and deletes

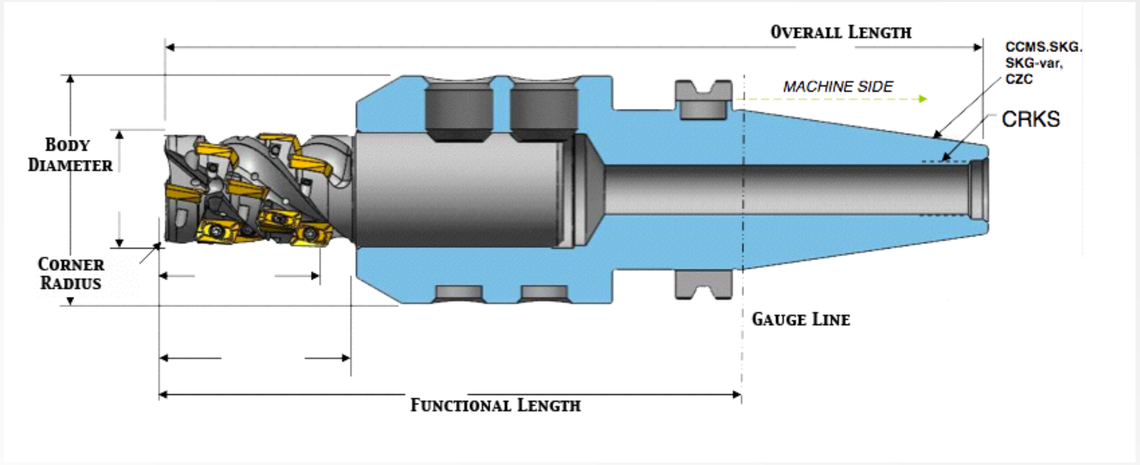


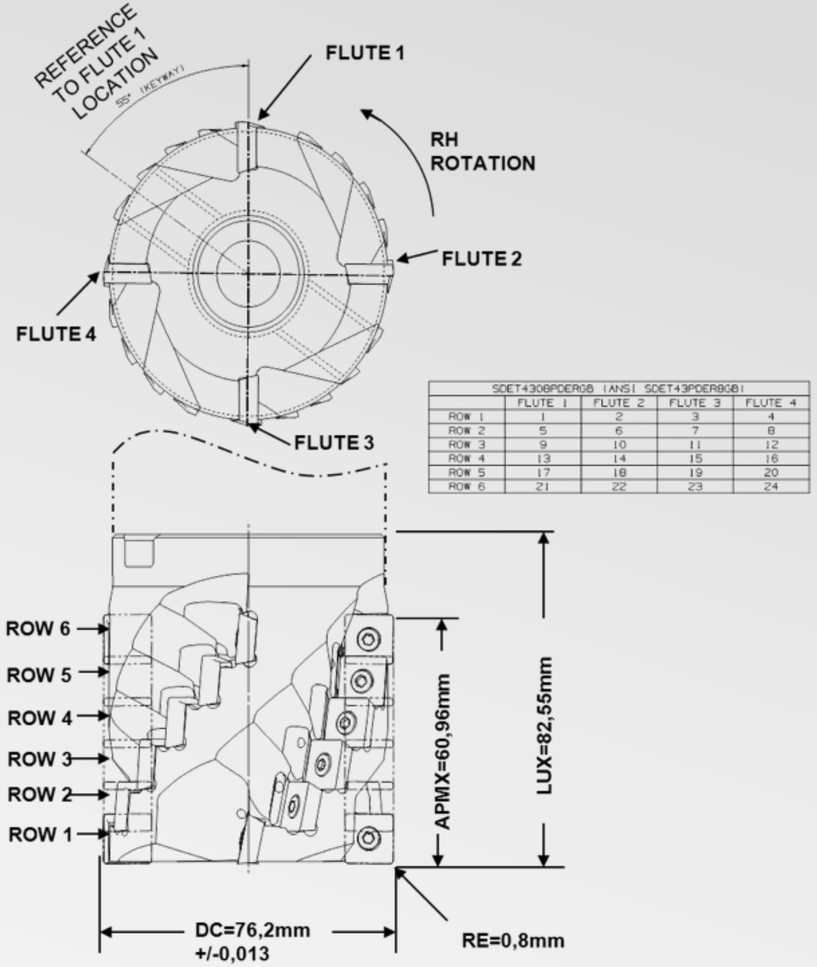
CuttingTool



Cutting Tools







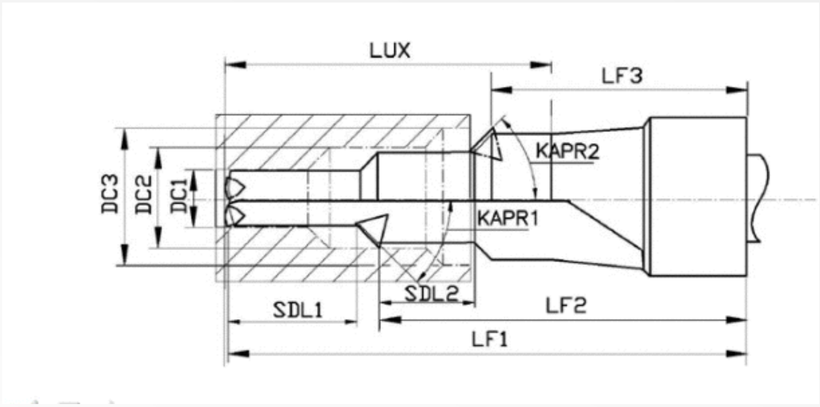
SDET4308PDERGB (ANSI SDET43PDERGB)

	FLUTE 1	FLUTE 2	FLUTE 3	FLUTE 4
ROW 1	1	2	3	4
ROW 2	5	6	7	8
ROW 3	9	10	11	12
ROW 4	13	14	15	16
ROW 5	17	18	19	20
ROW 6	21	22	23	24

Two Models: Archetype and Instance

- Archetypes are templates for all assets of that type – nominal or idealized
- Instances are individual assets – measured
- MTConnect models complete assemblies
- Augments ISO 13399 Cutting Tool Geometries with Application/Use data
- Uses ISO 13399 codes for measurements and geometries
- Instance can reference back to archetype

Archetype

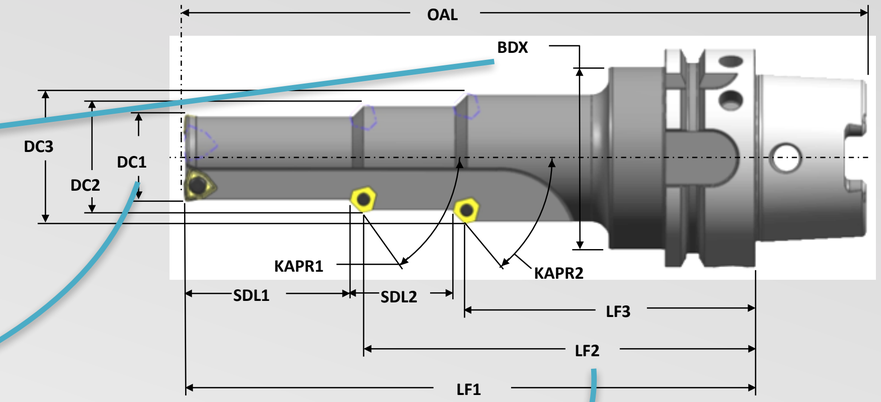


Instance

Translating a Tool to XML…

```
<CuttingTool serialNumber="1 " toolId="B732A08500HP" timestamp="2011-05-11T13:55:22" assetId="B732A08500HP.1" manufacturers="KMT">
    <Description>
        Step Drill KMT, B732A08500HP Grade KC7315
        Adapter KMT CV50BHPVTT12M375
    </Description>
    <CuttingToolLifeCycle>
        <CutterStatus><Status>NEW</Status></CutterStatus>
        <ProcessSpindleSpeed nominal="5893">5893</ProcessSpindleSpeed>
        <ProcessFeedRate nominal="2.5">2.5</ProcessFeedRate>
        <ConnectionCodeMachineSide>CV50 Taper</ConnectionCodeMachineSide>
        <Measurements>
            <BodyDiameterMax code="BDX">3[illegible]</BodyDiameterMax>
            <BodyLengthMax code="LBX" nominal="120.825" maximum="126.325" minimum="115.325">120.825</BodyLengthMax>
            <ProtrudingLength code="LPR" nominal="155.75" maximum="161.25" minimum="150.26">158.965</ProtrudingLength>
            <FlangeDiameterMax code="DF" nominal="98.425">98.425</FlangeDiameterMax>
            <OverallToolLength nominal="257.35" minimum="251.85" maximum="262.85" code="OAL">257.35</OverallToolLength>
        </Measurements>
        <CuttingItems count="2">
            <CuttingItem indices="1" manufacturers="KMT" grade="KC7315">
                <Measurements>
                    <CuttingDiameter code="DC1" nominal="8.5" maximum="8.521" minimum="8.506">8.513</CuttingDiameter>
                    <StepIncludedAngle code="STA1" nominal="90" maximum="91" minimum="89">89.8551</StepIncludedAngle>
                    <FunctionalLength code="LF1" nominal="154.286" minimum="148.786" maximum="159.786">157.259</FunctionalLength>
                    <StepDiameterLength code="SDL1" nominal="9">9</StepDiameterLength>
                    <PointAngle code="SIG" nominal="135" minimum="133" maximum="137">135.1540</PointAngle>
                </Measurements>
            </CuttingItem>
            <CuttingItem indices="2" manufacturers="KMT" grade="KC7315">
                <Measurements>
                    <CuttingDiameter code="DC2" nominal="12" maximum="12.011" minimum="12">11.999</CuttingDiameter>
                    <FunctionalLength code="LF2" nominal="122.493" maximum="127.993" minimum="116.993">125.500</FunctionalLength>
                    <StepDiameterLength code="SDL2" nominal="9">9</StepDiameterLength>
                </Measurements>
            </CuttingItem>
        </CuttingItems>
    </CuttingToolLifeCycle>
</CuttingTool>
```

Use Data



Archetype Example

```
<CuttingToolArchetype assetId="83675a2c-0c7d-11e5-bacc-28cfe91a82ef" timestamp="2015-06-04T13:29:12.12Z" toolId="SX12566644"
deviceUuid="cc8fbe2e-0c7d-11e5-81af-28cfe91a82ef">
    <Description>
        Shellmill - KMT, KSSP300R4SD43L240
        Adapter - KMT, CV50BSMC125150
    </Description>
    <CuttingToolDefinition>
        ISO 13399 STEP Model
    </CuttingToolDefinition>
    <CuttingToolLifeCycle>
        <ProcessSpindleSpeed maximum="13300" nominal="605"/>
        <ProcessFeedRate nominal="9.22"/>
        <ConnectionCodeMachineSide>CV50</ConnectionCodeMachineSide>
        <Measurements>
            <BodyDiameterMax code="BDX">73.25</BodyDiameterMax>
            <OverallToolLength nominal="222.25" minimum="221.996" maximum="222.504" code="OAL"/>
            <UsableLengthMax code="LUX" nominal="82.55"/>
            <CuttingDiameterMax code="DC" nominal="76.2" maximum="76.213" minimum="76.187"/>
            <FunctionalLength code="LF" nominal="120.65" maximum="120.904" minimum="120.396"/>
            <DepthOfCutMax code="APMX" nominal="60.96"/>
            <FlangeDiameterMax code="DF" nominal="98.425"/>
        </Measurements>
        <CuttingItems count="24">
            <CuttingItem indices="1-24" itemId="SDET43PDER8GB" manufacturers="KMT" grade="KC725M">
                <Measurements>
                    <CuttingEdgeLength code="L" nominal="12.7" minimum="12.675" maximum="12.725"/>
                    <WiperEdgeLength code="BS" nominal="2.56"/>
                    <IncribedCircleDiameter code="IC" nominal="12.7"/>
                    <CornerRadius code="RE" nominal="0.8"/>
                </Measurements>
        ....
```

Only nominal, minimum and max

No status or tool life

Cutting tool definition can hold any content

Asset Events

- When an asset is added or updated, an AssetChanged event will be generated
  - This is automatically handled by the agent
- When an asset is removed, an AssetRemoved event is generated
  - This is also automatically handled by the agent
- These events carry the asset type and the asset id
- The asset can be accessed using the asset API (covered in the next section on Protocol)

```
<AssetChanged dataItemId="dev_asset_chg" timestamp="2015-03-04T04:59:26.531505Z" sequence=“27
              assetType="CuttingTool">8b65a788-0c91-11e5-927c-28cfe91a82ef</AssetChanged>
<AssetRemoved dataItemId="dev_asset_rem" timestamp="2015-03-04T04:59:26.531505Z" sequence="28"
              assetType="CuttingTool">8cb848de-0c91-11e5-bbf9-28cfe91a82ef</AssetRemoved>
```

New Asset Types

- Asset model can be extended to hold any content
- New schemas need to extend the abstract asset type
- Only asset id and timestamp are required
  - Device uuid is recommended to identify source

Errors

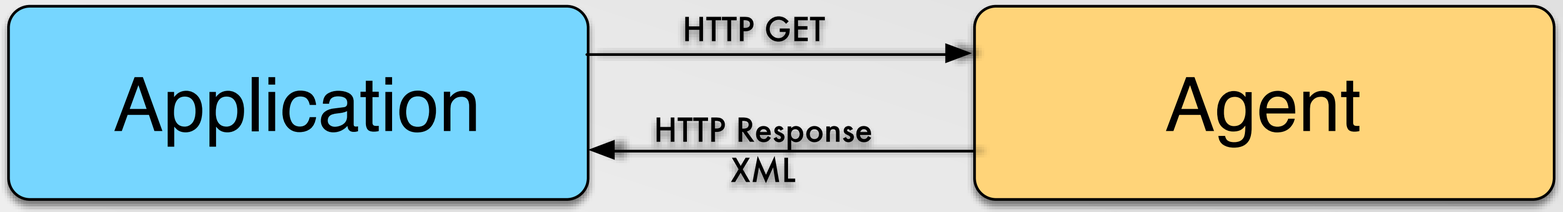
- Multiple errors can be reported at the same time
- Possible values for errorCode:
  - UNAUTHORIZED
  - NO_DEVICE
  - OUT_OF_RANGE
  - TOO_MANY
  - INVALID_URI
  - INVALID_REQUEST
  - INTERNAL_ERROR
  - INVALID_PATH
  - UNSUPPORTED
  - ASSET_NOT_FOUND

```
<MTConnectError xmlns="urn:mtconnect.org:MTConnectError:1.3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mtconnect.org:MTConnectError:1.3
MTConnectError_1.3.xsd">
   <Header version="1.3" creationTime="2001-12-17T09:30:47Z"
instanceId="1" sender="String" bufferSize="1"/>
   <Errors>
      <Error errorCode="ASSET_NOT_FOUND">Asset not found</Error>
   </Errors>
</MTConnectError>
```

Protocol

MTConnect Communications Architecture

- MTConnect Specifies the Communication from the Agent to the Application
- Application makes an HTTP request → Agent Responds



- Uses REST (REpresentational State Transfer) semantics – **client responsible for state**
- Agent is a special purpose HTTP server
- Open Source implementation of Agent available
- Response in XML
- Store and forward with publish / subscribe semantics
  - Can continuously stream sample or current
- Key value document storage

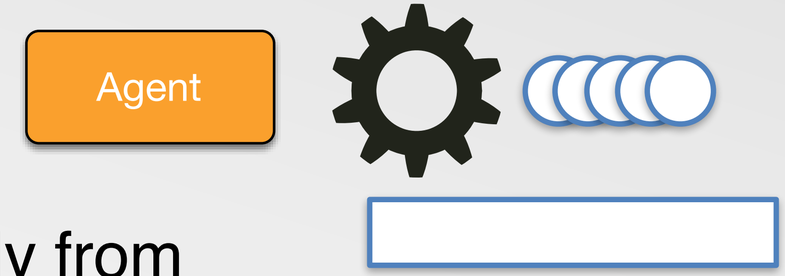
Buffering…

MTConnect Agent Buffers Data so Application Can Collect Data Less Frequently

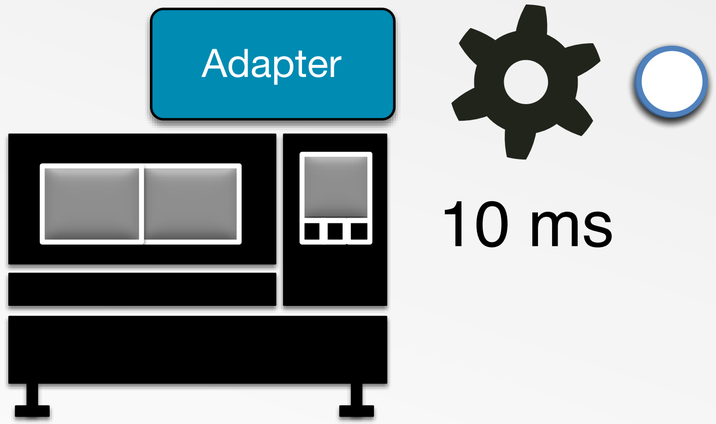
1 s

Application

Application can poll or get push updates



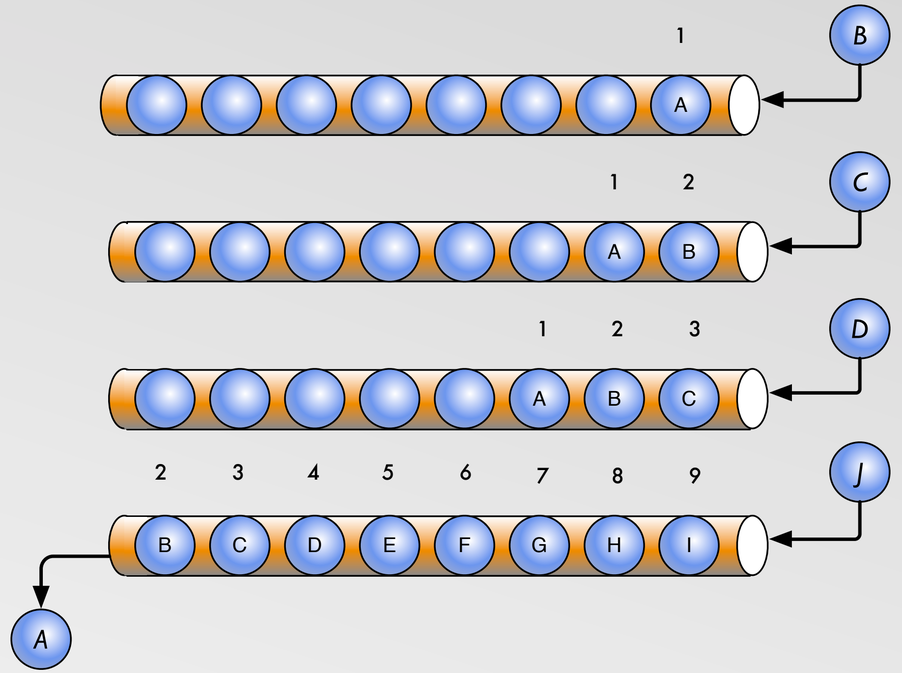
Adapter Collect Data Rapidly from Devices and Sensors

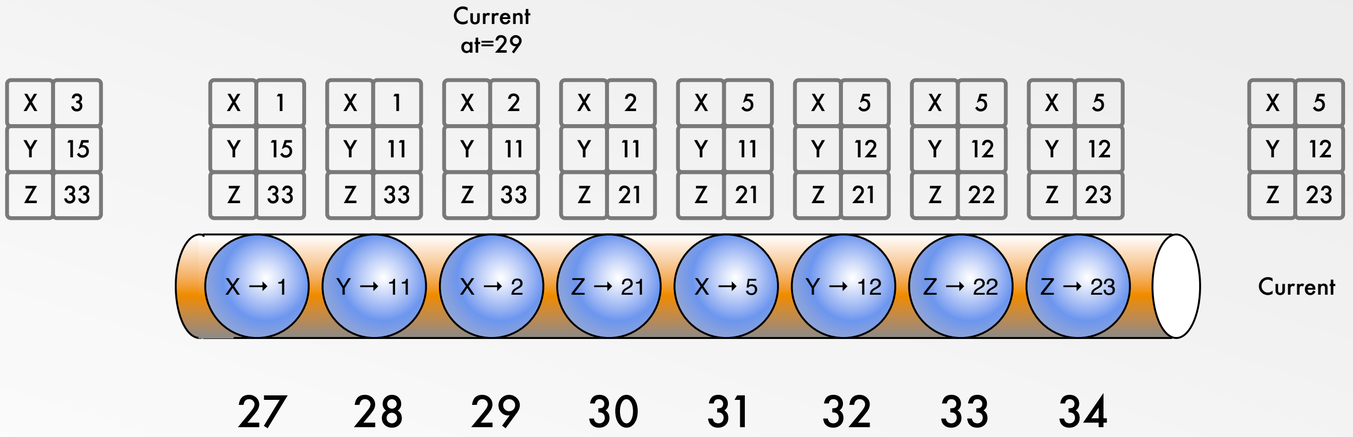


- *Benefits:*
  - *Less overhead*
  - *No missed data*
  - *Fault tolerance*

Event Management

- Events are stored in a circular buffer
  - Oldest events are removed once buffer is full
- Benefits
  - Fixed resource consumption
  - Provides limited history for recovery
  - Allows request of state at a particular point in time
  - Never miss a single event





Requests

- probe
  - MTConnectDevices
- current
  - MTConnectStreams at a point in time
  - Can also ask for the current data at a position in the buffer
- sample
  - MTConnectStreams for a given number of entries
- asset
  - Information models – Cutting Tools, Parts, …
- Usual order of operations:
  - probe → current → sample → sample -> …
  - probe → current → sample w/ interval (push based streaming)

Probe

- Requests the MTConnectDevices model – by default all devices
  - http://agent.mtconnect.org/probe
- For a single device
  - http://agent.mtconnect.org/VMC-3Axis/probe
- Every time you reconnect to the MTConnect Agent, it is suggested you probe first
- Use the **instanceId** to detect a agent restart
- If the instanceId has changed – there is a chance the MTConnectDevices may have changed – otherwise it is the same as before

```
<Header creationTime="2015-06-07T04:38:22Z" sender="mtcagent" instanceId="1425445166"
        version="1.3.0.9" bufferSize="131072" />
```

Current

- Snapshot of the data at a point in time
- One value per data item – all active conditions will be given
- Get the most recent values for all devices
  - http://agent.mtconnect.org/current
- Get the most recent values for one device
  - http://agent.mtconnect.org/VMC-3Axis/current
- Get the values at a certain sequence number
  - Allow rollback of history to view state at that point
  - Must be between the first and last sequence numbers
  - http://agent.mtconnect.org/current?at=1862633266

```
<Header creationTime="2015-06-07T04:38:22Z" sender="mtcagent" instanceId="1425445166"
        version="1.3.0.9" bufferSize="131072"
        nextSequence="1862658780" firstSequence="1862527708"
        lastSequence="1862658779"/>
```

Sample

- By default, sample will start from the first item in the buffer
- For a continuous stream of data, use the nextSequence number from the current as the starting point
- Arguments:
  - from – start at this sequence number
  - count – return at most this number of items at a time
- Example

```
<Header creationTime="2015-06-07T04:38:22Z" sender="mtcagent" instanceId="1425445166"
        version="1.3.0.9" bufferSize="131072"
        nextSequence="1862658780" firstSequence="1862527708"
        lastSequence="1862658779"/>
```

  - http://agent.mtconnect.org/VMC-3Axis/sample?from=1862658780&count=1000
  - Gets the next 1000 items starting at 1862658780
  - The header will give the nextSequence number to continue from …

Polling Example

- current

```
<Header creationTime="2015-06-07T05:23:08Z" sender="mtcagent"
    instanceId="1425445166" version="1.3.0.9" bufferSize="131072"
    nextSequence="1863257116" firstSequence="1863126044" lastSequence="1863257115"/>
```

- sample?from=**1863257116**&count=1000

```
<Header creationTime="2015-06-07T05:24:08Z" sender="mtcagent"
    instanceId="1425445166" version="1.3.0.9" bufferSize="131072"
    nextSequence="1863258116" firstSequence="1863145841" lastSequence="1863276912"/>
```

- sample?from=**1863258116**&count=1000

```
<Header creationTime="2015-06-07T05:25:14Z" sender="mtcagent"
    instanceId="1425445166" version="1.3.0.9" bufferSize="131072"
    nextSequence="1863259116" firstSequence="1863161083" lastSequence="1863292154"/>
```

- When nextSequence = lastSequence + 1, you have reached the end of the buffer…

```
<Header creationTime="2015-06-07T05:29:44Z" sender="mtcagent"
    instanceId="1425445166" version="1.3.0.9" bufferSize="131072"
    nextSequence="1863368548" firstSequence="1863237476" lastSequence="1863368547"/>
```

- If there is no new data, you will receive an empty streams document

```
<Streams/>
```

Continuous Stream – Pull Based

- As long as the instanceId remains the same, the sequence numbers will be ever increasing and you can resume where you left off – unless you fall too far behind…

```
<?xml version="1.0" encoding="UTF-8"?>
<MTConnectError …>
  <Header creationTime="2015-06-07T05:42:53Z" sender="mtcagent" instanceId="1425445166"
          version="1.3.0.9" bufferSize="131072"/>
  <Errors>
    <Error errorCode="OUT_OF_RANGE">'from' must be greater than or equal to 1863404211.</Error>
  </Errors>
</MTConnectError>
```

XPath Filtering

- Current and Sample can deliver limited sets of data
- Useful when only a few data items are required or to filter quantity of data
- Uses the XPath standard from the W3C: http://www.w3.org/TR/xpath20/
- Data Items or Components can be selected
  - If a component is selected, all sub-components will also be included by default
  - Data Items and Components can be selected by the element and any attribute
- XPath is specified by providing the path=<xpath> argument in the url
- Specifying the device will limit the expression to the one device

XPath Example

- To select only axis data
  - path=//Axes
  - Will include Rotary C and Linear X
- To select only Controller Path data
  - path=//Path
- To select position data
  - path=//DataItem[@type='POSITION']
- To select position and angle
  - path=//DataItem[@type='POSITION' or @type='ANGLE']
- Only position in the Linear X axis
  - path=//Linear[@name="X"]/DataItems/DataItem[@type="POSITION"]

```
<Device id="dev" iso841Class="6" name="VMC-3Axis" sampleInterval="10" uuid="000">
 <Description manufacturer="SystemInsights"/>
 <DataItems>
    <DataItem category="EVENT" id="avail" type="AVAILABILITY"/>
    <DataItem category="EVENT" id="dev_asset_chg" type="ASSET_CHANGED"/>
    <DataItem category="EVENT" id="dev_asset_rem" type="ASSET_REMOVED"/>
 </DataItems>
 <Components>
    <Axes id="ax" name="Axes">
     <Components>
        <Rotary id="c1" name="C">
         <DataItems>
            <DataItem category="SAMPLE" id="c2" name="Sspeed" nativeUnits="REVOLUTION/MINUTE"
subType="ACTUAL" type="ROTARY_VELOCITY" units="REVOLUTION/MINUTE"/>
         </DataItems>
        </Rotary>
        <Linear id="x1" name="X">
         <DataItems>
            <DataItem category="SAMPLE" id="x2" name="Xact" nativeUnits="MILLIMETER"
subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
         </DataItems>
        </Linear>
     </Components>
    </Axes>
    <Controller id="cn1" name="controller">
     <Components>
        <Path id="pth" name="path">
         <DataItems>
            <DataItem category="EVENT" id="cn2" name="block" type="BLOCK"/>
            <DataItem category="EVENT" id="cn3" name="mode" type="CONTROLLER_MODE"/>
            <DataItem category="EVENT" id="cn6" name="execution" type="EXECUTION"/>
         </DataItems>
        </Path>
     </Components>
    </Controller>
 </Components>
</Device>
```

Push Based Streaming

- The MTConnect agent can push data at a given interval to the client application
- Benefits:
  - Lower protocol overhead – data is only pushed if it has changed
  - 10 second (default) heartbeat for connection keep alive – fault detection
- Uses x-multipart/replace
  - Older standard, now supplanted by WebSockets
- Request include the interval=<milliseconds> argument in the url

  http://agent.mtconnect.org/sample?count=1000&interval=1000
- You can change the heartbeat interval as well – default is 10000

  http://agent.mtconnect.org/sample?count=100&interval=1000&heartbeat=2000
- Back-pressure (not consuming fast enough) will cause agent to close stream
  - 8k default outgoing TCP buffer…

Agent Response

- Header has content type of multipart/x-mixed-replace
- MIME content encoding like with email
- Each “chunk” is separated by a **boundary** consisting of “--“ and then a unique string
- After the boundary is a brief MIME header with a Content type and length
- NOTE: The HTTP response **MUST NOT** have a content-length and **MUST** have Transfer-Encoding: chunked

```
HTTP/1.1 200 OK
Date: Sun, 07 Jun 2015 20:57:40 GMT
Server: MTConnectAgent
Expires: -1
Connection: close
Cache-Control: private, max-age=0
Content-Type: multipart/x-mixed-replace;boundary=efd6b911c36f73aae7cfd4f274c88204
Transfer-Encoding: chunked

--efd6b911c36f73aae7cfd4f274c88204
Content-type: text/xml
Content-length: 151730
```

Continuous Chunks…

```
--efd6b911c36f73aae7cfd4f274c88204
Content-type: text/xml
Content-length: 151730

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="/styles/Streams.xsl"?>
<MTConnectStreams xmlns:m="urn:mtconnect.org:MTConnectStreams:1.3"
xmlns="urn:mtconnect.org:MTConnectStreams:1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.3 /schemas/MTConnectStreams_1.3.xsd">

…

</MTConnectStreams>

--efd6b911c36f73aae7cfd4f274c88204
Content-type: text/xml
Content-length: 151654

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="/styles/Streams.xsl"?>
<MTConnectStreams xmlns:m="urn:mtconnect.org:MTConnectStreams:1.3"
xmlns="urn:mtconnect.org:MTConnectStreams:1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.3 /schemas/MTConnectStreams_1.3.xsd">
```

Heartbeat

- The agent will automatically wait up to heartbeat interval for data to arrive
- When there is no data to send (sample only), an empty streams will be sent as follows:

```
--ddad64b18ec4e092b1cba946b175baf6
Content-type: text/xml
Content-length: 606

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="/styles/Streams.xsl"?>
<MTConnectStreams xmlns:m="urn:mtconnect.org:MTConnectStreams:1.3" xmlns="urn:mtconnect.org:MTConnectStreams:1.3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.3 /
schemas/MTConnectStreams_1.3.xsd">
  <Header creationTime="2015-06-07T21:27:02Z" sender="mtcagent" instanceId="1425445166" version="1.3.0.9"
bufferSize="131072" nextSequence="1876410620" firstSequence="1876279548" lastSequence="1876410619"/>
  <Streams/>
</MTConnectStreams>
```

Interval with current

- A Current request can also be given an interval (though lesser utility)
- The current will send the most recent values of each data item at the frequency specified
  - It will never send an empty stream – so heartbeat does not apply

    http://agent.mtconnect.org/current?interval=1000

- at=… and interval will produce an error

```
<?xml version="1.0" encoding="UTF-8"?>
<MTConnectError xmlns:m="urn:mtconnect.org:MTConnectError:1.3" xmlns="urn:mtconnect.org:MTConnectError:1.3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mtconnect.org:MTConnectError:1.3 /schemas/
MTConnectError_1.3.xsd">
  <Header creationTime="2015-06-07T21:35:47Z" sender="mtcagent" instanceId="1425445166" version="1.3.0.9"
bufferSize="131072"/>
  <Errors>
    <Error errorCode="INVALID_REQUEST">You cannot specify both the at and frequency arguments to a current request</Error>
  </Errors>
</MTConnectError>
```

Real-Time Updates

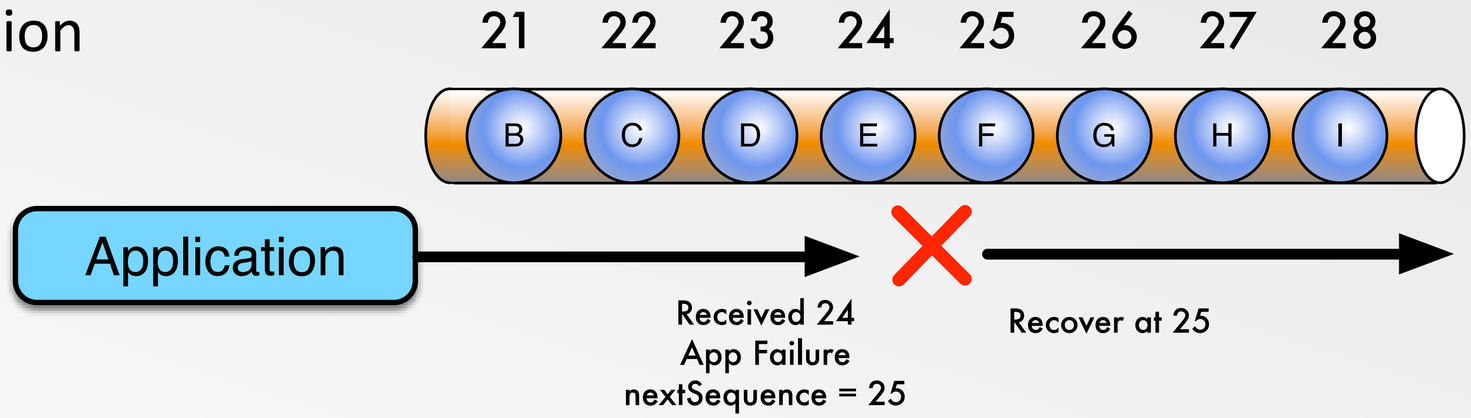
- Certain use cases require real-time notification
- This can be done using MTConnect and push based communication
- Method
  - Use **interval=0** : this means that the changes will be sent with no delay
  - Heartbeat still applies if no changes occur
  - Updates can be propagated in ~3ms on average PC and < ~10ms on ARM embedded
- **NOTE:** Use a path! You will want to limit the data to only the data items you are interested in
  - The agent has been implemented to make this efficient

?interval=0&path=//DataItem[@type="EXECUTION" or @type="CONTROLLER_MODE"]

High Availability

- In case of failure, the application can always continue where they left off
- Check the instance id has not change
- Request samples from that sequence number
- *Application Suggestion*:
  - Save the nextSequence and instanceId for every MTConnectStreams doc
    A. Same Instance Id: Continue from nextSequence. If error → B.
    B. Different Instance Id: Current → Sample …

- *REST Note*: Client is responsible for state…
  Agent does not keep client session
  - simplifies *Agent* implementation



Open Source Implementations

- C++ Agent is available on github: http://github.com/mtconnect/cppagent
- Implements all the protocol and XML generation
- There are open source implementation of stream processing in the following languages (available on http://github.com/mtconnect):
  - Ruby
  - Python
  - C
  - C#/.NET

- If you need, I can provide the following:
  - JavaScript (node.js)
  - Java/Scala using netty
- Other languages?

Assets

- Assets are primarily identified by their asset id (Key)
  - http://example.com/asset/eb33cdfc-0d72-11e5-a66e-28cfe91a82ef
- Multiple assets can be retrieved at the same time by separating them with ;
  - http://example.com/asset/hh1;cc;123;g5
- You can select a number of assets, returned newest to oldest
  - http://example.com/assets
- You can control the number of assets returned by specifying a count
  - http://example.com/assets?count=1000
- To get 1000 CuttingTool assets
  - http://example.com/assets?type=CuttingTool&count=1000

Asset Removal

- Assets are marked as removed, not actually removed
- Once marked they will be retrieved
  - If requested directly with their asset id
  - If the optional argument removed=true is added to the URL

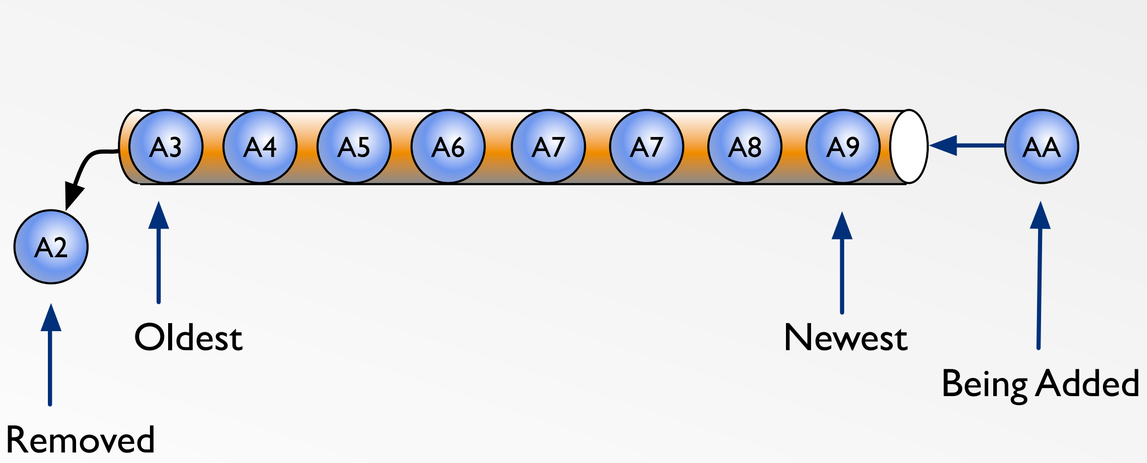
    http://example.com/assets?type=CuttingTool&removed=true

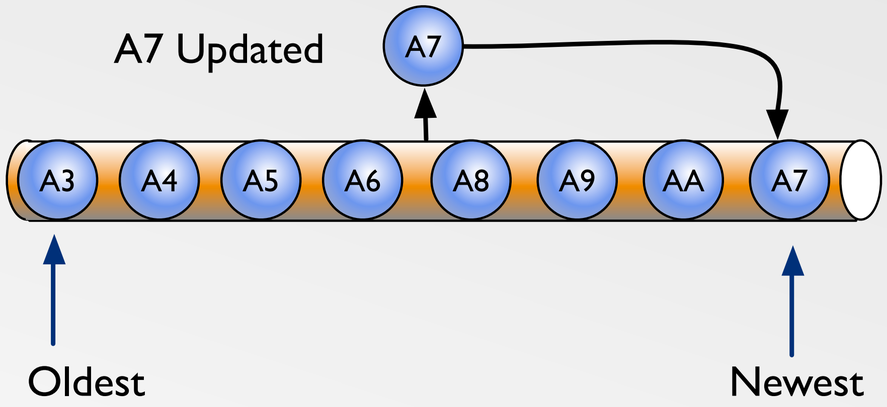
- There is an attribute in every asset that indicates if it is removed:

```
<CuttingTool serialNumber="1 " toolId="B732A08500HP"
             timestamp="2011-05-11T13:55:22"
             assetId="B732A08500HP.1" manufacturers="KMT"
             removed="true">
```

Asset Storage

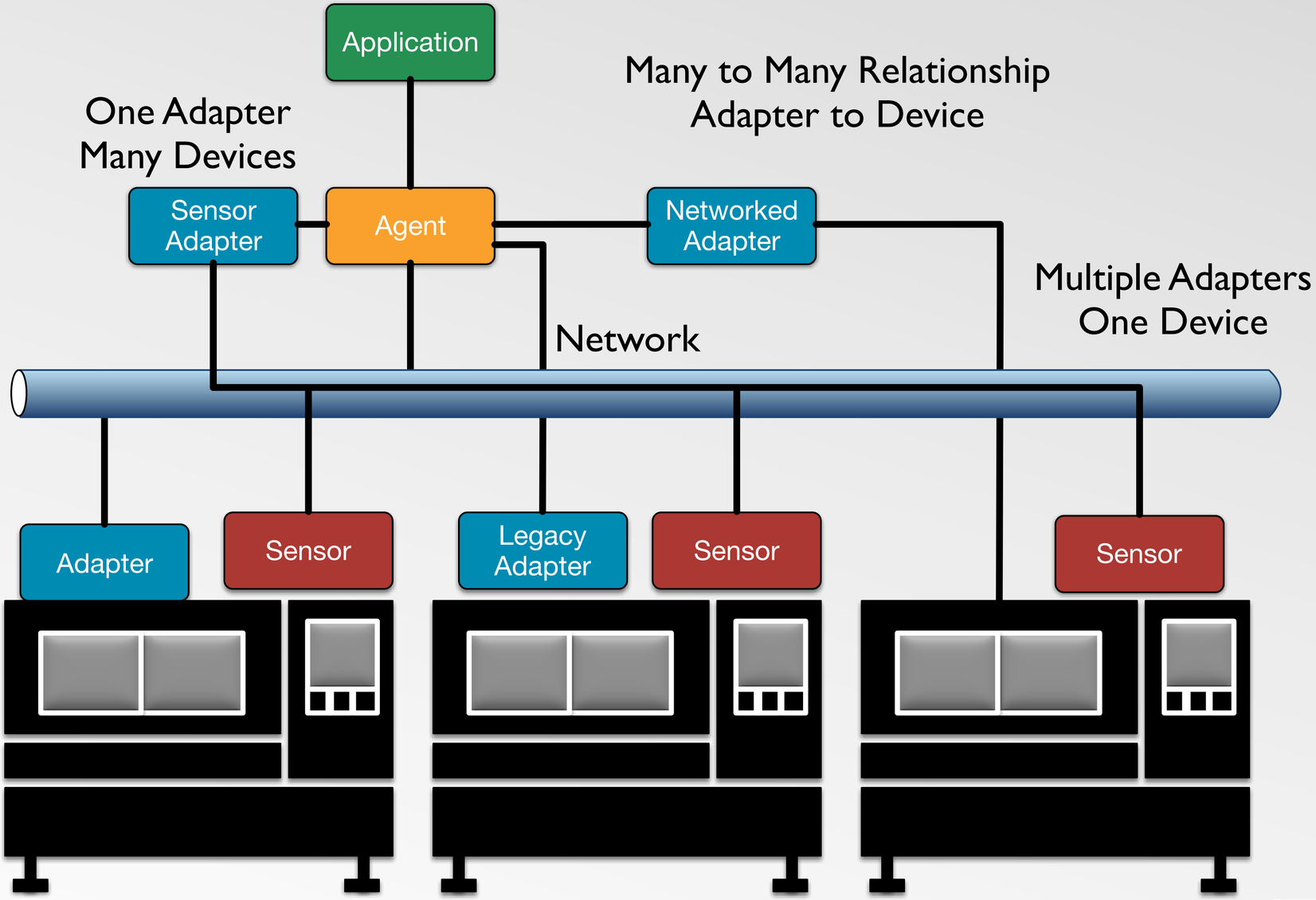
- Assets have limited resource constraints like events, samples, and conditions
- There is a maximum number of assets an Agent will store
- The oldest assets will be removed once there is no space for a new asset
- The standard does not require or prohibit persistent storage
- When assets are removed, they are moved to the beginning of the list
- When assets are marked as removed, they are not promoted to the first position





Adapters

Adapter Architecture



Adapter Communications

- Adapters are simple socket servers
- Agent connects to the adapter
- Adapter begins streaming data
- Heartbeats to detect Agent or Adapter failures
  - Agent sends “* PING”
  - Adapter responds with “* PONG *<Frequency in ms>*”
    - * PONG 10000
  - This will tell the agent to send a “* PING” every 10 seconds
- Data is pipe delimited starting with a timestamp in UTC (GMT)
  - Relative time supported as well for clock skew corrections
- Events and samples are key/value matching data item id, name, or source in probe
- Conditions, Messages, and Timeseries have more data

Adapter Protocol

- Documented on: https://github.com/mtconnect/cppagent
- Samples and Events

```
<timestamp>|<data_item_name>|name 1|value 1|name 2|value 2|name 3|value 3|…
2009-06-15T00:00:00.000000|power|ON|execution|ACTIVE|line|412|Xact|-1.1761875153|Yact|1766618937
```

- Conditions

```
<timestamp>|<data_item_name>|<level>|<native_code>|<native_severity>|<qualifier>|<message>
2014-09-29T23:59:33.460470Z|htemp|WARNING|HTEMP|1|HIGH|Oil Temperature High
```

- Messages

```
<timestamp>|<data_item_name>|<native_code>|<message>
2014-09-29T23:59:33.460470Z|message|CHG_INSRT|Change Inserts
```

- Time Series

```
<timestamp>|<data_item_name>|<count>|<sample rate>|<message>
2014-09-29T23:59:33.460470Z|current|10|100|1 2 3 4 5 6 7 8 9 10
```

Libraries Available in Multiple Languages

- Open Source libraries and tutorials available on github
  - C++
  - C#/.NET
  - Ruby
  - Python
- Simple protocol, can be written in any language
- Other implementations out there for many platforms

Extending the Schema

- New Components
- New Data Items
- New Assets

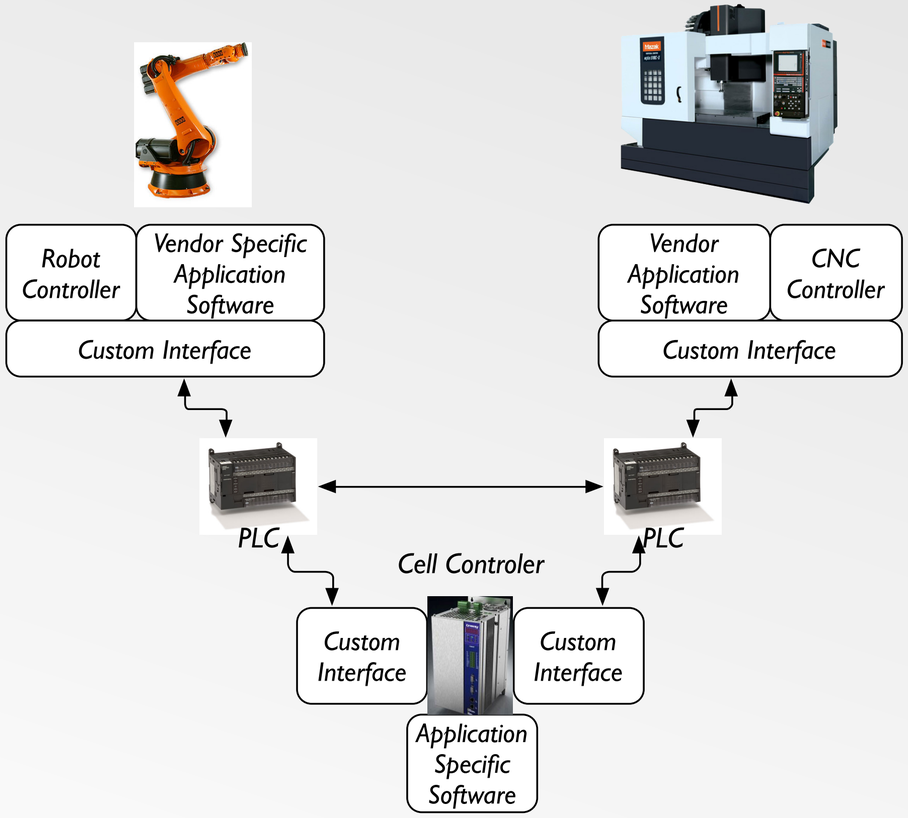
M2M

Read-Only Interfaces – Observation

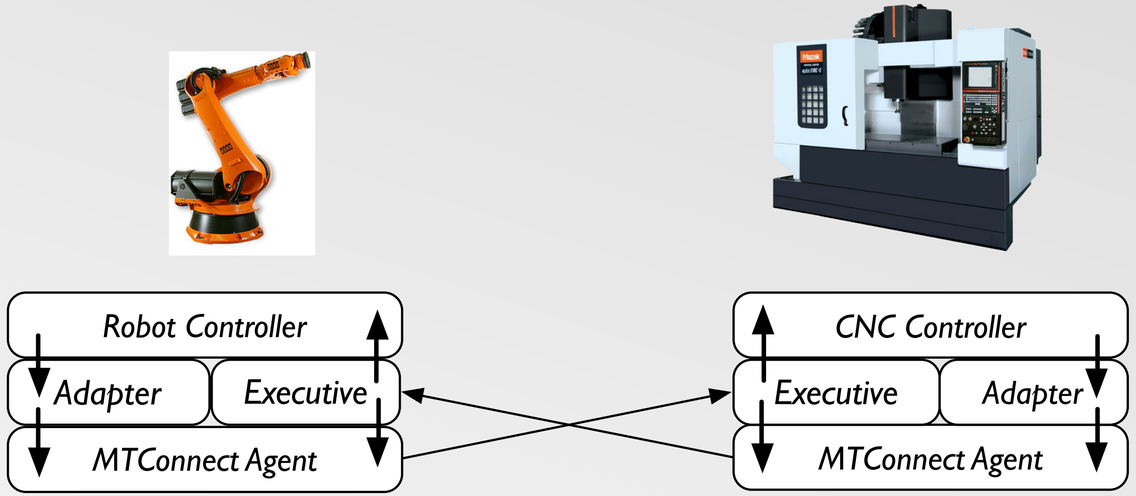
Interfaces

It's about the $$$

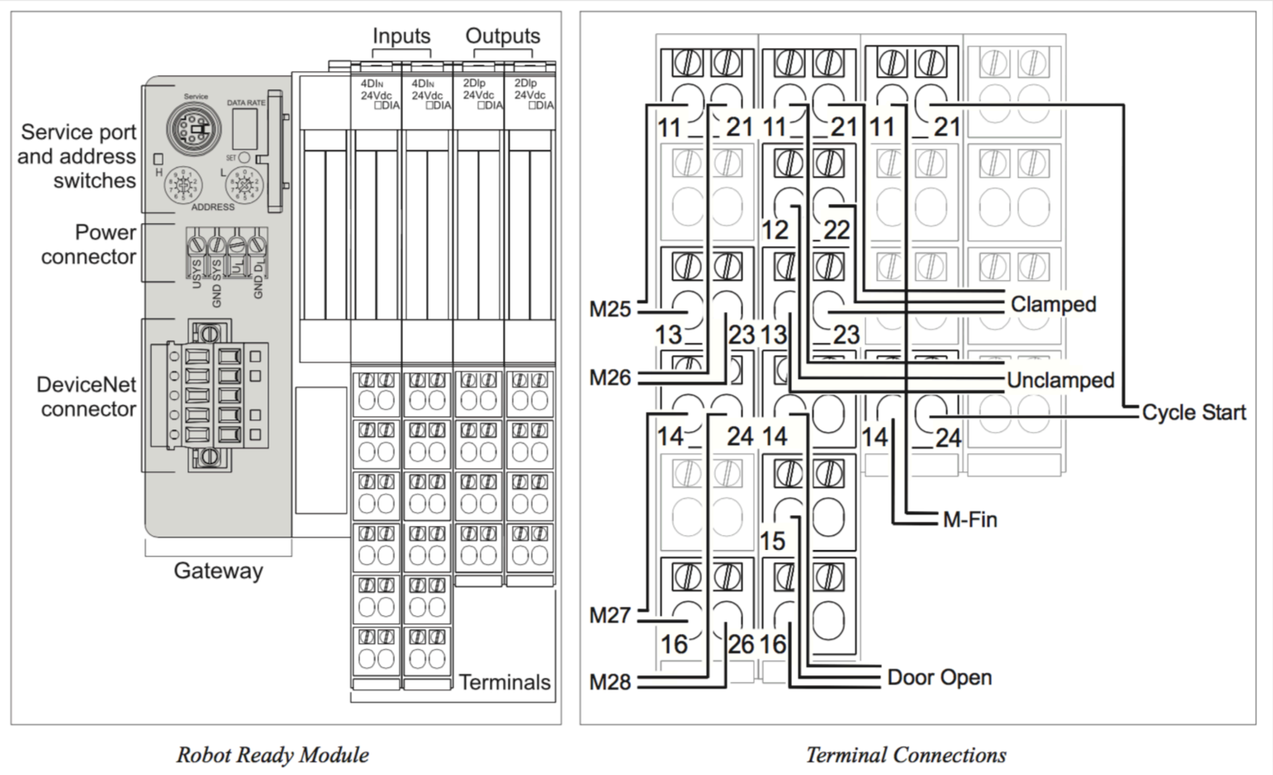
Present - $$$$

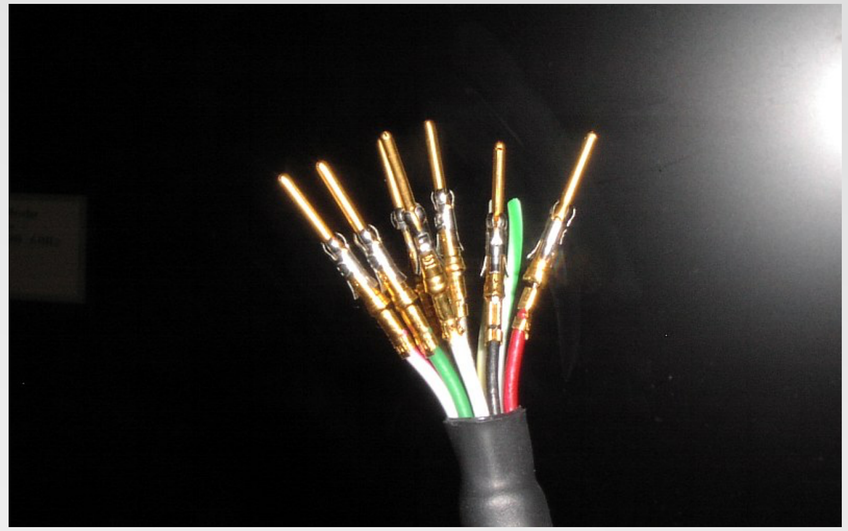


MTConnect - $



Wires

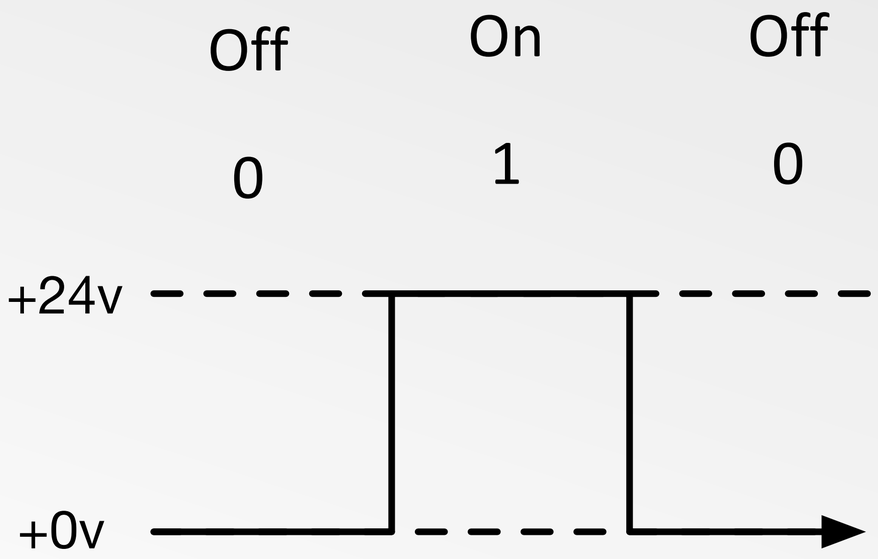




Haas Robot Ready Option - 2014

Communications

Present



MTConnect

```
 <PartArchetype assetId="X11255678"
timestamp="2004-10-05T12:00:00Z" revisionId="7">
  …
<ProcessStep stepId="40”>
 <Description>FINISH FWD</Description>
   <Targets>
    <TargetDevice>SL-75</TargetDevice>
   </Targets>
   <ControlPrograms>
    <ControlProgram name="Finish Fwd" programName="152440"
revisionId="2"/>
   </ControlPrograms>
     <TargetExecutionTime>1036</TargetExecutionTime>
     <TargetSetupTime>600</TargetSetupTime>
     <CuttingTools>
      <CuttingToolSetup sequence="1">
       <CuttingToolAssetId>CRGPL-203V-RPGN-3V</
CuttingToolAssetId>
      </CuttingToolSetup>
      <CuttingToolSetup sequence="2">
        <CuttingToolAssetId>NSL203D-NR3047L</CuttingToolAssetId>
      </CuttingToolSetup>
```

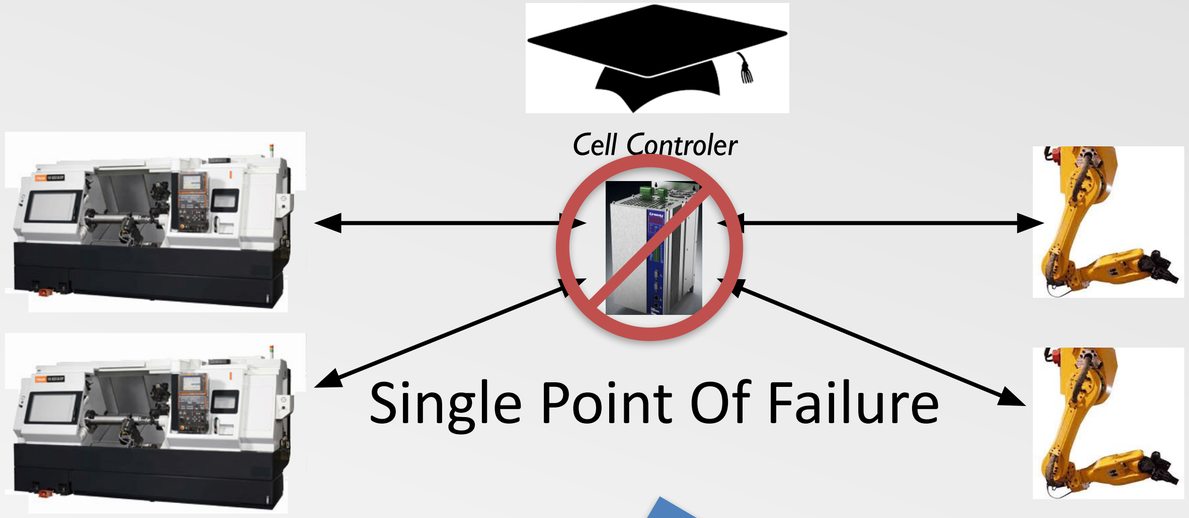
Distributed Intelligence

Present

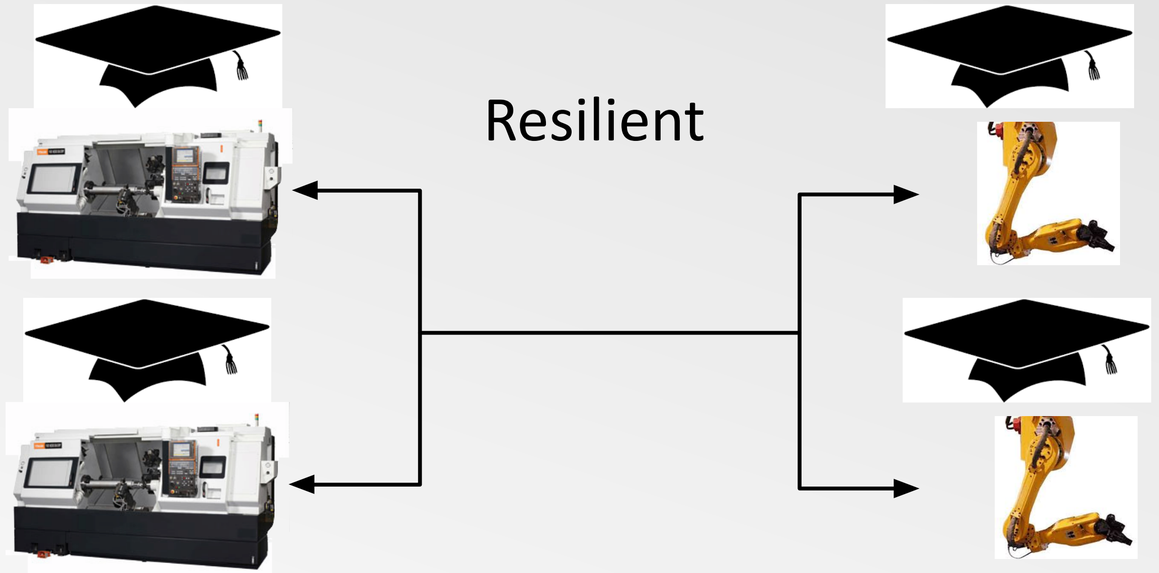
Cell Controler

MTConnect

Increased Complexity and Agility



- More resilient to failure – self healing
  - Can operate when single components fail
  - Can dynamically schedule and work around problems
- No single point of failure
- Lower cost
- Greater flexibility



Current Standardized Interfaces

Chuck

Door

Material Handler

Bar Feeder

More?

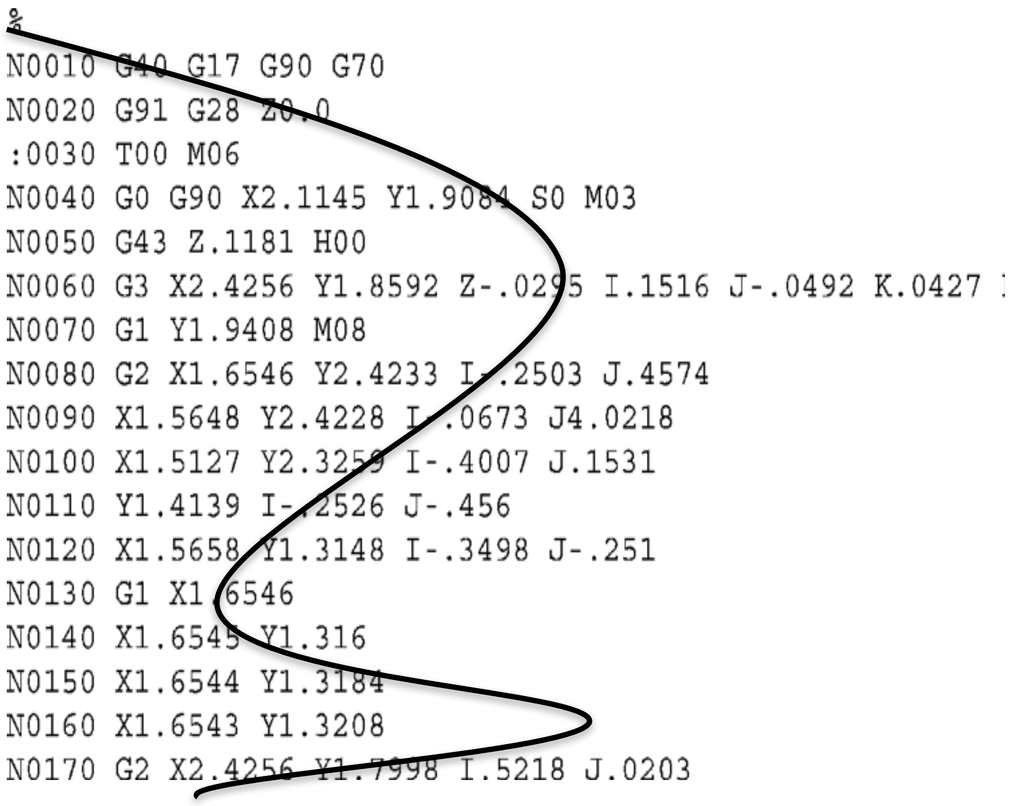
Research

UC Berkeley –LMAS

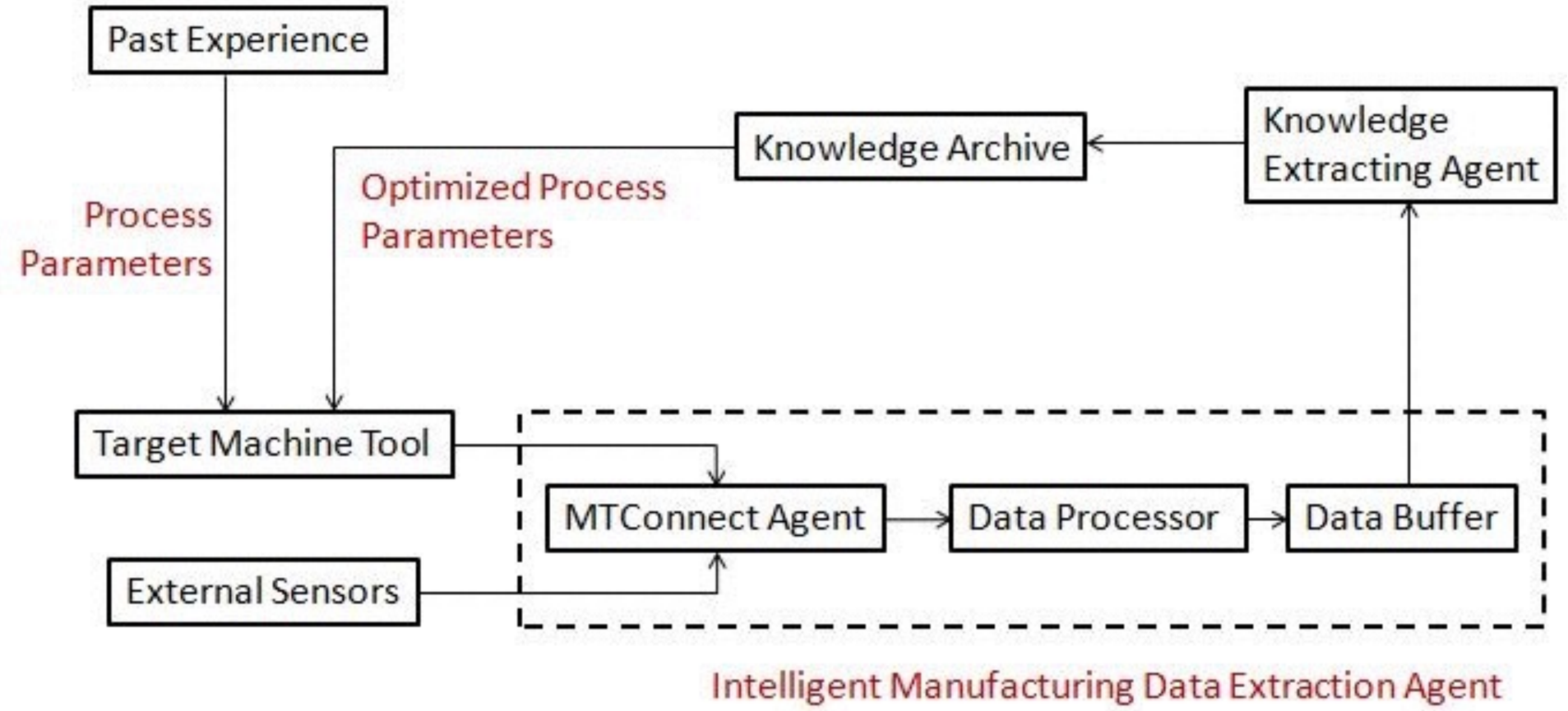
Raunak Bhinge, Dave Dornfeld

Concept: Real-Time Data with NC Code

- NC Code governs all the characteristics of a process
- Fusion of Real-Time Data with Controller Data
- Diagnostics / Prediction based on machine learning techniques

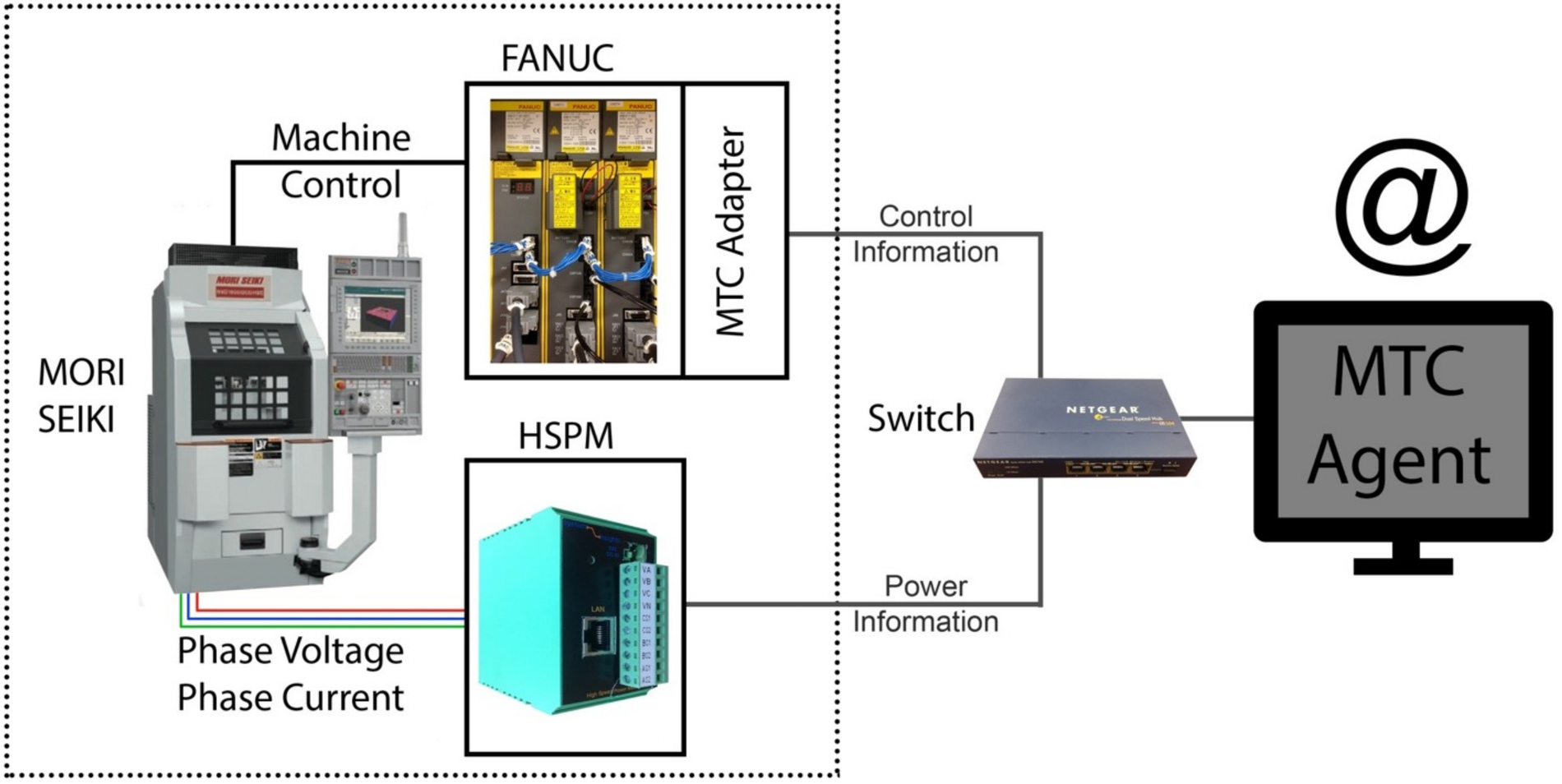


Data-Driven Manufacturing



Bhinge, Raunak, et al. "An intelligent machine monitoring system for energy prediction using a Gaussian Process regression." *Big Data (Big Data), 2014 IEEE International Conference on*. IEEE, 2014.

Machine Tool Platform



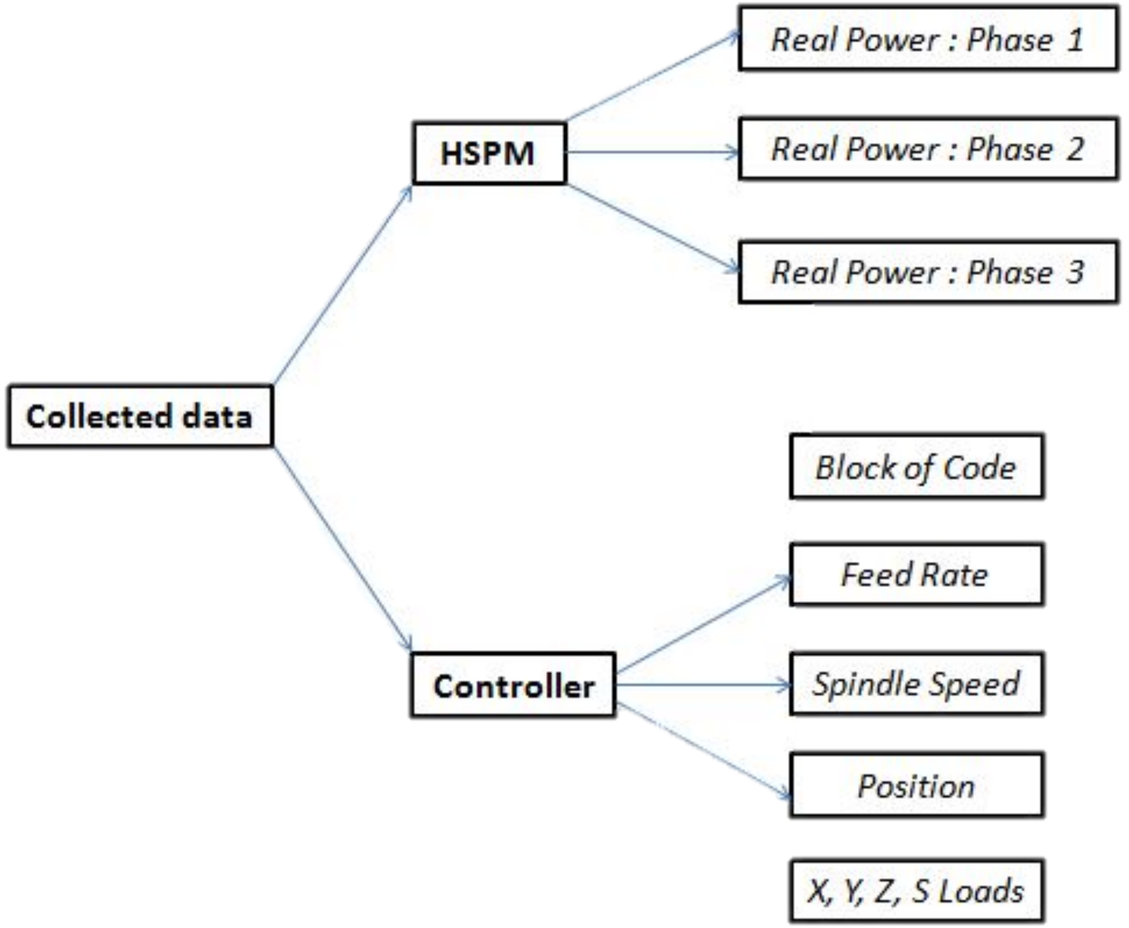
T. Bänziger. *Analyzing the Relationship between the Power Demand of a CNC Machine Tool and the Surface Roughness Imparted on a Machined Part*. MS Thesis, ETH Zurich, 2014.

Data collection and processing

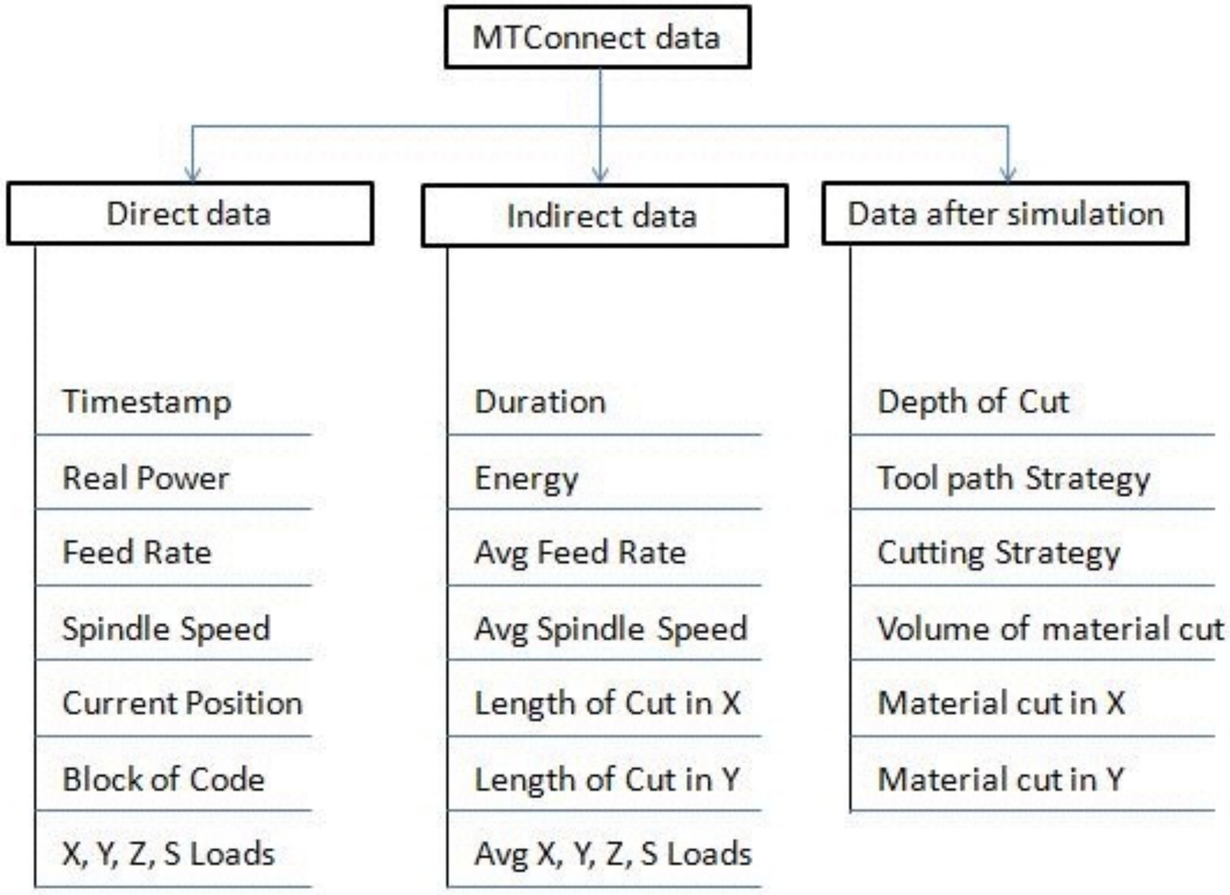
1) MTConnect

2) Data Condensation

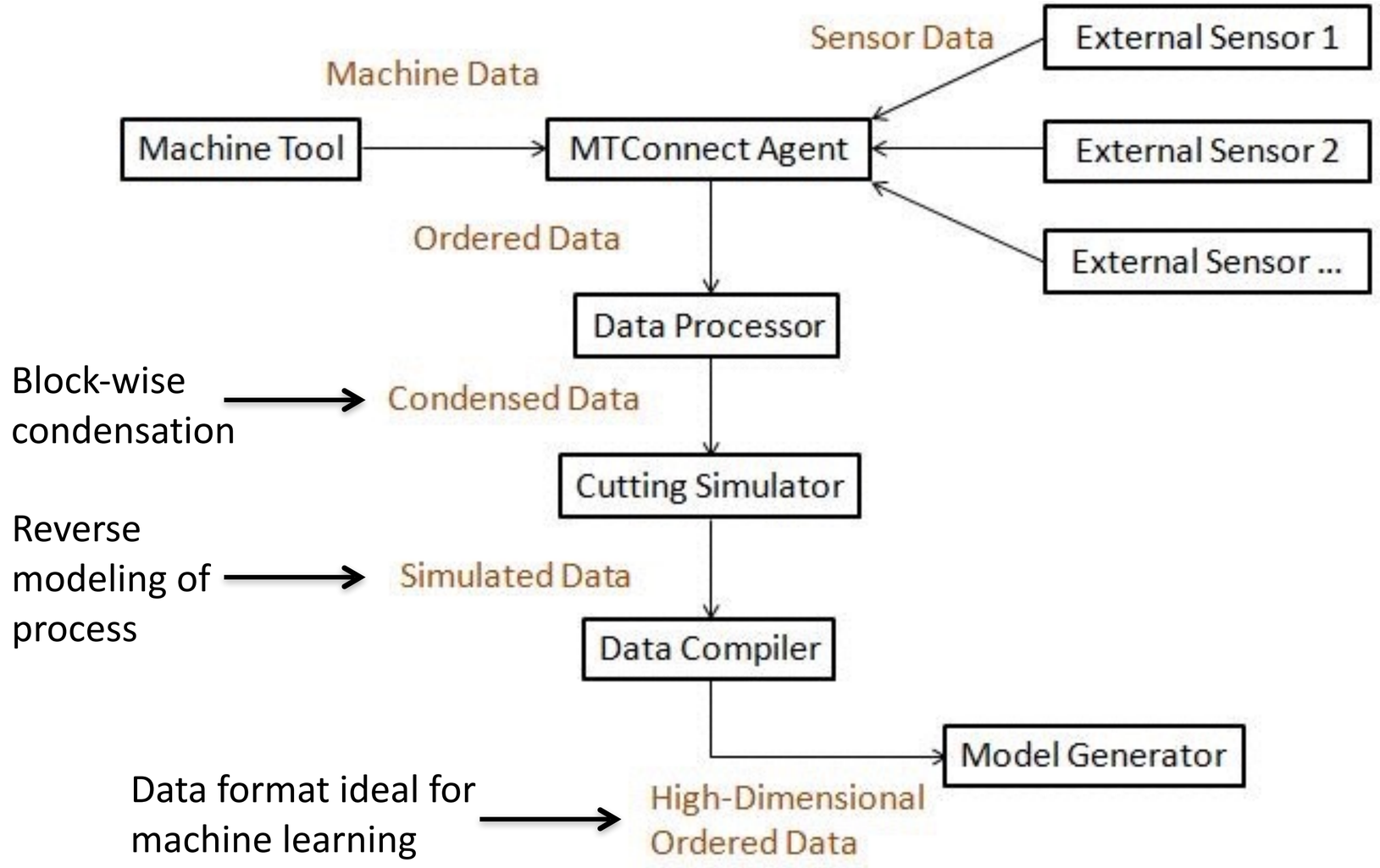
3) Cutting Simulation



Data collection and processing



Data Processing Framework

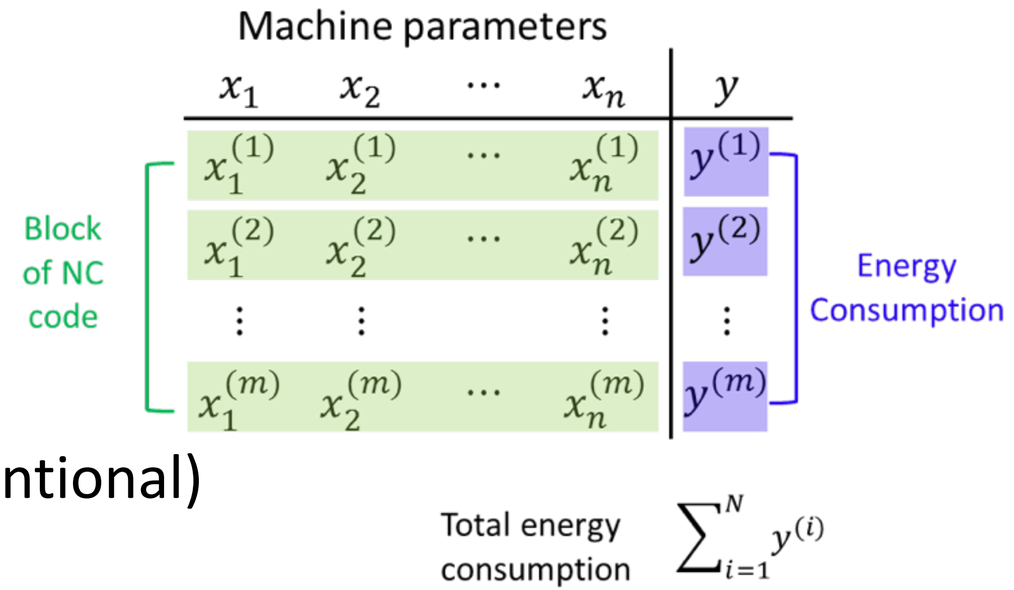


Adapted from Bhinge, Raunak, et al. "An intelligent machine monitoring system for energy prediction using a Gaussian Process regression." *Big Data (Big Data), 2014 IEEE International Conference on*. IEEE, 2014.

Features for Machine Learning

Machining parameters used:

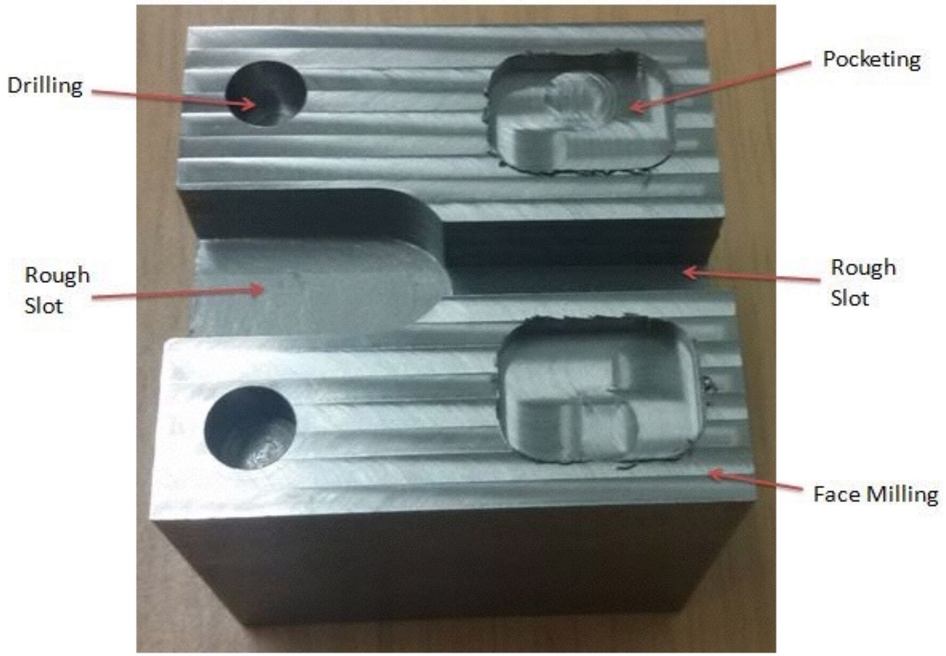
- Spindle Speed
- Feed Rate
- Direction of Cut (X, Y, Z, X-Y)
- Cutting Strategy (Climb, Conventional)
- Length of Cut
- Ratio of length of cut to length of material removed



- Energy / Length of cut -> Chosen as the output feature for energy prediction in order to characterize any spatial cut

Case Study 1 : Uncertainty Analysis

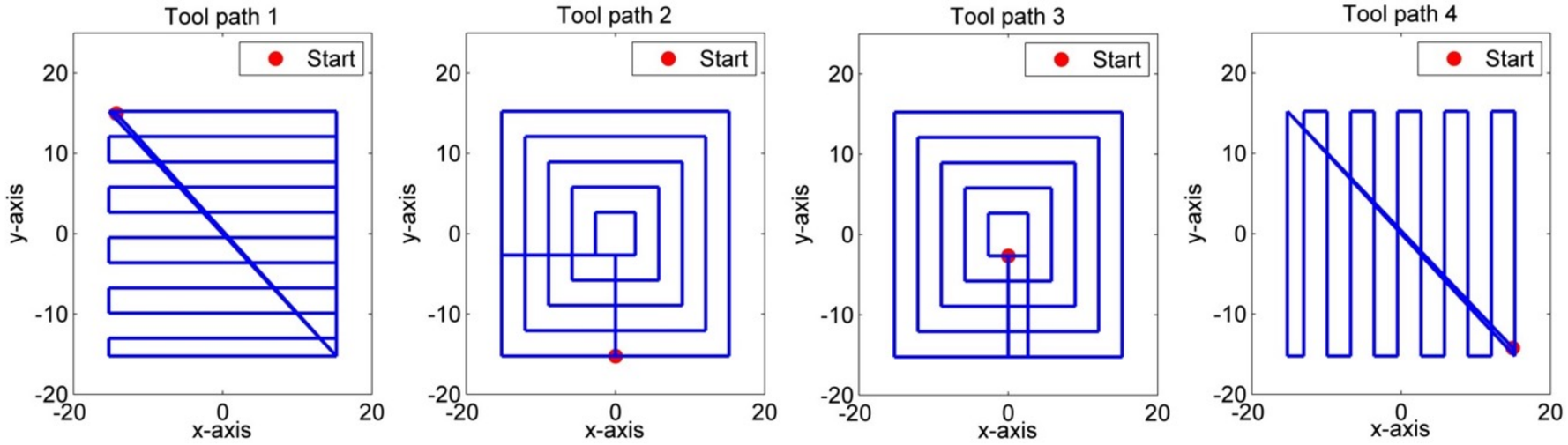
Part	Spindle Speed (RPM)
Training Set (Set of 18 parts)	*[1500; 3000; 4500]*
Test Part 1	*[1500; 3000; 4500]*
Test Part 2	*[1700; 2800; 4300]*
Test Part 3	*[2125; 2400; 3750]*



Objective : To investigate and verify the estimated uncertainty in energy prediction for the unexplored parameter space

Park, J., Law, K., Bhinge, R., Biswas, N., Srinivasan, A., Dornfeld, D., Helu, M., and Rachuri, S., "A generalized data-driven energy prediction model with uncertainty for a milling machine tool using Gaussian Process," Proc. ASME 2015 International Manufacturing Science and Engineering Conference (MSEC2015), July, 2015, Charlotte, North Carolina.

Case Study 2 : Toolpath selection



Objective : To predict the most energy-efficient toolpath from a set of 4 possible toolpaths commonly used for pocketing

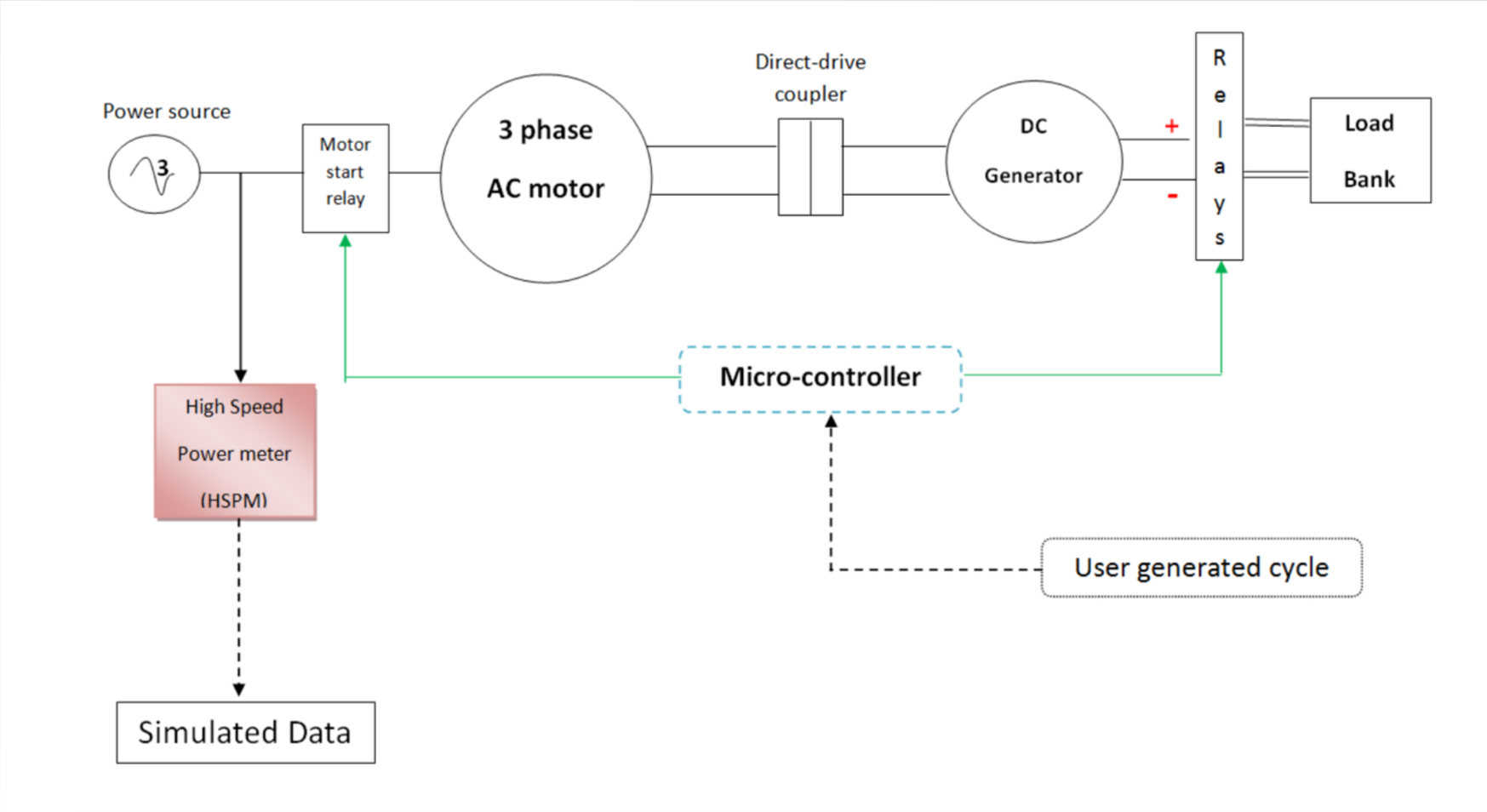
IIT Madras

Harsh Dash, Rajat Tiwari, Athulan Vijayaraghavan

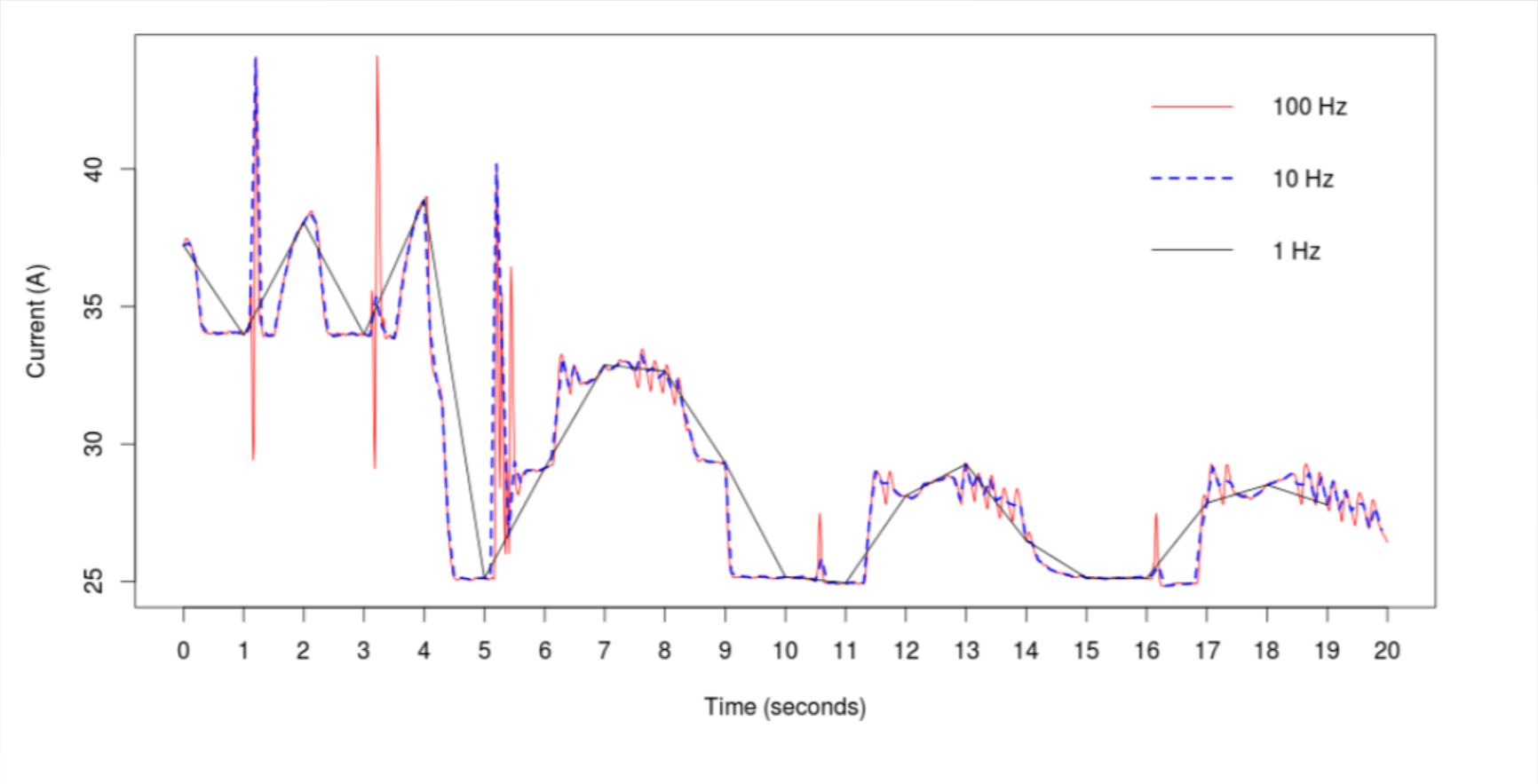
Energy Equivalent Machine Tool Simulator

- Energy equivalent simulator with basic electrical components, which can replicate the current consumption behavior of a single spindle machine tool
- Successfully simulated several machine cycles involving various operations in a single spindle machine tool. This research will provide a novel and inexpensive way of simulating machine tools
- A high speed power meter (HSPM) device, which records the current drawn by the machine at a high frequency (100 Hz), can be used as a way to monitor different performance related parameters like machine cycle, down time, cycle time, tool life, etc.

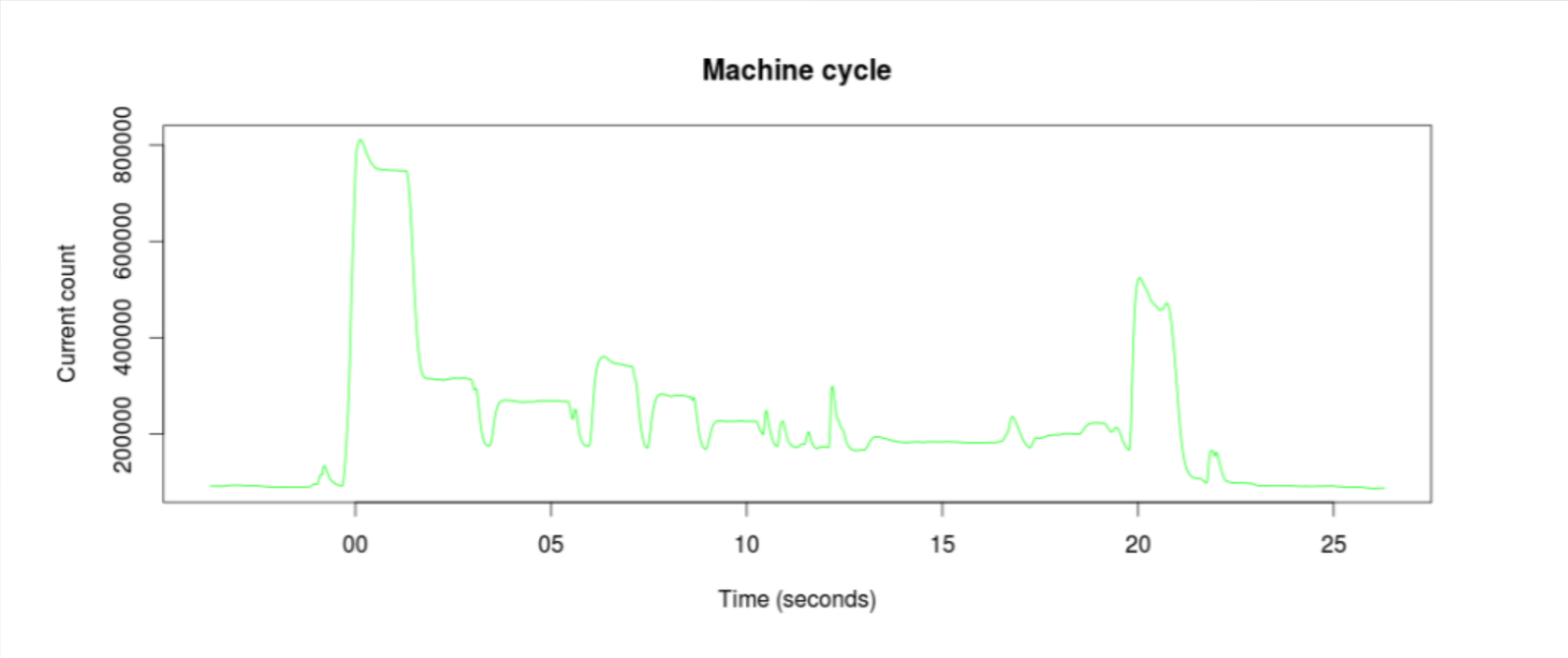
Hardware and Sensors



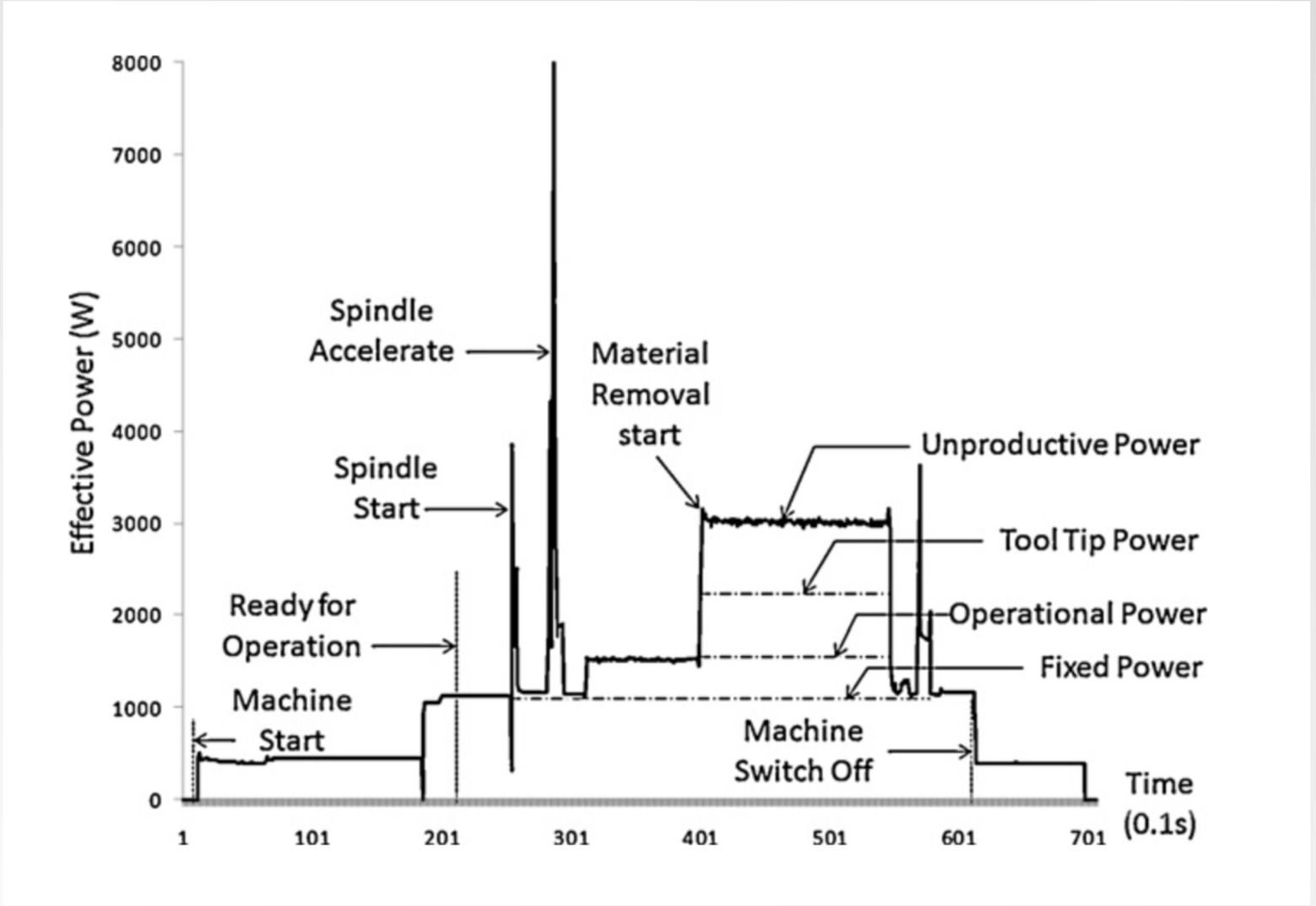
Sampling Frequency Comparison



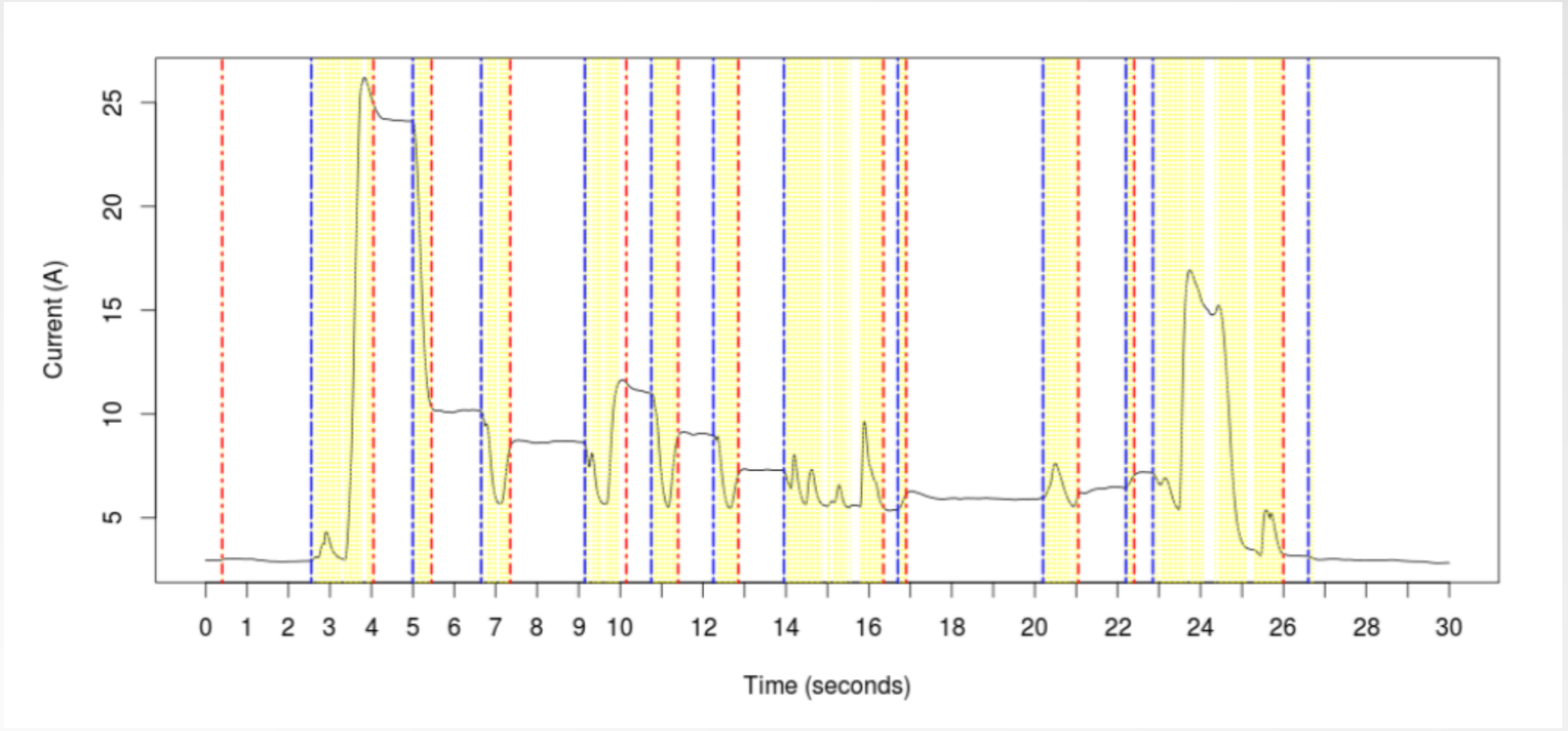
Recurring Machine Cycle



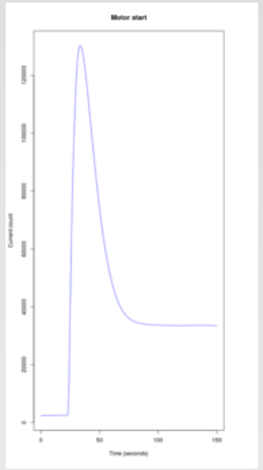
Energy consumption by a machine tool



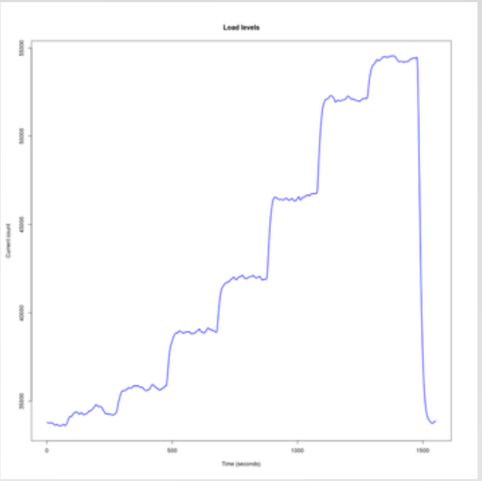
Feature Finding Algorithm – Identifying Motifs

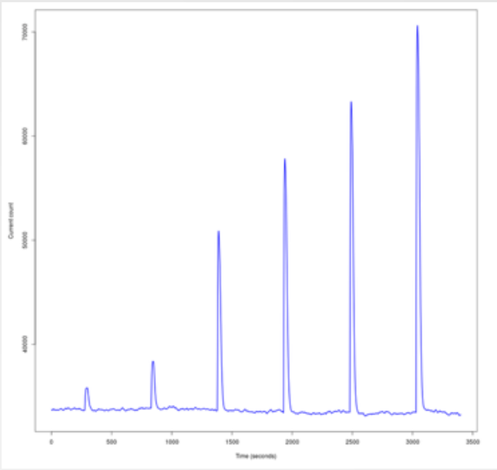


Motor Start



Load Levels

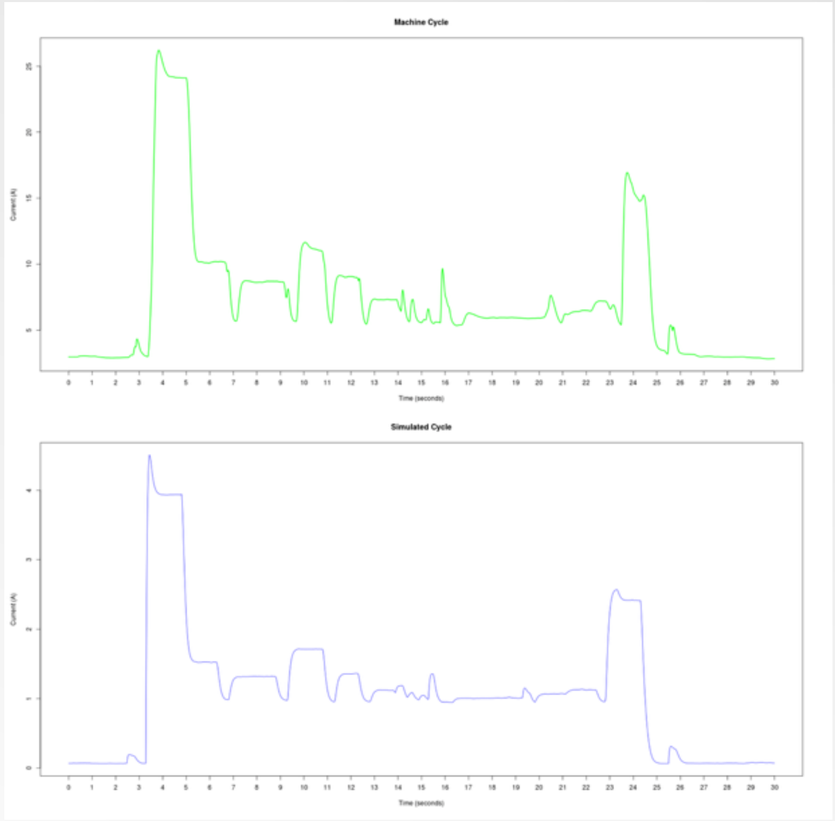




Rapids

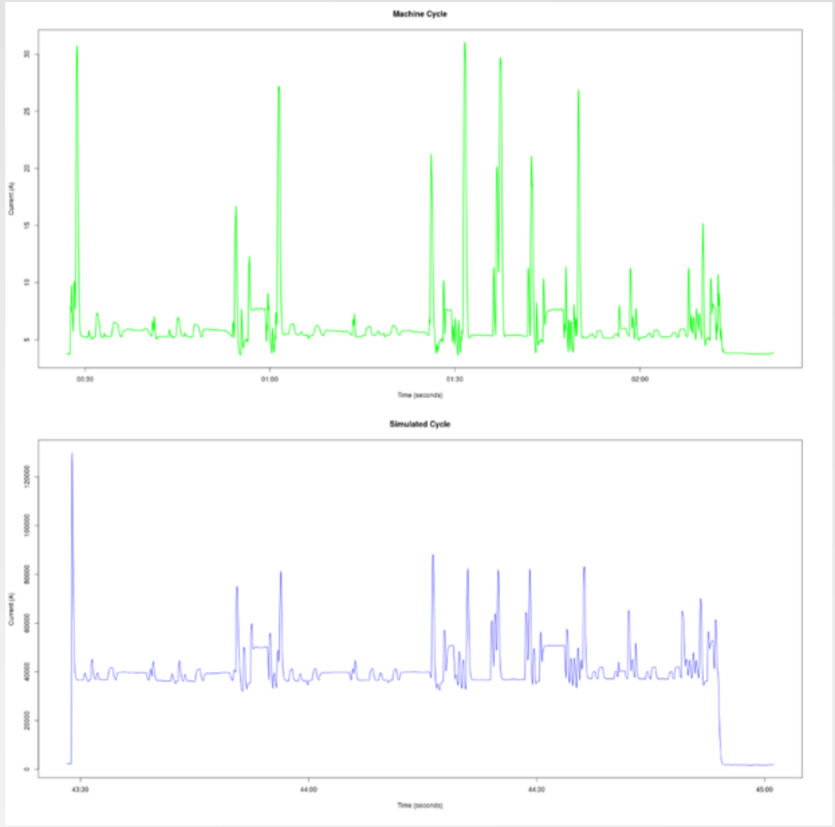
Comparison of Cycles

Actual



Simulated

Actual



Simulated

The Pearson correlation coefficient was found to be 0.82, where a coefficient of 1.0 indicates a perfect match.

IIT 3D Printing

Rohan, Roshan, and Athulan Vijayaraghavan

Project Goals and Approach

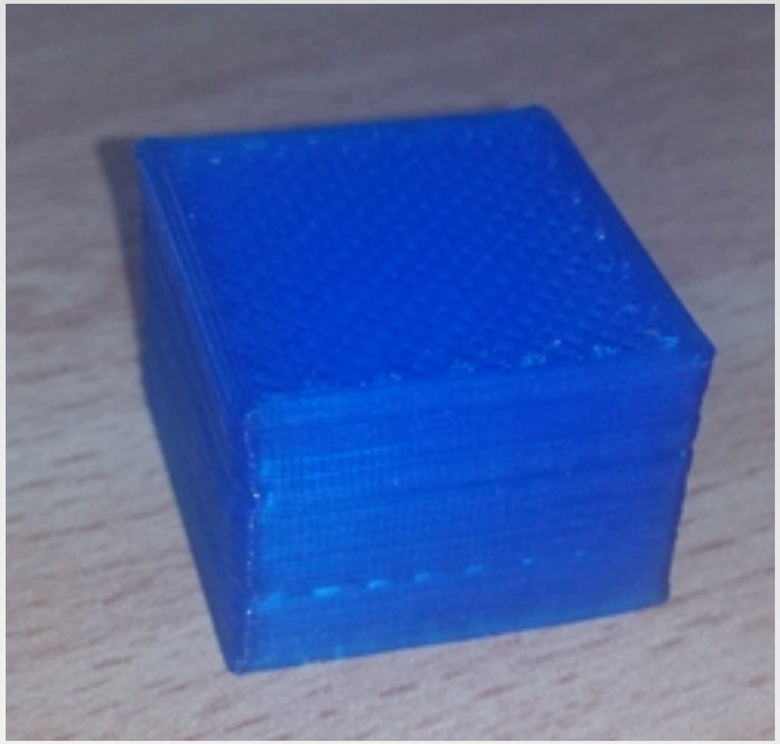
- Correlating process parameters and final printed object
- Building a model which would identify this relation
- Focus: Shape and Geometry of 3D printed product
- Two parameters to define quality:
  - Strength
  - Dimensional Accuracy
- Process for correlation of quality parameters to the process parameters
  - Selection of Process Parameters
  - Design of Experiment for Strength and Dimensional accuracy tests
  - Printing Samples
  - Measurement techniques
  - Building Models to identify relation between process parameters and Quality parameters

Controls and Variables

- Chose 7 variables for dimensional accuracy and 7 variables for strength
- These parameters were chosen with following things in mind
  - Past literature reviews related to FDM and non-FDM additive manufacturing process suggested influence of temperature, speed, infill density and infill angle
  - All the parameters chosen for strength testing process are easily accessible through Slic3r or any other 3D printing software interface
  - Advanced parameters in Slic3r are based on finer tuning of these 7 parameters and hence were grouped together and controlled relative to these basic parameters
  - Parameters chosen for dimensional accuracy tests are a mix of firmware parameters and process parameters. Marlin firmware was modified to observe dimensional accuracy
  - Ambient Temperature, PID values, Slicing Software, Filament type were all kept constant while conducting the whole process

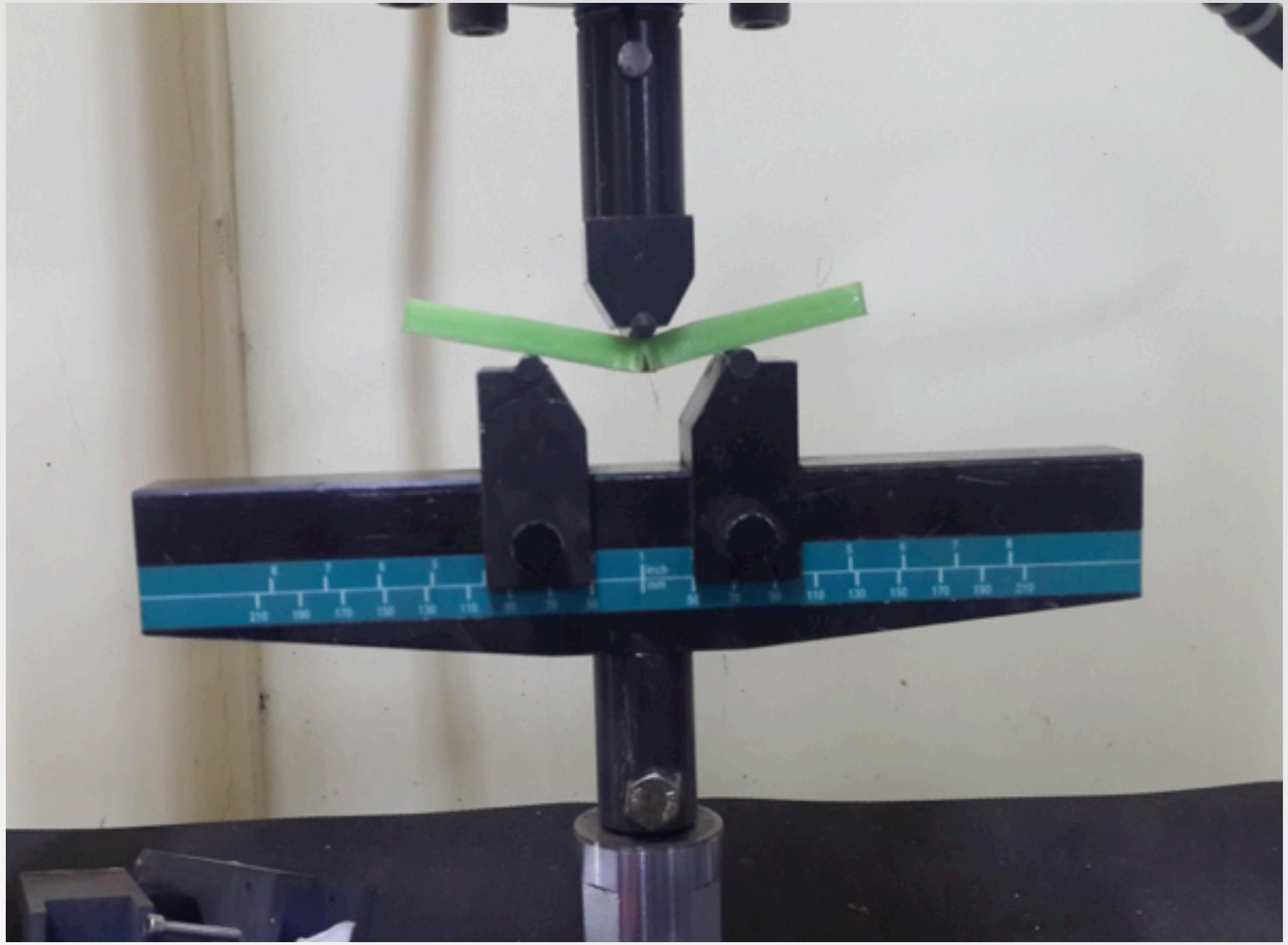
Focus Areas

- Strength
  - For conducting the experiment, we used a L16 orthogonal array to determine the combination of input variables for the printed test objects. Though we are concentrating only on 7 variables, we decided to use L16 instead of L8 as we planned on studying the variable interactions at a later stage.
- Quality
  - For the dimensional accuracy, cubes of 25 mm x 25 mm x 10 mm size were printed.
  - A vernier caliper was used with 10 micron accuracy for measurements. Vernier caliper was chosen as errors in the order of 0.1 mm were observed



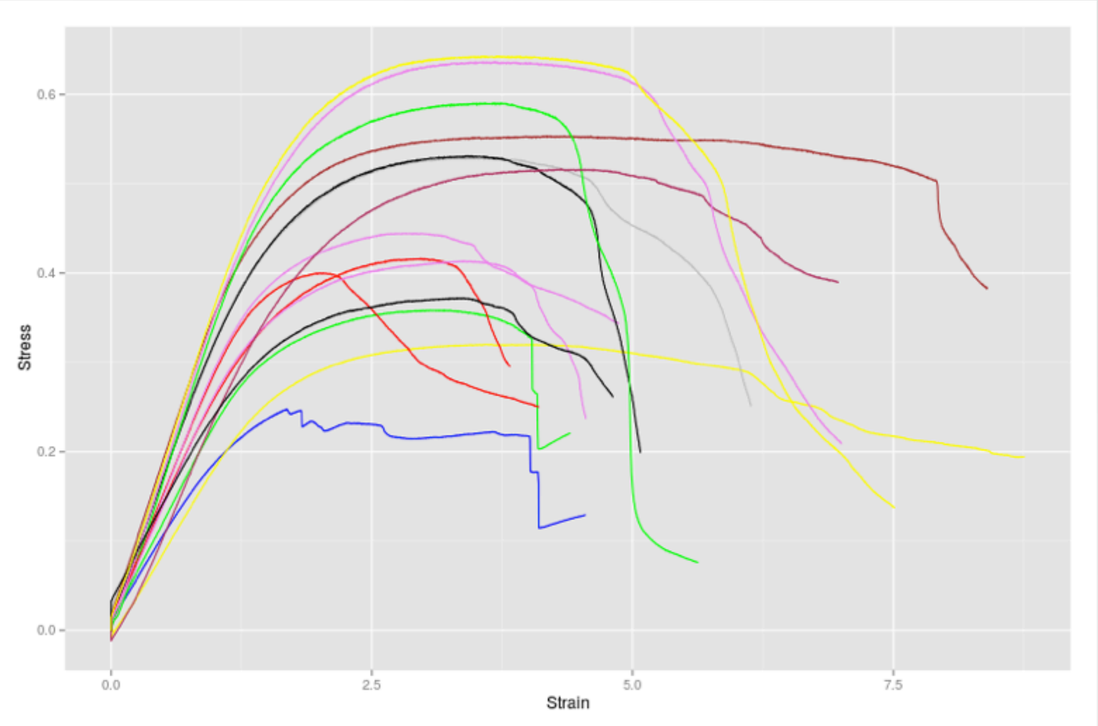
Strength – Ultimate Tensile Strength (UTS)

- The 3-point bend test output data had load and displacement data. Along with the dimensional data of the printed objects, other variables such as stress, strain, flexural modulus and Ultimate tensile strength were determined



Strength Results

- The important factors in terms of strength are :
  - Extruder temperature – important for modulus determination but not so much for UTS determination
  - Infill pattern – important factor for determining both UTS and modulus
  - Slice height – strong predictor for UTS but had little significance when it came to determining modulus
  - Infill density – did not have much significance in modulus determination but had an important role in UTS determination



Accuracy Results

- For dimensional accuracy, the variables didn't have a very strong significance in the final prediction.
- Effect of the the variables was expected to be the same for X and Y axis. However the models for X and Y axis had opposing importance levels for the variables.
- For the X and Y axis models, the intercept had a very high significance value indicating the lack of variability in the dimensional accuracy. Even in the Z axis model, the intercept has the highest significance.

Free Tools and Tech

Open Source & Demo Apps and Libraries

- Available on github.com/mtconnect
  - MTConnect Reference Agent cppagent
  - Adapter Lab in C# using the dot_net_sdk
  - Application Lab in C#
  - Android Client for MTConnect
  - Google Glass Demo
  - Unity 3D Demo
  - MTConnect Demo app in Ruby on Rails (demo.mtconnect.org)
  - ROS integration with MTConnect using ROS/I for machine to machine
  - Stream parsing in C#, C (C++), Python, Ruby, & Objective-C

Great Resources

- R Studio – IDE for data analysis
  - https://vimeo.com/97166163
- Microsoft's machine learning platform
  - Free to get started and has GUI – simple to use
  - http://azure.microsoft.com/en-us/documentation/videos/overview-of-ml/
- Amazon Machine Learning
  - Batch and Real-Time processing – integrates with other AWS services S3, Redshift, EC2, …
  - Good tutorials online – not as simple as MS to get started
  - Cheap to get started
  - https://youtu.be/PAHU8tPA7xs
- Some R Scripts for working with MTConnect data
  - https://github.com/systeminsights/vrtk

Getting Started

- Collect data from MTConnect
- Easiest way: Stream XML data to a file and turn it into CSV
  - Contact us for more tools – we are putting open source tech together for you
  - Contact us for data sets – we have lots of data
- Once you have CSV files, do the following
  - Import data sets into R
  - Play with data…
- MTConnect will provide consistent data across multiple devices
  - Focus on analytics and value

- System Insights will be creating webinars for the following:
  - MTConnect data collection and management
  - Data analytics using R Studio – what we use for predictive analytics – and providing instruction