Industrial IoT:

Architecture Framework
Use Cases

Artur Borycki
Teradata Labs
IoT represents more than just "things":

It must represent systems (and systems of systems)
The Internet of Things: It’s About The Sensor Data, Stupid!

1. Sensors sometimes lie

2. Even when they are not lying, sensors may not tell the whole truth

3. Extracting useful signal requires “multi-genre” Analytics – and additional data

4. Sensors typically don’t measure the quantity of interest directly

5. By itself, sensor data is frequently not actionable

Don’t assume that sensor data is accurate, complete and consistent; do apply Information Management best-practices.

Capture raw sensor data wherever possible; otherwise, understand how sensor data has been summarised.

Plan to support a wide variety of Analytics – path/ pattern/ graph/ time-series/ text - just to prepare an ADS from time-series data.

Some model scoring may take place on the smart device itself; build models centrally, but be prepared to deploy them widely.

Integrate your Data Lake and Data Warehouse environments, so that observation and transaction data can be joined.
Real value requires a system (and system of systems) view:

INVESTMENT FOCUS

VALUE FOCUS

Systems Science or Theory

@naotar
IoT: "Enables lots of metrics to be passed around to add value":

Analytics at scale
IoT: "Enables lots of metrics to be passed around to add value".
IIoT: "Raw data is too siloes to allow analytics at scale"
Teradata IIoT Architecture:
Let's start talking details:

**Data Governance Framework**
- Provenance Service
- Authorization Service
- Orchestration Service
- Data Storage
  - HDFS
  - S3
  - NoSQL
  - In-memory
  - Teradata

**Access Framework**
- Dashboard Service
- Reporting Service
- Bulk Push Service
- Plugins:
  - Spark
  - sqoop
  - NoSQL
  - Solr
  - Teradata
  - Aster
  - RestAPI

**Processing Framework**
- Config Service
- Pipeline Registry
- Dependency Management
- Scheduling Service
- Plugins:
  - Aster
  - Map Reduce
  - MQ
  - SQL (Hadoop)
  - Stream
  - Spark
  - Storm
  - Teradata Table Operators

**Advanced Analytics & ML**
- Config Service
- Advanced Analytics & ML
  - Plugins:
    - Spark
    - sqoop
    - NoSQL
    - Solr
    - Teradata
    - Aster

**STEM**
- Plugins:
  - Python
  - R
  - Fortran

**Collection Framework**
- Config Service
- Validation Module
- Metadata Module
- Packaging Module
- Upload Service
- Bulk Collection Service
- Stream Collection Service
- Plugins:
  - fs
  - SAN
  - NFS
  - S/FTP
  - scp
  - S3
  - JMS
  - *MQ
  - Kafka
  - Listener

**Orchestration Service**
- HDFS
- S3
- NoSQL
- In-memory
- Teradata
Teradata IIoT:
Use cases
What the hell !!!???
Understanding the challenge

Business Challenge
- Highly skilled engineers spending too much time loading / manipulating / analyzing data
- Unable to look at all the data – need to split up into smaller sub-sets

Technology Challenge
- Data Storage (cost effective)
- Scalable Analytical Engine
- Visualization / Front End

Analytical Challenge
- Multi-Petabyte Data Volume
- Million+ Variables
- Constantly Changing Structure

Process to identify Key Variables so Action can be Taken

Product Complexity
Data Volumes
Number of Variable

Time to Insight
Failure
IT Cost & Complexity

Business
Technology
Analytical
Action
Insight every day / hr / sec without any code changes

1. Take Sensor data that is highly variable and very wide
2. Convert to JSON format for flexible schema and load into Teradata
3. Un-pivot data from columns to row in Teradata RDBMS
4. Use standard Teradata SQL to access “wide” sensor data and execute applied mathematics at scale

Real optimization comes from changing the right processes and inputs – not from changing the data structures or code

@naotar
UNPIVOT

```sql
SELECT *
FROM SCRIPT(
ON(select id, CAST(json.j1 AS CLOB ) FROM mfg.jtable)
SCRIPT_COMMAND('./mfg/unpivots.py')
RETURNST( 'oid VARCHAR(32), ckey VARCHAR(32), cvalue VARCHAR(256) ')
) AS d
ORDER BY 1,2;
```

```python
#!/usr/bin/python
import sys
import json
import codecs
#-- coding: utf-8 --
UTF8Writer = codecs.getwriter('utf8')
sys.stdout = UTF8Writer(sys.stdout)

for line in sys.stdin:
    line, decode = 'ascii', 'ignore'
    line = line.strip()
    fields = line.split('t');
    id = fields(0)
    if len(jsondata) > 16000000:
        raise Exception('document too large')
    parsedjson = json.loads(jsondata)
    for key, values in element.items():
        print u''\%s\%s\%s (id, key, values)
```

<table>
<thead>
<tr>
<th>old</th>
<th>ckey</th>
<th>cvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>CategoryData1</td>
<td>CS3_6057</td>
<td></td>
</tr>
<tr>
<td>CategoryData10</td>
<td>5xn_2527</td>
<td></td>
</tr>
<tr>
<td>CategoryData100</td>
<td>QVR_12617</td>
<td></td>
</tr>
<tr>
<td>CategoryData1001</td>
<td>DIB_6300</td>
<td></td>
</tr>
<tr>
<td>CategoryData1004</td>
<td>WFC_14981</td>
<td></td>
</tr>
<tr>
<td>CategoryData1008</td>
<td>034_49</td>
<td></td>
</tr>
<tr>
<td>CategoryData1007</td>
<td>DSP_6543</td>
<td></td>
</tr>
<tr>
<td>CategoryData101</td>
<td>QVR_12613</td>
<td></td>
</tr>
<tr>
<td>CategoryData1010</td>
<td>LIO_11412</td>
<td></td>
</tr>
<tr>
<td>CategoryData1012</td>
<td>900_2622</td>
<td></td>
</tr>
<tr>
<td>CategoryData1013</td>
<td>PMC_12196</td>
<td></td>
</tr>
<tr>
<td>CategoryData1014</td>
<td>PMC_12178</td>
<td></td>
</tr>
<tr>
<td>CategoryData1015</td>
<td>PSD_12336</td>
<td></td>
</tr>
<tr>
<td>CategoryData1016</td>
<td>PMS_12239</td>
<td></td>
</tr>
<tr>
<td>CategoryData1018</td>
<td>PIZ_12118</td>
<td></td>
</tr>
<tr>
<td>CategoryData1019</td>
<td>PMD_12221</td>
<td></td>
</tr>
<tr>
<td>CategoryData102</td>
<td>QVR_12613</td>
<td></td>
</tr>
<tr>
<td>CategoryData1021</td>
<td>PSD_12337</td>
<td></td>
</tr>
<tr>
<td>CategoryData1022</td>
<td>WFC_14940</td>
<td></td>
</tr>
<tr>
<td>CategoryData1023</td>
<td>R09_13871</td>
<td></td>
</tr>
<tr>
<td>CategoryData1024</td>
<td>O11_13</td>
<td></td>
</tr>
<tr>
<td>CategoryData1025</td>
<td>WSU_15092</td>
<td></td>
</tr>
<tr>
<td>CategoryData1028</td>
<td>PKR_12169</td>
<td></td>
</tr>
<tr>
<td>CategoryData103</td>
<td>QVR_12613</td>
<td></td>
</tr>
<tr>
<td>CategoryData1030</td>
<td>PES_12110</td>
<td></td>
</tr>
<tr>
<td>CategoryData1031</td>
<td>WFC_14837</td>
<td></td>
</tr>
<tr>
<td>CategoryData1033</td>
<td>O24_36</td>
<td></td>
</tr>
<tr>
<td>CategoryData1034</td>
<td>WSU_15091</td>
<td></td>
</tr>
<tr>
<td>CategoryData1037</td>
<td>PKR_12164</td>
<td></td>
</tr>
</tbody>
</table>

---

@naotar
Wafer maps

```
SELECT maskset, lotID, waferID, 
       AVERAGE(param) AS avg_yld, 
       visual.wfrmapcolorgen5(lotID, xloc, yloc, stepX, stepY, nrows, ncols, param, 
       0.0, 1.0, 0.0, 1.0) AS wfrmap 
FROM ( 
      SELECT maskset, lotID, waferID, w.xloc, w.yloc, m.stepX, m.stepY, 
              nrows - 1 AS nrows, 
              ncols - 1 AS ncols, 
              CAST(pass_fail AS FLOAT) AS param 
      FROM mfg.waferData w, 
          m.maskset m 
      WHERE 1 = 1 
      AND m.maskset = w.maskset 
      AND w.xloc = m.xloc 
      AND w.yloc = m.yloc 
      AND lotID = 'L50002' 
    ) x 
GROUP BY 1,2,3;
```
```python
from PIL import Image
from pylab import *
from numpy import *

@matplotlib inline

def histeq(im, nbr_bins=256):
    # get image histogram
    imhist, bins = histogram(im.flatten(), nbr_bins, normed=True)
    cdf = imhist.cumsum()  # cumulative distribution function
    cdf = 255 * cdf / cdf[-1]  # normalize

    # use linear interpolation of cdf to find new pixel values
    im2 = interp(im.flatten(), bins[:-1], cdf)

    return im2.reshape(im.shape), cdf
def compute_average(imlist):
    # open first image and make into array of type float
    averageim = array(Image.open(imlist[0]), 'f')

    for imname in imlist[1:]:
        try:
            averageim += array(Image.open(imname))
        except:
            print imname + ' ...skipped'
        averageim /= len(imlist)

    # return average as uint8
    return array(averageim, 'uint8')

im = array(Image.open('3360733.003_6C_6W732116SLC2_161110051647.JPG').convert('L'))
title("Wafer")
imshow(im)
figure()
title("Histogram")
hist(im.flatten(), 256)

im2, cdf = histeq(im)
figure()
title("Histogram Equalization")
imshow(im2)
```

Out[4]: `<matplotlib.image.AxesImage at 0x114fc0450>`
Oil Exploration Survey:
Tens of Thousands of Seismic Traces, which must then be processed

- Each Receiver collects one 6-8 Second Trace, each containing several thousand time-points
  - Example: 8 seconds of collected data @ 4ms sampling rate yields a 2000 entry trace
- Vessel can tow 10K++ receivers – meaning 10K++ traces collected per shot
Case Study: Caterpillar Large Power Systems Division
Sensor Data Qualification

Challenge
• Producing “qualified data sets” for data engineers to minimize wasted effort on data wrangling of sensor data

Solution
• A rules-based engine to set up and execute data management rules to deal with:
  • Light analytics and visualization that potentially gets pushed to the edge computing device

Results Actual (and Expected)
• Productivity improvements: Reduction in data engineer effort through standardization of data wrangling; data payload available 6 days earlier!
• Promise of reduction in central compute platform resources and network bandwidth

@naotar
Siemens Digital Services powered by Sinalytics –
Example: Predictive maintenance of trains and locomotives

Rail Transport
- Market drivers
- Rail operator challenges
- Rail user demands

Trains/Locomotives
- Rail vehicle engineering
- Mechanical vibrations
- Sensor properties
- Maintenance operations

Data Science
- Pattern identification
- Machine learning
- Automated alert generation

Results
- Improved asset availability
- Avoidance of unnecessary maintenance
- Reduction of maintenance costs

Domain know-how + Context know-how + Analytics know-how = Customer value

Siemens
Maintenance: Build a predictive model for engine failures on trains:

**Business Objective**
- Avoid unplanned train downtime through prediction model
- Repair trains before failure occurs, secure uninterrupted process

**Data Challenge**
- Sensor data from engines and data from maintenance management system
- Understand patterns to failure (pattern analysis)
- Understand correlation of errors / sensor readings around a failure
- Based on this insight, build a predictive model
- Example: daily pattern of temperature readings mid – low – mid often occurs 3 days prior to an engine problem

**Solution**
- Unified Data Architecture which includes Teradata Integrated data warehouse, Aster discovery platform and Hadoop
  - Leverage Aster to perform discovery analytics
  - Sensor data stored in Hadoop
  - Scoring of predictive model performed in Teradata Warehouse

**Opportunity to Impact**
- Pre-Dispatch required spare parts in time
- Avoid unplanned downtime, penalty, process interruption
- Save time on failure analysis
Smart Cities

- Smart Government
- Smart homes
- Smart healthcare
- Smart buildings
- Electric vehicles
- Enhanced safety & security
- Citizens, as customers, communities, prosumers
- Integrated, personalised retail
- Integrated travel & transportation
- Geolocation services
- Social interaction
- Smarter communications services

Data & analytics

The Smart City

The Internet of Things (IoT)
Why Smart Cities Need Integrated Data:

Independent Investigations

Intersecting Correlations
Car systems (today):

- Various technologies work in conjunction with each other to make a car autonomous.
- LIDAR Sensors
- RADAR Detectors
- GPS
- INS
- Electromechanical Systems
- Software and Algorithm
Avatars (virtual car):

- Decouple physical world from digital world
- Model the function independently from the device
  - Not all data needs to be stored
  - Devices can change, but function remains
  - Keep track of devices
- Similarities with the Party - Persona construct in Retail LDM

- Device Shadows [http://aws.amazon.com/iot/how-it-works/]
What is system of systems in this case:

- GPS
- Radar
- Traffic system
- Control Center
- Cars (n)
- Train data (AI)
- Route data and history
- Emergency data
- Historian (analytics at scale)
- Computation (model build, AI algorithms, Global Neural Network and Genetic Algorithms, ...)
- Emergency data (Accidents, construction, ...)
- Static maps

(TERADATA)
I want to stream my music.

Traffic?

I want my car to be a wi-fi spot.

What remote services do I use most?

I want access to my email.

How far can I go on the remaining charge?

I want to schedule service.

What happened on the last hard brake?

Does my car need an upgrade?

What else should I have checked while my car’s being serviced?

I want my car to remember me.

What are the road conditions ahead?

I want my car to be a wi-fi spot.

I want to find my [stolen] car.

How risky is my teen’s driving?
THANK YOU: