Semiconductor Factory Data Integration & Analysis: 2 Large Projects 2005-2007 and 2010-2011 Case Studies, Legacy Factories “R” and “A”

Michael Clayton, Independent Contract Engineer
3055 N Red Mtn, Unit 153, Mesa, AZ 85207 (480-332-4076)
Carnegie Institute of Technology, BS Metallurgical Eng, 1962; ASU, MBA 1972, SSBB – Motorola 1993

Introduction

Manufacturing data integration is painfully detailed, slow, expensive process if you want the data to reside in a database that allows fast loading and fast unloading (queries for analysis.) The sources of the data involve complex factors:

- Real-time operating systems on testers and process tools
- Tool/tester data communications protocols are variable
- Context queries of WIP tracking DB, SPC, metrology DB needed by lot
- High speed sampling of process machine sensors with:
  - Requirements for high precision at high sampling rates for fault detection

Thus the team has to include Engineers from Process, Test, Production, CIM, IT, and perhaps outside experts as needed for DBA issues, or binary decoders for older equipment, or decisions as to precision and scaling of test data, and statistical decisions about sampling rates for sensor-based data from machines.

Two multi-year projects were successfully completed for Semiconductor Manufacturing Operations at Company R (2005-2007) and Company A (2010-2012). JMP was used for queries and analysis, replaciing the old proprietary GUI’s for most engineers. Hundreds of data sources were integrated to existing proprietary yield management databases (YMS) running on Oracle DBMS [PDF dataPOWER in Company R, and Klarity ACE in Company A.]

Scope for Company R:

Wafer fabs in 2 countries, 22 Assembly and Test Sub-cons globally.

MES (WIP tracking), SPC (metrology) Device and Circuit test data.

Dozens of process flows, hundreds of products, millions of parts each year.

Data must be stored for traceability for 7 years (automotive requirement)

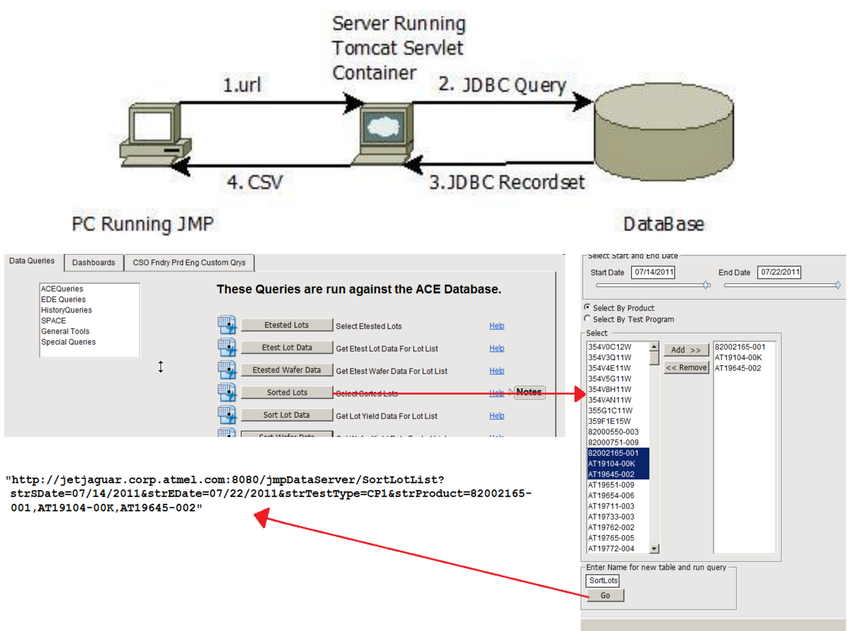
Scope for Company A:

One US wafer fab, and 5 global foundry fabs, plus 5 assembly and test sites.

Same volume and type of data as Company R roughly, but most data can be archived after one year (consumer products).

Note: Both companies ran older generation of process and test equipment, requiring special cell or station controllers to capture data. Upgrades to these controllers were part of the project.

At company A, a unique data server was created to avoid constant upgrades to individual PC’s, and that was subject of prior poster at Denver Discovery Conference by Neil Verzwyvelt of Atmel Corporation. (*pictures below for reference*)



Methods for data integration project

Company R: Phase One used “waterfall” type IT planning and execution, while Phase Two used Agile rapid-prototype approach which cut integration costs in half. [Detailed requirements planning and C code was much slower than Agile rapid prototyping and Perl/Python code.] As project neared completion, central and local IT teams developed more secure FTP systems for external (sub-con) sites.

Local skills determine best project strategy, and using local talent leaves expertise within the operations but in this case, data integration skills were lacking locally. So heavily documented code was critical. Phase One required team of 4 part time programmers. Phase Two involved more global assembly and test data sources, and more delays, so only one part time programmer was used assure continuity over long delay times. But there was time to assure local IT expertise was enabled for Perl.

Company A : Both wafer fab and backend phases used Agile approach, and newer JMP version for scripting (v8 rather than v5 or v6). [Multi-year script support can be big issue in legacy operations.] Moving most scripts to data server rather than PC cuts support costs dramatically. Perl scripts were primary formatting tool, but gradually Python became more popular, easier to maintain for complex efforts.

Both companies used Oracle DBMS but hosting different proprietary yield management DB schemas (PDF dataPOWER vs Klarity ACE). But since Oracle was common, the queries could use the Oracle JDBC interface. However unique knowledge of the YMS schemas was critical. We found one contract analyst-programmer for company R and one in-house IT staff analyst at company A that had supported those unique YMS schemas.

Equipment vendors often did not support their older tools, but ex-field engineers were found that could help with key information when needed. (Example: SECS II RS-242 data ports vs High Speed SECS protocol using Ethernet port, or special binary-to-ascii translators not provided by system vendors.

Discussion of training and results

- Over 100 engineers were trained to use JMP for queries and analysis, as well as design of experiments, at each company. If JMP Query capability is established early, JMP users can help IT and CIM data integration folks by sanity-testing the “rapid prototype” loading. CIM and IT teams learned to map complex networked systems.
- The multi-year integration effort was sustained and file loading success was tracked with JMP graphical reports to management monthly . Key driver for data owners was the “missing file” report (% Sent and % Loaded successfully).
- Critical data which had been missing from costly legacy yield management systems was captured and cleanly loaded, but it took > 1 year for fab data phase, and another year for assembly test phase. Prior to that, engineers used mostly flat file repositories and a myriad of graphical software tools, mostly Excel.
- New product yield learning curves improved greatly, and new hardware was added as systems grew, including storage area networks, and archiving systems
- Cost for legacy DB upgrades and license support was reduced, in favor of JMP support as the GUI of choice, customized for local requirements by local CIM engineers. Same JMP query capability can reach many mfg DB’s.. Proprietary YMS schemas were fixed at old version to avoid support costs. But JMP for key users was updated to current version at each company.
- Due to need for ongoing help from busy engineers and managers, the rapid prototype method, and the early adoption of JMP query capability for existing production and engineering DB’s were critical. Even though these were legacy factories, the investment paid off in yield improvement and customer delight with quick response to any field problems. Any questionable analysis could be re-done from DB .
- JSL code change rollout was simpler at Company A due to data server instead of local PC location of query scripts, which is recommended approach for future efforts anywhere based on our experience over past few years.

Acknowledgement

The author wants to thank SAS/JMP sales and training support, and the local CIM and IT and Test teams at each company. And thanks to Neil Verzwyvelt for his earlier poster on the data server tools.