ISMI RaP Prototype

Jim Hollister
Alan Weber & Associates

Lance Rist
ISMI

Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

Prototype Motivations

- **Improve the RaP specification**
  - **Clarity and organization of the document**
  - **Completeness and consistency of the information**
- **Verify that RaP is implementable**
  - **Identify design and implementation issues related to the RaP specification content**
  - **Modify the specification to make implementation easier, where practical**
  - **Demonstrate RaP usage scenarios**

Challenges to Prototyping

- **When To Do It**
  - Too early – specification isn't firm enough
  - Too late – specification is too rigid
  - Accidentally chose a good solution
    - Just after the concepts specification is approved and during development of implementation standards
- **Flexibility Is Critical**
  - Some parts of the document continue to change during the prototyping process
    - To some degree, because of the prototype effort
  - Must accept that the prototype will not represent the final SEMI specification 100%
  - But must react to changes that have substantial effects on the result

Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

Prototype Approach

- **Simulate the process a supplier will use to add RaP support to equipment**
  - Generate detailed software requirements for adding RaP support
  - Design and build a prototype implementation
- **Deliver a lessons-learned report at the end of each SW development phase**
  - Provide feedback to the RaP Task Force as soon as possible

Prototype Phases

- Protocol Mapping
  - Created a “stand-in” SECS-II mapping
- Requirements Analysis
  - RaP support for the TrackSys TS-400 tool
- System Analysis
  - Impact on tool HW & SW
  - Relationship to other SEMI standards
- Design
- Implementation
- Demonstration

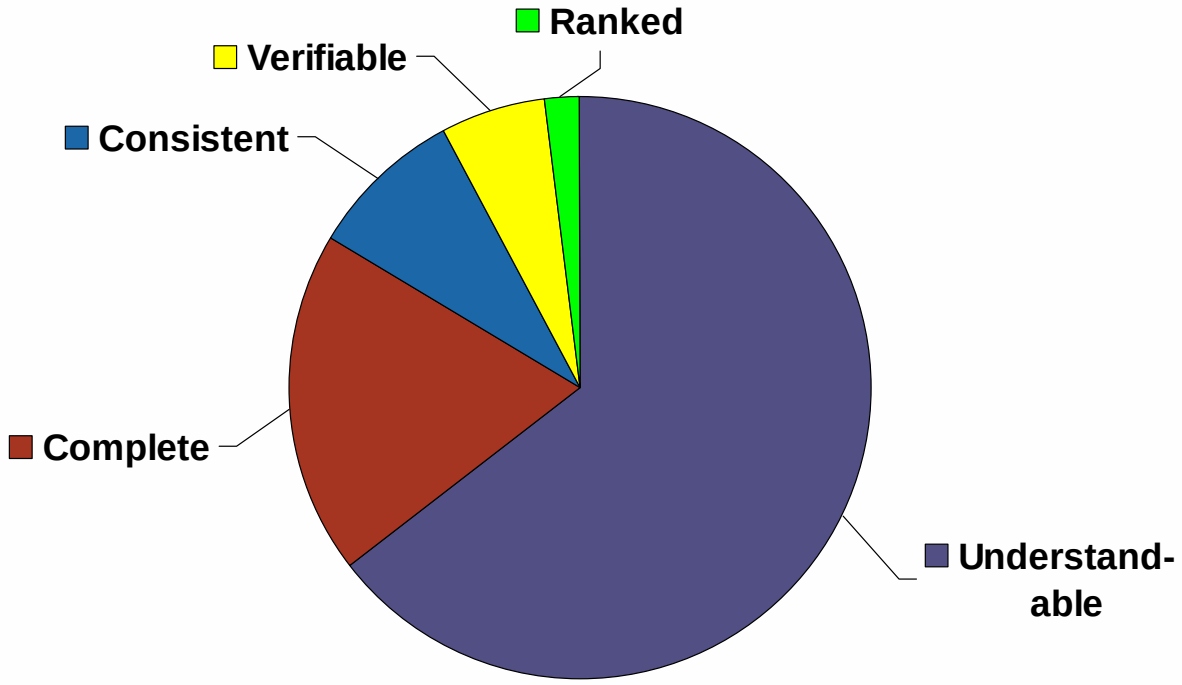
} of a RaP Prototype

Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

RaP Requirements Assessment

- Assess the RaP specification as a set of software requirements
  - Assessment criteria used were a subset of the criteria in the IEEE Software Requirements Recommendation (IEEE Std 830-1998)
    - Unambiguous (Understandable)
    - Complete
    - Consistent
    - Ranked
    - Verifiable
- Assessment is contained in “TrackSys TS-400 Software Requirements” deliverable

Requirements Issues Breakdown



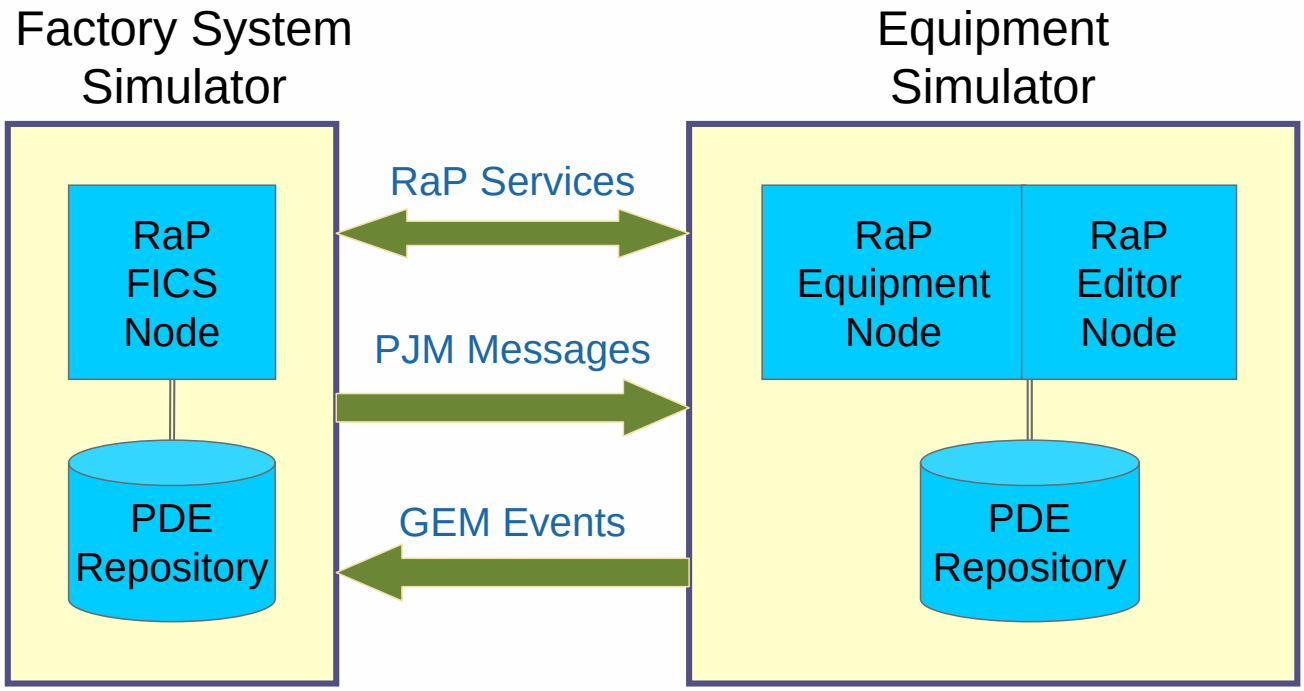
104 issues identified during Requirements Phase

Sample Requirements Issues

- **Understandability**
  - 8.4.3 The text says “This PDEeditor shall be capable of initiating transfer of PDEs to and from the equipment.” What if the equipment and editor share recipe storage area?
- **Completeness**
  - 8.3.5 The description of the TransferContainer manifest does not describe the content of the manifest when PDEheaders are being transferred.
- **Consistency**
  - 8.3.3, Table 16: The entry for tcid says its form is UUID; the entry for uid says its form is String, and there is no entry for gid. These should all (tcid, uid, gid) have consistent data types.

Note: PDE -> Process Definition Element = Recipe Component

RaP Prototype Components



- **The prototype consists of two applications:**
  - **Factory System Simulator**
  - **Equipment Simulator**

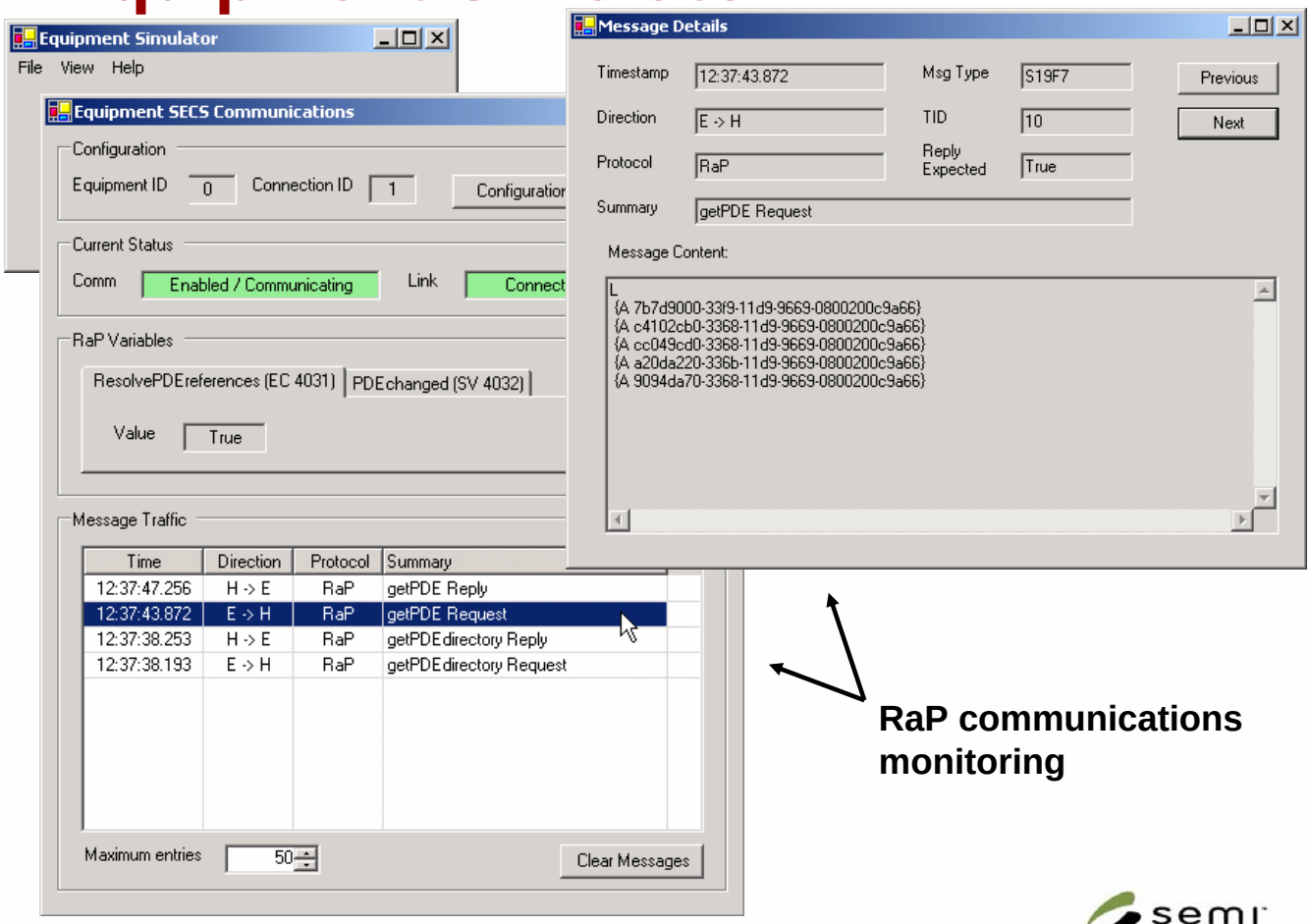
Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

RaP Prototype Implementation

- **Technologies Used**
  - **Microsoft .NET and C#**
  - **Cimetrix CIMConnect for SECS-II and HSMS communications**
  - **SQLite (a lightweight RDB) for persistent storage of recipe metadata**
    - **Windows file system for recipe body storage**
  - **SharpZipLib for recipe compression and multi-file archive support**

Note: None of these technology choices were dictated by the RaP specification

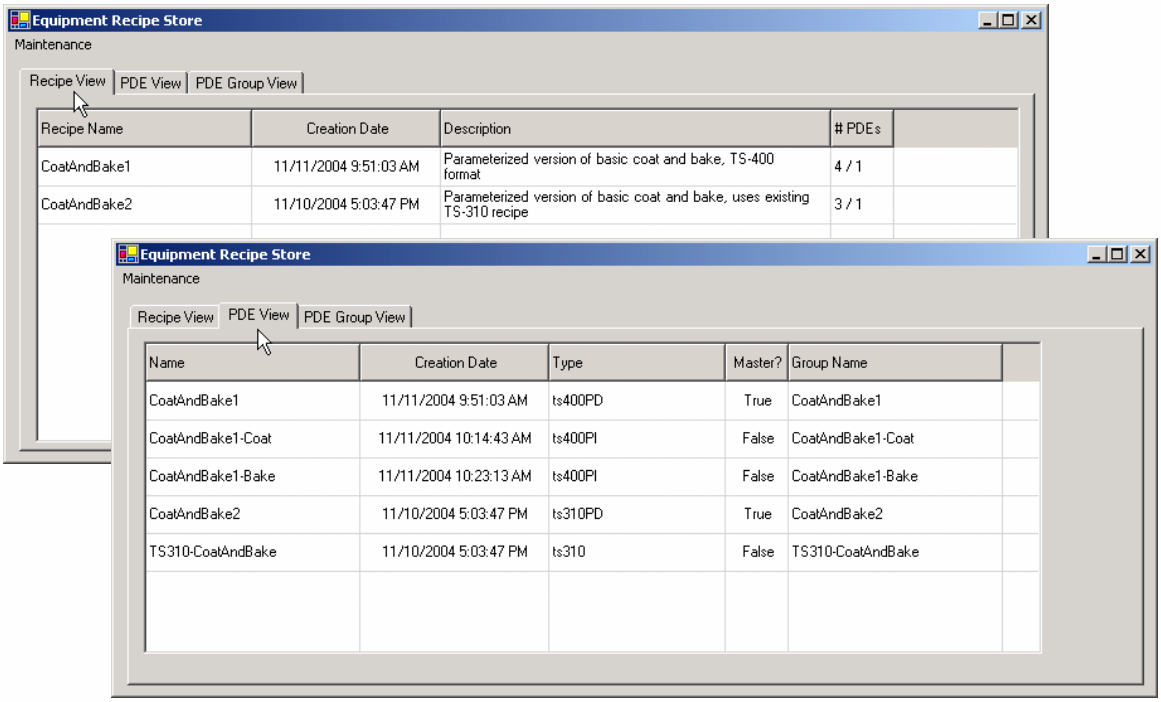
Equipment Simulator



RaP communications monitoring

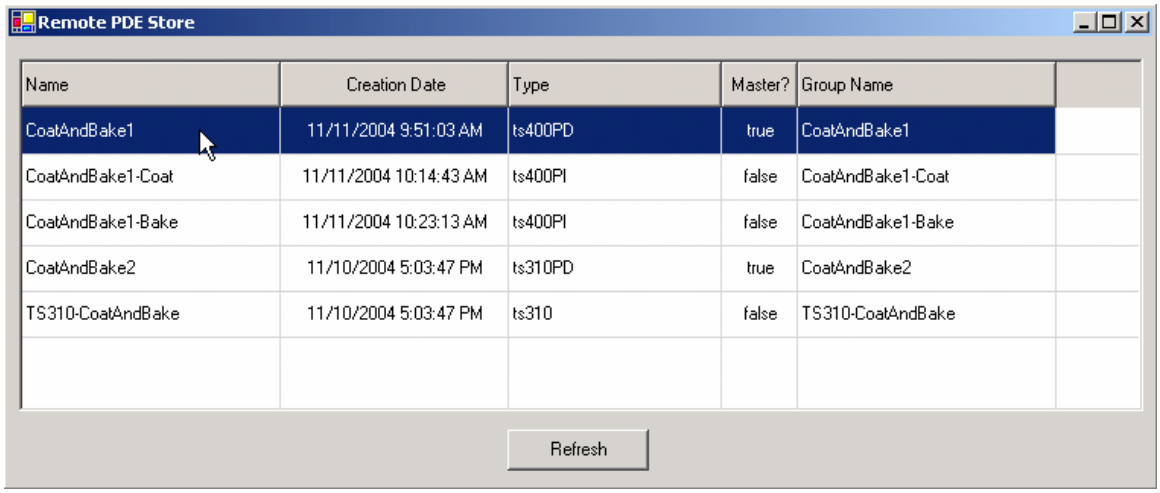
Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

Equipment Simulator



Multiple views of local recipe store contents

Equipment Simulator



View of remote recipe store contents. Needed to support getPDE (download) service.

Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

Design And Implementation Issues

- **Calculating checksums over XML documents**
  - **XML supports multiple (byte-level) representations for the same document content, so:**
    - **XML must be fully canonicalized before checksum calculations**
    - **The "original" XML must be accurately reproduced from recipe storage in order to verify checksum**
- **Recipe storage must be carefully designed to avoid having to open/read every stored recipe component (PDE) when:**
  - **Servicing a getPDEdirectory request**
  - **Traversing PDE dependencies within a recipe**
  - **Determining PDE group membership**

Very Large Recipes (> 4 GB)

- **Difficulties due to large recipes**
  - HSMS limits total message size to 4GB; SECS-II limits single data item size to 16MB*
    - RaP SECS mapping must work around these constraints
      - Probably using multi-message transfer of recipe
    - Implementation of this mapping will be more complex than for a single request-response message pair
  - RaP implementer must be careful not to instantiate entire recipe in memory
    - RaP Prototype makes makes heavy use of Streams to pass recipe data from one sub-system to another
      - Streams are read or written as infrequently as possible
    - Must be sure that helper utilities avoid in-memory instantiation as well

*** These constraints don't apply to the RaP WebService (SOAP/XML) mapping.**

Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

Very Large Recipes (> 4 GB)

- **Large recipes lead to long processing times**
  - Recipe transmission time
  - Local and remote recipe processing time
    - XML parsing, XML canonicalization
    - Checksum calculation and recipe content validation
    - Data compression/decompression
    - File I/O
  - As a result, any of these operations can take a long time:
    - Local generation of a RaP request and/or processing of the RaP response
    - Remote request processing; remote response generation
    - Message transmission to and from remote node

Very Large Recipes (> 4 GB)

- Large recipes lead to long processing times
  - It may not be feasible to block other host or equipment operations
    - First and foremost, wafer processing must not be affected
    - Process or metrology data still needs to be collected
    - User interfaces need to stay responsive
  - The solution is asynchronous (multi-threaded) programming
    - Multi-threaded programming can be complex and error-prone
    - It's hard to think in parallel

Prototype Deliverables Summary

- Reports from each phase of the activity
  - Identifying issues with the RaP document and content
  - Identifying design and implementation issues
- RaP support requirements
  - For a sample process tool: TrackSys TS-400
  - Lots of detail, 57 pages
- A prototype RaP implementation
  - Factory System Simulator
  - Tool Simulator

Prototype Activity Assessment

- This effort was effective in identifying a large number of potential RaP improvements
  - Many of these have already been incorporated into the specification
- The Requirements Phase was most effective in identifying document and content issues
  - Use of SW requirements assessment criteria could be adopted by other standards teams
- The Design and Implementation Phases were necessary to identify implementation difficulties
  - Fewer issues, difficult to uncover before implementation

Standards Technical Education Program on the Recipe and Parameter Management Standard (SEMI E139)

Conclusions

- Prototyping is particularly effective when
  - The main (concepts) document is near approval
  - The implementation standards are in early draft
- Stable commercial implementations possible up to two years earlier
  - Changes detected by early prototyping would ordinarily be
    - Found during commercial implementation ~6-12 months later
    - …and corrected ~6-12 months after that (12-24 months total)
- The industry would benefit from a process that encourages/requires early prototyping
  - A model is needed for how to fund and execute such prototyping activity
  - ISMI may do this for select priority issues, but the industry needs it done consistently for all cases