JPL

Information-Driven Control of Product Generation and Reconciliation

Generating Cassini ISS & VIMS data products on a budget

A.Culver, R.Patel, A. Stanboli, H.Lee, S. Noland, J.Diehl,
C.Avis, J.Henricks

JPL

Agenda

- Driving Requirements & Implications
- Design Challenges
- Static and Dynamic Inputs to Processing
- Our Solution - Database-centric design
- Server Descriptions
- Conclusions
- Acknowledgements

JPL

Driving Requirements

- Minimize Operations Costs
  - Prime shift operations
  - Small workforce numbers
  - Long duration mission
- Ensure Timely Performance
  - Initial version of product within 20 minutes
  - Product accounting and assessment reports in 12 days
  - Final version of product in 12 days

JPL

Driving Requirements (cont)

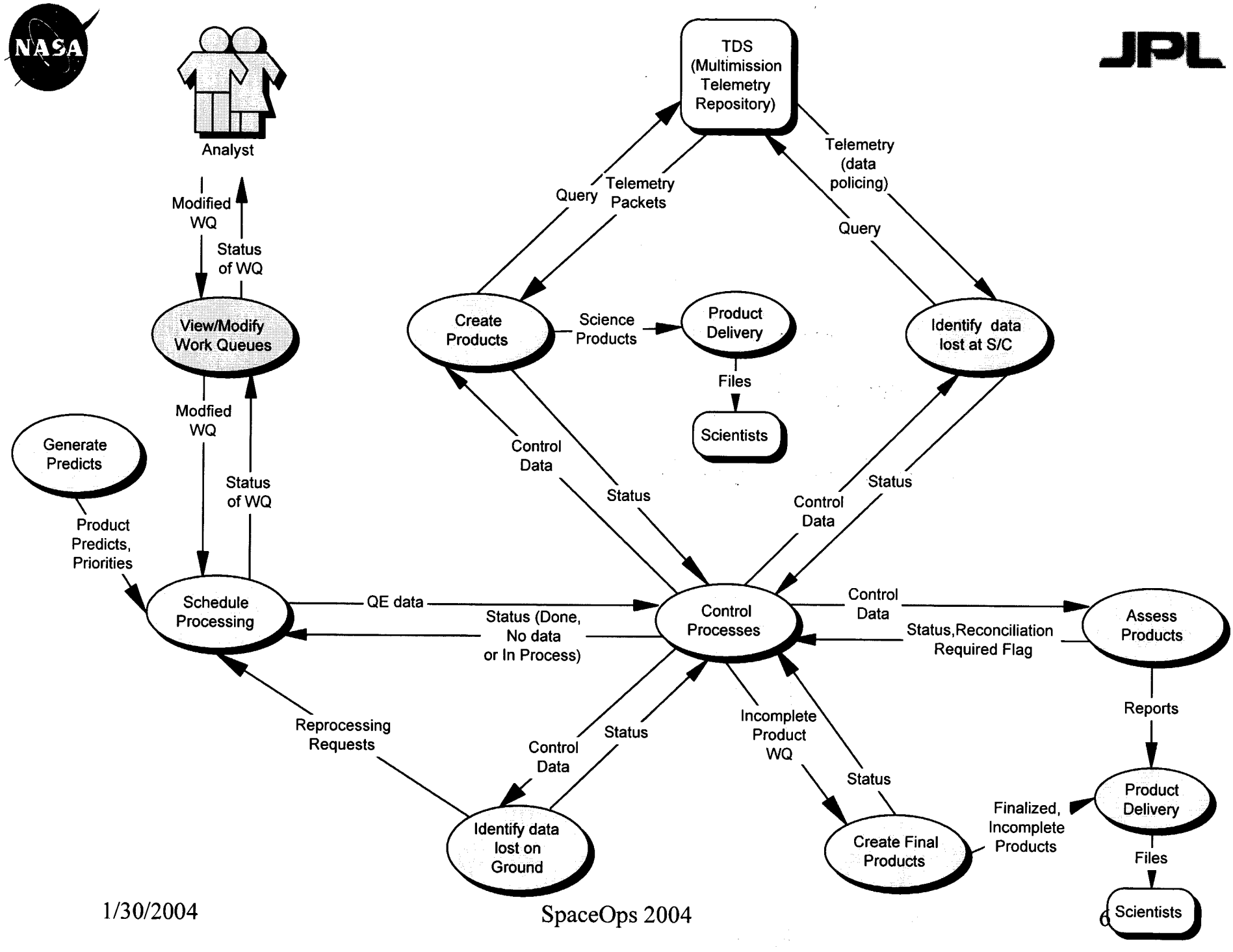
- Constrain development cost
  - Integrate with Multimission capabilities
    - Reduces development cost to acquire telemetry data
    - Limits flexibility
    - Introduces interface issues, maintaining connectivity across firewalls, using remote servers, managing configuration changes
  - Reuse ground software developed pre-launch
    - Transition from manual product generation program to automated system capable of high volume production and reconciliation

JPL

Which boils down to...

Self monitoring, lights-dim system capable of:

- Scheduling telemetry processing
- Accessing and deciphering telemetry
- Product formatting, delivery and quality assessment
- Reconciling received versus expected products
- Accounting for missing data


NASA
JPL
Analyst
TDS (Multimission Telemetry Repository)
Modified WQ
Status of WQ
Query
Telemetry Packets
Telemetry (data policing)
Query
View/Modify Work Queues
Create Products
Science Products
Product Delivery
Identify data lost at S/C
Files
Scientists
Modfied WQ
Status of WQ
Control Data
Status
Control Data
Status
Generate Predicts
Product Predicts, Priorities
Schedule Processing
QE data
Status (Done, No data or In Process)
Control Processes
Control Data
Status,Reconciliation Required Flag
Assess Products
Reports
Reprocessing Requests
Control Data
Status
Incomplete Product WQ
Status
Identify data lost on Ground
Create Final Products
Finalized, Incomplete Products
Product Delivery
Files
Scientists

JPL

Design challenges

- Develop information driven system (rather than human driven)
- Access and utilize information on predictable events
- Replace traditionally human control functions by software components
- Implement automated, self-monitoring, fail-safe system which allows for human intervention

JPL

Static Information Needed to Drive Automation

- Processing rules and parameters
  - Job and Task sequencing
    - Relative timing of jobs
    - Sequential list of tasks to be performed to accomplish a job
  - Location of input and output data
  - Information each task requires to do its job
  - System configuration
    - Server locations
    - Server assignments

JPL

Static Information Needed to Drive Automation (cont)

- Project
  - Mission phases
  - Sequence boundaries
  - Product routing information
- Instrument
  - Flight Software versions

JPL

Dynamic Information Needed to Drive Automation

- Predicted events from sequence generation
  - expected products
  - downlink windows
- Generated product information
  - Product metadata
  - Assessment results
- System resource availability
- Task status to determine future processing (including directives from humans)

JPL

The Design

- Use database as persistent repository of static and dynamic information
- A set of distributed servers which startup and execute processing based on information residing in the database
  - Enables safe server restart
  - Servers determine outstanding work based on database (stateless servers)
  - Minimizes corruption of system in the event of failure
- Distribute processing across many machines (servers primarily communicate via database)

JPL

Server Functionality

- Request Analyzer
  - Ingests predicted events data, defines all data processing jobs
  - Splits jobs along project, instrument boundaries

JPL

Server Functionality (cont)

- Scheduler
  - Query database for jobs eligible for processing
  - Release all eligible jobs to Control
- Control
  - Accept jobs to initiate from Scheduler
  - Query database for available SubControl servers to accomplish the jobs
  - Hand-off jobs to SubControl servers
  - Update database with processing assigned to SubControl
  - Monitor system - ensure that each SubControl server is executing jobs as expected

JPL

Server Functionality (cont)

- SubControl
  - Accept jobs to process from Control
  - Accept outstanding jobs to process from the database
  - Query database
  - Identify tasks and task execution sequence needed to complete job (rules table)
    - Task processing parameters (jobConfig table)
  - Construct ancillary parameter files needed by tasks
  - Initiate sequential task processing necessary to complete the job
  - Handle and report error conditions

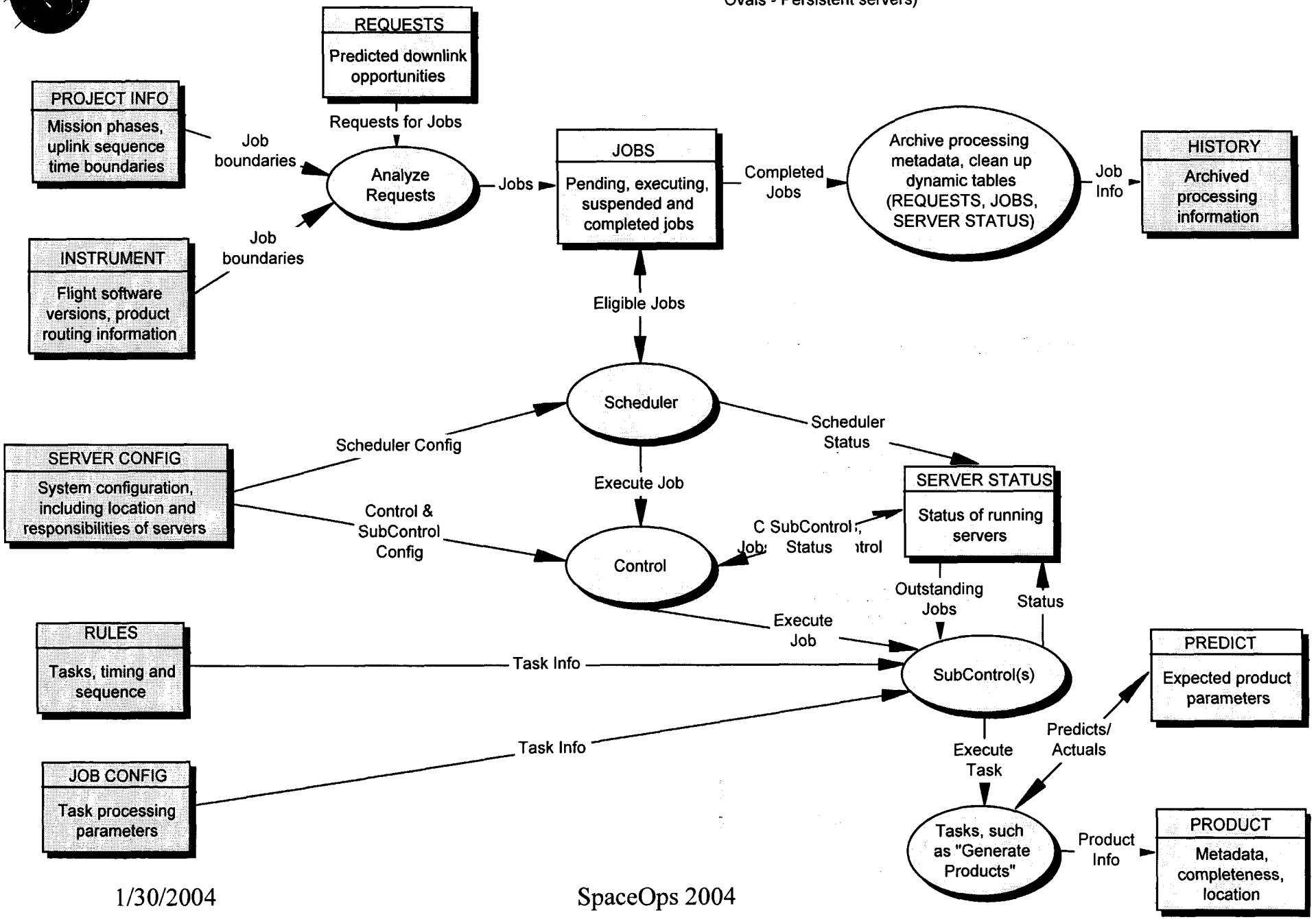
JPL

Server Functionality (cont)

- Post-Request Processor
  - Need for this server derived from the design (to maintain system performance)
  - Clears dynamic data from database, archives in static table
  - Preserves archiving processing metadata

Database/Server Interaction Diagram

(Rectangles - Database tables,
Ovals - Persistent servers)



Conclusions

- Minimized workforce required per product to schedule, generate, and reconcile data products
- System is self-monitoring
- Compared to Jupiter product generation and reconciliation, we can handle 4X the work, with 1.5X workforce

JUPITER	SATURN
Manual production	Lights dim production
5,000 data products/month	19,000 data products/month
Prime shift, 2-3 times/week	Round-the-clock production
Limited data accounting	Full data accounting
30+ days to complete reconciliation	12 days to complete recon.
1.7 employees needed	2.5 (or fewer) employees needed

JPL

Acknowledgements

- This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.