Rapid Data Prototyping

Shaping Directionless Data into Useful Information

Event Date(s): June 19 thru 22, 2005

Speaker Name: Rusty Johnson
Speaker Title: Senior Architect, AIMP
Northrop Grumman Corporation
Rapid Data Prototyping (RDP)

The tools and techniques enabling the gathering, manipulating, and integrating of data with domain-specific knowledge to deliver information
Essential RDP Qualities

- *Lingua franca* for processing data
- Compact, tactile data manipulation
- Integration of domain-specific concepts into data processing
- Improved communications
- Large domain independent tool set
RDP Success

- 18 years of answering information requests that “no one else can answer”
- Lisp based and easily extensible
- Portable – have changed platforms without impacting production or customers
Why RDP?

- Decision-maker policies drive information requirements
- Data capture evolves trying to keep up
- Our understanding of data evolves
- Information requests are many times imprecise
- Deadlines are always short
Free the developer from the machine and processing details

Bring the developer and the customer onto the same playing field
Freeing the Developer

- Concise, crisp control of processing
- No *a priori* knowledge of data structures required
- Data accessed by name, not position
- Improve developer throughput

Developer focuses on “what”, not “how”
Into the Customer’s World

- Adoption and integration of domain abstractions
- Increase expressiveness AND control in ways meaningful to the customer
- Communicating in the “right” language improves quality and timeliness
- Elicit ownership from Subject Matter Experts (SMEs)
RDP Roles

- **Decision Maker**
- **Information**
- **Request**
- **Subject Matter Expert**
- **Data & Information**
- **Request & Refinement**
- **Developer**
What is a Subject Matter Expert?

- Typically passionate about specific domains
- Usually very smart
- Frequently blissfully unaware of underlying data issues
- Always overtaxed
Developer, SME Team

- Must educate each other
- Identify processes to be formalized
- Iterate quickly refining results
  - Keep the team focused

“Be the Ball” – Ty Webb
RDP Tool Suite

- Data Acquisition
- Data Transformation
- Information Output
- Data Warehousing
- Domain-Specific Tools
- Portability
Portability Layer

- Data Acquisition
- Data Transformation
- Data Warehousing
- Information Output
- Domain-Specific Tools
- Portability

<table>
<thead>
<tr>
<th>Common Lisps</th>
<th>HTTP Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Open Genera</td>
<td>- CL-HTTP</td>
</tr>
<tr>
<td>- Allegro</td>
<td>- AllegroServe</td>
</tr>
<tr>
<td>- LispWorks</td>
<td></td>
</tr>
<tr>
<td>- MCL</td>
<td></td>
</tr>
</tbody>
</table>
Data Acquisition and Warehousing Layer

Data Acquisition
- Common File Formats
  - XML
  - CSV
  - Tab Delimited
  - Fixed Width
  - DBF
  - Lisp
  - Dataset
  - HTML

Data Transformation

Data Warehousing
- Logical databases
- Versioning databases
- Synchronized sets

Information Output

Domain-Specific Tools

Portability
Data Transformation Layer

The *dataset* is the key abstraction
- A “pile” of records
- Description of the record structure

Operations are written only in terms of data requirements
Dataset Operations

Fields are accessed by name and not by position

- Filtering and structuring
- Set Operations
- Summarizing (Rollup)
- Rotating, Joining, and Merging
- Sorting and Comparing
Lazy Dataset Evaluation

Dataset operations can be deferred

- Operations are daisy-chained into a pipeline
- Data is pulled through the processing pipe
- Data streams through dataset operations, rarely pooling in large collections (reduce memory allocation)
- Significant increase in throughput, no change in readability (changing one keyword argument)
- No data moves until the dataset is finalized

Think data fire hose instead of bucket brigade
Information Output Layer

- Delivery to databases, windows, printers, browsers and various file formats
- Uses standard layout engines based on report specifications and output media
Domain-Specific Tools Layer

- Speaking the customer’s language
  - Encapsulating SME’s knowledge
  - Adding domain information
  - Encoding domain practices and processes

Enabling developers with little domain knowledge to get consistent, correct results
Implementation

Current hardware, HP DS-25 (4 GB memory):
- Developer WS: Open Genera 2.0
- DB Servers: Allegro 6.2 (64-bit)

Previous Ports:
- LispWorks (Unix & PC), MCL
Conclusions

Rapid Data Prototyping:
- Reduces cultural impedance between the SME and developer
- Increases data-handling flexibility and dexterity
- Promotes integration of domain concepts
- Supports formalization of prototypes into enduring applications

Rapid Data Prototyping gathers data, manipulates it using domain-specific knowledge, and delivers meaningful information on demand.