



RULES FEST



Executing Processes, Taking Decisions and Detecting Situations

Daniel Selman

Product Architect

IBM WebSphere ILOG BRMS

Agenda

Why should we care?

For each in {process,decision,events}:

- Description
- State
- Runtime
- Context
- Reuse
- DSL
- Human interface
- Dealing with Change

How to combine them?

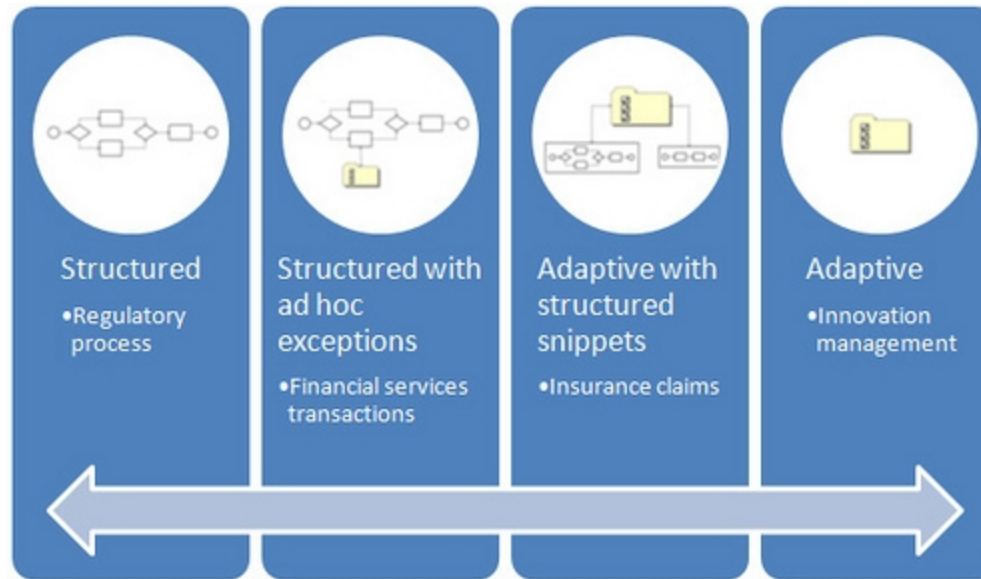
Conclusions

Why?

- Businesses need to:
 - Orchestrate long-running stateful business processes
 - Respond rapidly to system and business events and take appropriate action
 - Make consistent, reliable and high-quality decisions
- What are the characteristics of the available technologies and how should they be combined?

Process

- Wikipedia: “BPM considers processes to be strategic assets of an organization that must be understood, managed, and improved...”
- E.g. Order to Cash
 - How does your business convert a customer order into cash on the balance sheet?



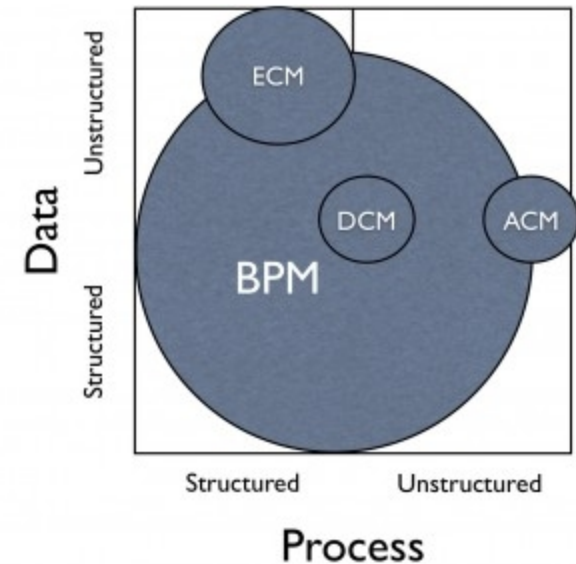
(Sandy Kemsley)

BPM (Business Process Management) – The focus is on structured data (forms) and structured flow.

ECM (Enterprise Content Management) – The focus is on unstructured data (documents) and structured flow.

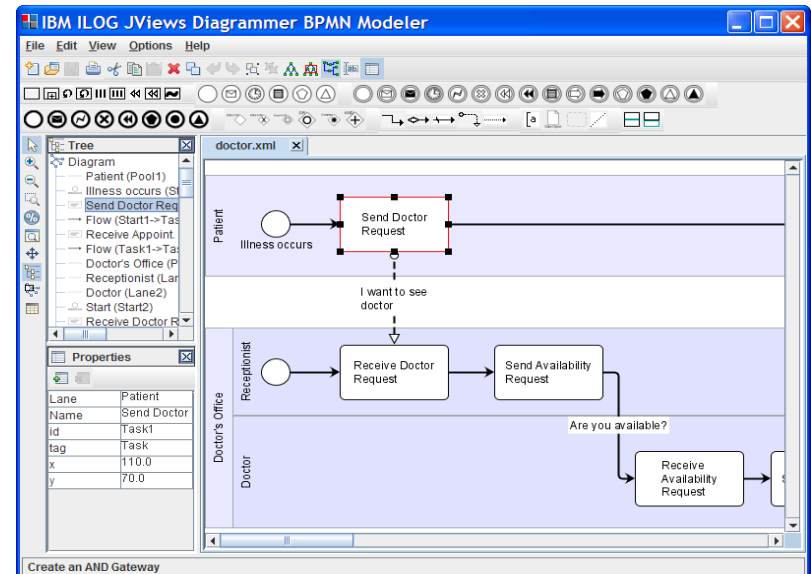
DCM (Dynamic Case Management) – The focus is on structured+unstructured data (forms and documents) and semi-structured flow.

ACM (Adaptive Case Management) – The focus is on structured+unstructured data (forms and documents) and unstructured flow.



State

- Stateful and long-running.
- Transitions are explicit. Process diagram is a vital artifact and communication tool



Runtime

- BPMN2 or proprietary
- State stored in relational database
- Typically asynchronous and long-running
- Message queues used for async communication

Context

- The context is determined using message correlation (custom or automatic):
 - Loan application XYZ
 - Customer: Daniel Selman
 - Sex: Male
- A process instance defines values for process variables

Reuse

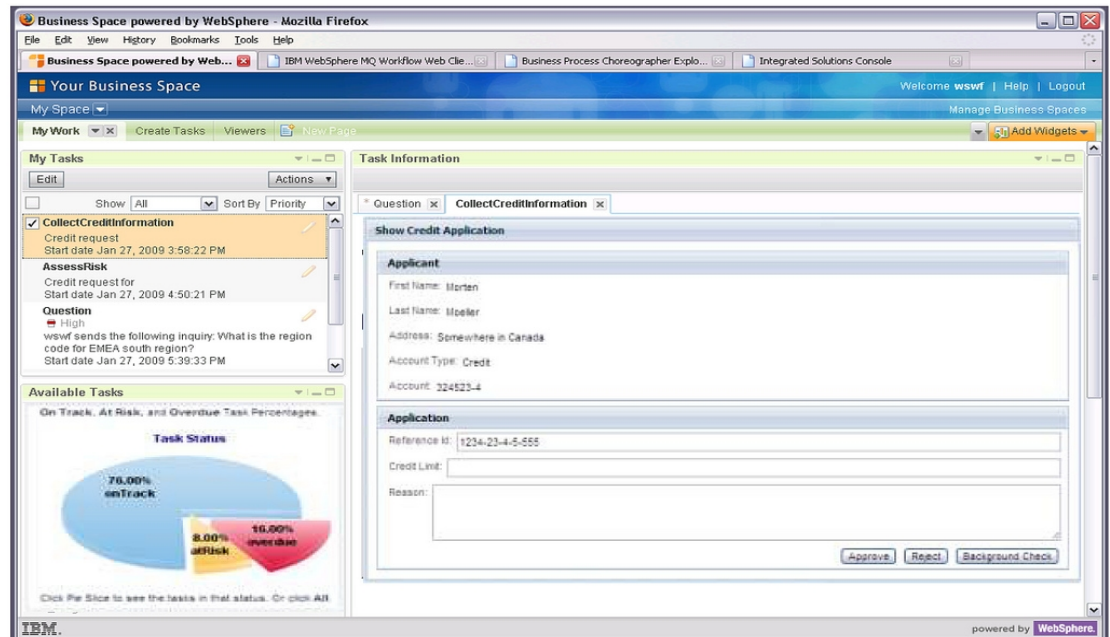
- Libraries contain processes and models
- Processes can invoke sub-processes from other libraries

DSL

- BPMN2 diagram for process definition
- BPEL diagram to wire web services
- Scripting languages/rules to implement tasks

Human Interface

- Generate forms/pages for data entry
- Task lists
- Process instance management



Dealing with Change

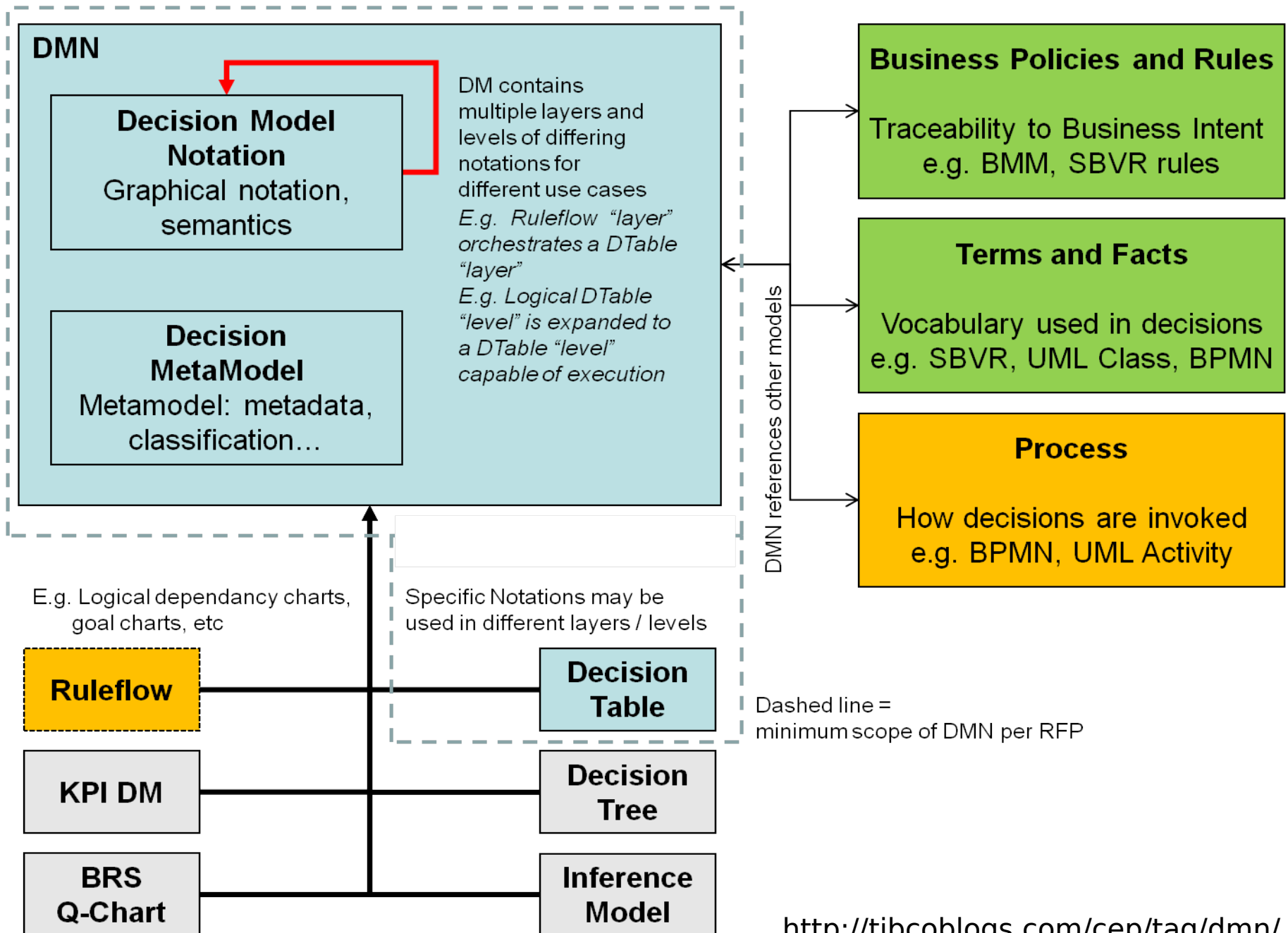
- Policy changes: Medium
- Model changes: Hard
- Requires:
 - Retrain humans
 - Update diagrams
 - Update screens
 - Migrate process instances
 - Update test data

Decision

- “The process of selecting from several choices, products or ideas, and taking action.”
- E.g: Is customer eligible for seat upgrade? What rate should we offer Daniel Selman?

Classification, Evaluation, Selection,
Approval, Assessment, Assignment,
Allocation, Diagnosis, Prediction





State

- Typically stateless (idempotent?) and short-lived (a decision at a point in time).

Runtime

- OUT Function(IN)
 - OUT and IN are often Maps of name/value pairs
- Implementation may be:
 - Ruleset (RETE or sequential)
 - Predictive model (Bayesian)
 - Code

Context

- Context is supplied by client (IN data)
- Context filtering may be performed by conditions of rules:
 - ?c : Customer
 - ?tx : Transaction where ?c.id == ?tx.id
 - If ?tx.amount > 100 then ...

Reuse

- Atomic rules are easy to reuse across decisions
- However some rules have significant implicit pre-conditions and are hence hard to reuse.

DSL

- If/then rules
- Decision Tables
- Decision Trees
- Ruleflow
- Scorecards

Human Interface

- Form generation
- Interactive Q&A to prompt operator to supply missing data.

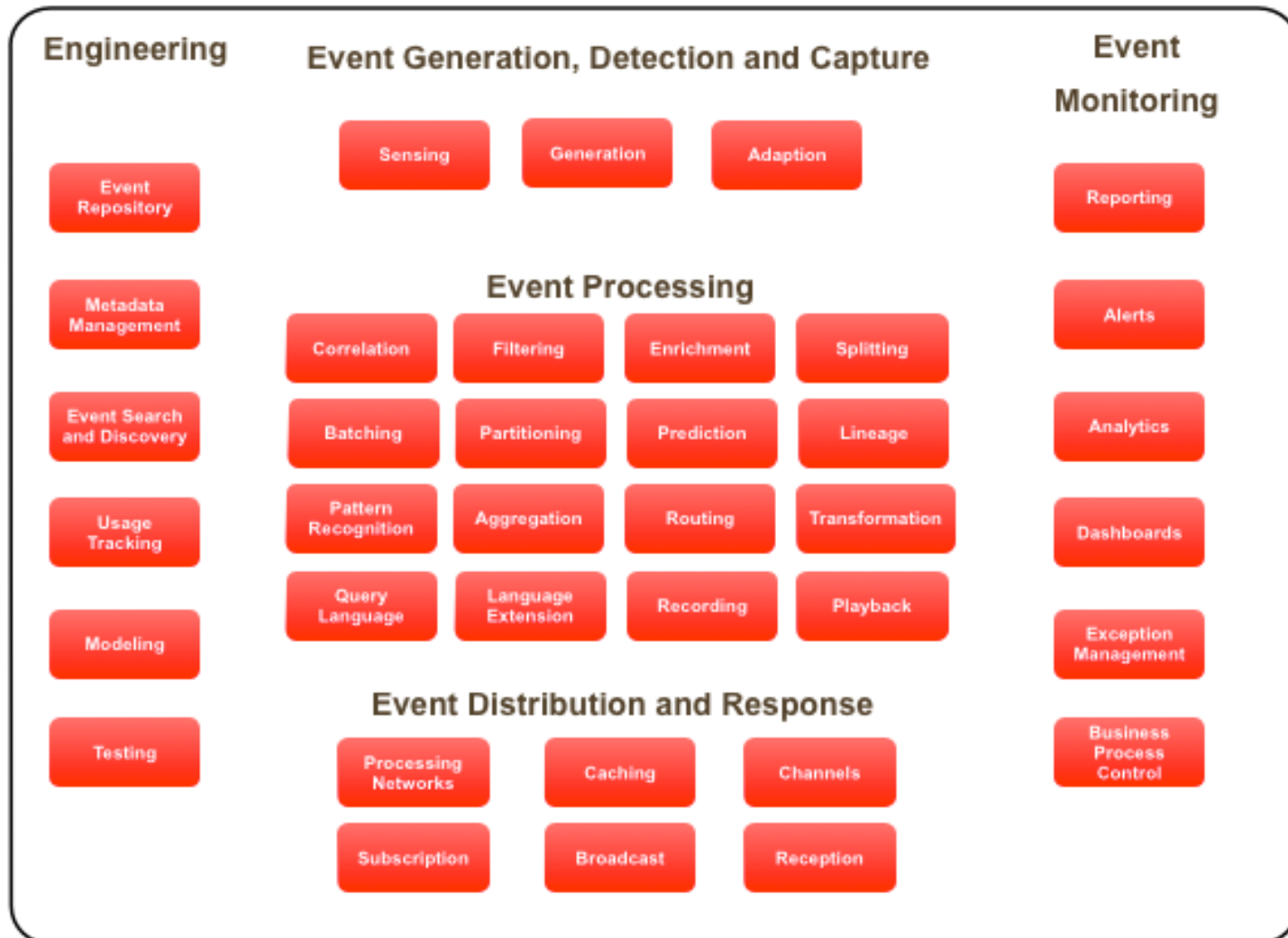
Dealing with Change

- Policy changes: Easy
- Model changes: Medium
- Requires:
 - Update business rules
 - Update test data

Event Processing

- Wikipedia: "...consists of processing many events happening across all the layers of an organization, identifying the most meaningful events within the event cloud, analyzing their impact, and taking subsequent action in real time."
- E.g. monitoring and response to alarms

Functional View



State

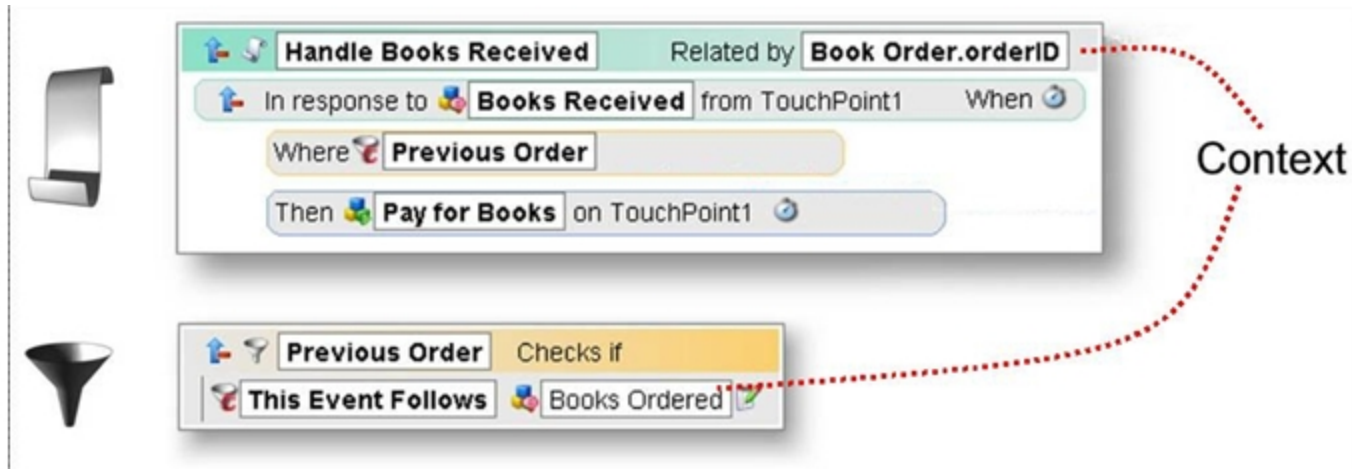
- Typically stateful over a (relatively short?) time-window

Runtime

- State persistence
- Event enrichment and transformation
- Event processing languages
- Continuous query engine
- Event Processing Network

Context

- Messages are correlated using a “related-by” clause.



Reuse

- Rules are tricky to reuse across event applications because they have significant implicit pre-conditions.

DSL

- Write:
 - Filters (conditions)
 - Actions
 - Event Query Language statements

Human Interface

- None?

Dealing with Change

- Policy changes: Medium
- Model changes: Medium
- Requires:
 - Update filters/actions/queries
 - Update test data

How to Combine?

- 1) Decisions | Events | BPM
- 2) Decisions + Events | BPM
- 3) Decisions + BPM | Events
- 4) Events + BPM | Decisions
- 5) Events + Decisions + BPM

Observe,
Orient,
Decide, Act.
- or -
Detect,
Decide, Do.

Users and their roles.
Artifact life-cycles.
Governance...



Decisions | Events | BPM

- Three peer systems
- Separate change cycles
- Connectivity based on SOA patterns
- Have to deal with installation and management complexity
- Performance?

Decisions + Events | BPM

- Unified decision and event lifecycle, connectivity to/from BPM
- Shared language infrastructure between decisions and events
- Runtime convergence?
- Ability to share/reuse rules across decisions and events?

Decisions + BPM | Events

- BPM and decision cycles are merged, connectivity to events.
- IBM implemented this for simple rules in IBM BPM 7.5
- Simplifies tooling for BPM centric projects

Events + BPM | Decisions

- Highly dynamic BPM that combines declarative BPMN diagrams with event-based dispatch
- I've not seen this pattern yet but it may be useful where BPMN fails to efficiently capture the required transitions.

Events + Decisions + BPM

- Cycles completely merged
- Original concept for Drools Fusion?
- Processes implemented using something like RETE as runtime?
- Architecturally attractive but problematic in terms of stakeholders and change management etc.

Conclusion

- 3 separate communities, each with strong opinions and a fear of being subsumed!
- Users want to stick with familiar modeling and DSLs. Frustrated by overlap in tools and lack of guidance.
- Widely different objectives in terms of simulation, scale, performance and governance
- Most common patterns?
 - Decisions + Events | BPM
 - Rule language, runtime and lifecycle sharing
 - Decisions | Events | BPM
 - Best-of-breed
- Sorry, I do not have a “silver bullet” solution...