Rules Fest 2010

International Conference on Reasoning Technologies

October 11-14 • San Jose, CA USA

Grindwork Corporation
Intelligent Automation

IBM

JBoss by Red Hat

morris technical solutions

VisionArts communications

A Simple Inference Framework for Connecting the Dots

Jacob Feldman, PhD

OpenRules, Inc.

Cork Constraint Computation Centre

www.openrules.com

www.4c.ucc.ie

Motivation

- **January 8, 2010. Tom Davenport about "Connect the dots":**

  "Everybody, including President Obama, is criticizing the U.S. intelligence agencies for not keeping accused underwear bomber Umar Farouk Abdulmutallab off the Christmas Day flight from Amsterdam to Detroit. Why didn't they "connect the dots" or "put the pieces together"?

- But is this really a fair criticism?

  "Just how easy is it to connect the dots? Granted, there were numerous indications of Abdulmutallab's evil intent. But it would have been difficult to put them together before the flight. Combining disparate pieces of information about people — *whether they are customers or terrorists* — is akin to solving a complex jigsaw puzzle."

http://blogs.hbr.org/davenport/2010/01/why_they_didnt_connect_the_dot.html

Tom Davenport: “Connect the dots" Solution

- “If you doubt that this is hard and you come from a corporate setting, ask yourself how often some of your best customers have slipped through the cracks of your information and knowledge systems. Or if you're a consumer, how often do companies connect the dots on your own relationship with them? And I'm guessing you don't even have evil intent toward those companies!”

- A remedy?

  “Perhaps the only palatable remedy would be an intelligence community that views **high-quality information and knowledge management** as its primary job. If I were Barack Obama, that's the approach I would be viewing as **the real solution to the ‘connect the dots’ problem**”

A simple Framework for “Connecting the Dots” - CONDOTS

- In this presentation we introduce a *simple yet practical inference framework* for the creation and continuing development of various “Connecting the Dots” systems
- At the heart of the framework is an “*always running*” inference engine that:
  - can accept new facts
  - propagate them through the existing knowledge base
  - solicit new facts if necessary
  - and, finally, reach a conclusion by connecting all the facts together
- The framework does not invent a new “magic” technology but rather integrates well-proven techniques and expert knowledge in an ingenious manner
- Key differentiator: this framework allows subject matter experts (non-programmers) to quickly incorporate new terms, facts, and supporting processing rules into a perpetually running system

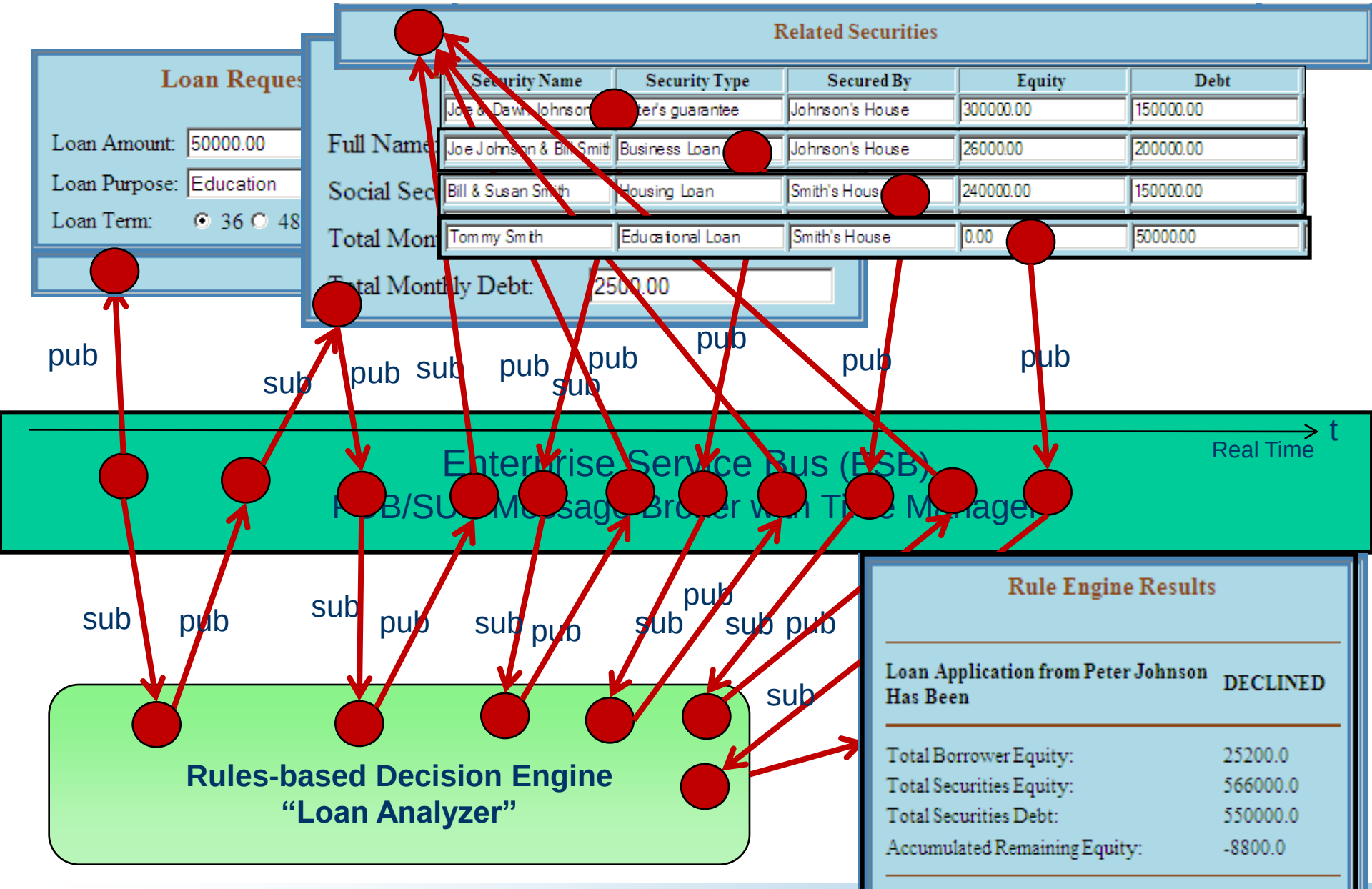
Other Examples of "Connecting the Dots" Scenarios

- Complex Loan Approval Process with Dynamically Discovered Facts (will be used for the framework demonstration)
- Identifying Suspicious Groups of Airplane Passengers
- Maintenance of User Profiles for Investment Portfolio Balancing
- Name your own scenario
- Common Features:
  - New facts come from different sources in different times
  - New Facts require reconsideration of all previously analyzed facts!

Scenario: Loan Approval with Dynamically Discovered New Facts

#	Events and Facts	Peter Johnson's Loan Application Decision Process	Additional Analysis
1.	Peter Johnson applied for $50K educational loan. *Analysis shows: Insufficient Income*	**Declined**	**But** a bank manager found that their valuable client with the same address can be a guarantor
2.	Joe & Dawn Johnson agreed to be Peter's guarantors. They have a Housing Loan with Available Equity = $300K and Remaining Debt = $150K *Analysis shows: $125,200K surplus*	**Approved**	**But** Conducting more detailed analysis, the manager notices a joint borrowing on Mr Johnson file which is not with his wife
3.	Joe Johnson and his partner Bill Smith (50/50) have a Business Loan for $200K with Available Equity $52K. *Analysis shows: Accumulated remaining equity is ($48,800)*	**Declined**	**But** Bill and Susan Smith have other facilities outstanding against their property as well as the business loan
4.	Bill & Susan Smith have a Housing Loan with Available Equity = $240K and Remaining Debt = $150K *Analysis shows: Still a surplus $41,200*	**Approved**	**But** a lending clerk at the lending operations center while preparing the collateral documentation, noticed a secured personal loan in the name of Tommy Smith for $50K secured by her parents
5.	Their son Tommy Smith has $50K loan secured by her parents *Analysis shows: Available Equity ($8,800)*	**Declined**	The business debt would be $8,800 short on cover

Loan Approval Process



Live Demo with OpenRules Forms and State Machines

OPEN RULES

Loan Approval with Dynamically Defined Securities

CONNECT The Dots

Loan Request

Full Name:	Peter Johnson
Loan Amount:	50000
Loan Purpose:	Education
Loan Term:	(•) 36 ( ) 48 ( ) 60 ( ) 72
Total Monthly Income:	4000
Total Monthly Debt:	2500

New Loan Request | Modify Loan Request

Related Securities

Security Name	Security Type	Secured By	Equity	Debt
Peter Johnson	Peter's Income&Debt	Salary	115200	140000
Joe & Dawn Johnson	Peter's guarantee	Johnson's House	300000	150000
Joe Johnson & Bill Smitl	Business Loan	Johnson's House	26000	200000
Bill & Susan Smith	Housing Loan	Smith's House	240000	150000
Tommy Smith	Educational Loan	Smith's House	0	50000

New Related Fact | Finilize

Loan Evaluation Results:

DECLINED

Explanations

Securities	Equity	Debt	Accumulated Equity
Peter Johnson	115200	140000	-24800
Joe & Dawn Johnson	300000	150000	125200
Joe Johnson & Bill Smitl	26000	200000	-48800
Bill & Susan Smith	240000	150000	41200
Tommy Smith	0	50000	-8800

Loan Analysis Finilized

OpenRules, Inc.

FSM “Loan State Machine”

runLoanAnalyzer()

Rules String **runLoanStateMachine**(String event)					
Condition1	Condition2	Action1	Action2	Action4	Action3
loan.state.equals(state)	event.equals(inEvent)	loan.state = state;	runLoanAnalyzer();	loan.waitingDays = days;	dialog().status = outEvent; return outEvent;
String state	String inEvent	String state	String x	int days	String outEvent
Current Loan State	**Subscribed Event**	**Transfer Loan to State**	**Run Loan Analyzer**	**Set Waiting Time (days)**	**Publish New Event**
New	New Loan Request	Waiting for Related Facts	X	3	Investigate Related Securities
Waiting for Related Facts	Modify Loan Request	Waiting for Related Facts	X	7	
Waiting for Related Facts	New Related Fact	Waiting for Related Facts	X	7	
Waiting for Related Facts	Related Fact Modified	Waiting for Related Facts	X	5	
Waiting for Related Facts	Waiting Time Expired	Finilized	X		Loan Analysis Completed (Investigation Time Expired)
Waiting for Related Facts	Finilize	Finilized	X		Loan Analysis Finilized
Finalized	Modify Loan Request	Waiting for Related Facts	X	7	Investigate Related Securities

Loan Analyzer

- Defined in Excel
- Invoked from the Loan State Machine
- Calculates and analyses Accumulated Equity across All known securities

Method String **runLoanAnalyzer()**

```
loan.accumulatedEquity = 0;
for(int i =0; i < loan.numberOfSecurities; ++i) {
  Security s = relatedSecurities[i];
  loan.accumulatedEquity += (s.equity - s.debt);
}
return loanApprovalRules();
```

Simple Rules for Equity and Debt Calculation

Rules void calculateBorrowerEquityDebt()	
Set Borrower Equity	**Set Borrower Debt**
:= loan.monthlyIncome * 0.8 * loan.loanTerm	:= loan.monthlyDebt * loan.loanTerm + loan.loanAmount

Rules String **loanApprovalRules()**	
If **Accumulared Equity**	**Then** **Set Loan Status**
< 0	DECLINED
>= 0	APPROVED

Defining Data Types and Data Facts in Excel

Datatype Loan	
String	borrower
int	loanAmount
int	loanTerm
String	purpose
int	monthlyIncome
int	monthlyDebt
String	state
String	status
String	result
int	waitingDays
int	numberOfSecurities
int	accumulatedEquity

Datatype Security	
String	id
String	type
String	securedBy
int	equity
int	debt

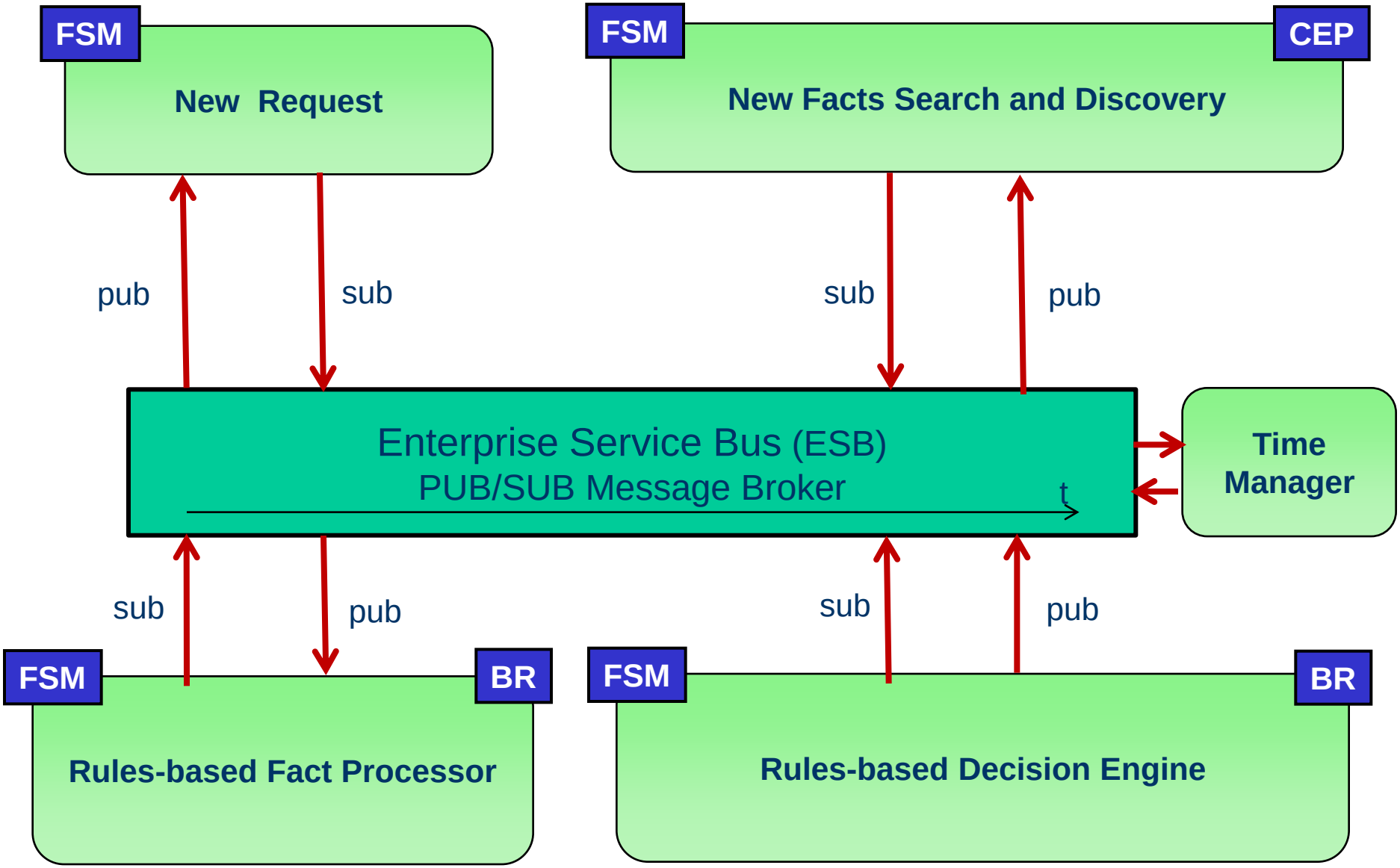
Variable Loan loan						
Borrower	**Loan Amount**	**Loan Term**	**Purpose**	**Monthly Income**	**Monthly Debt**	**State**
Peter Johnson	50000	36	Education	4000	2500	New

Data Security relatedSecurities				
Security Name	**Security Type**	**Secured By**	**Equity**	**Debt**
Peter Johnson	Peter's Income&Debt	Salary	0	0
Joe & Dawn Johnson	Peter's guarantee	Johnson's House	300000	150000
Joe Johnson & Bill Smith	Business Loan	Johnson's House	26000	200000
Bill & Susan Smith	Housing Loan	Smith's House	240000	150000
Tommy Smith	Educational Loan	Smith's House	0	50000

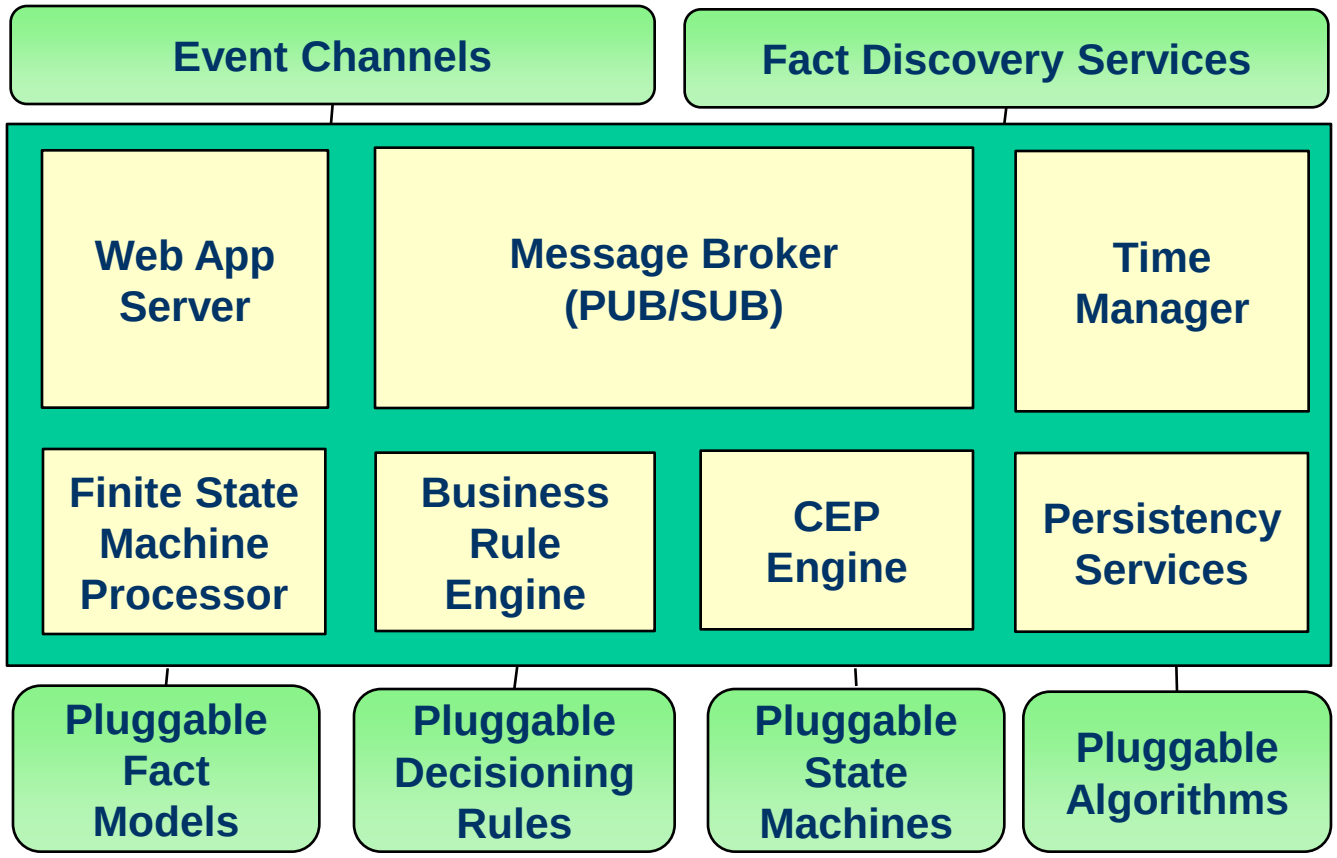
A Simple Inference Framework for Connecting the Dots "CONDOTS"

- **Common Components:**
  - Message Broker with a Time Manager (Apache ActiveMQ)
  - Web App Server (Apache Tomcat)
  - Business Rules Repository (OpenRules)
  - Decision Engine (OpenRules)
  - Finite State Machines (OpenRules FSM)
  - Web-based Questionnaire Builder (such as OpenRules Dialog "ORD")
- **Problem Specific Components:**
  - Business Object Model (OpenRules Data Types or Java)
  - Adding New Event Types without coding
  - Adding New State Machines without coding
  - Adding New Decisioning Rules

Functional Scheme for Connecting The Dots systems

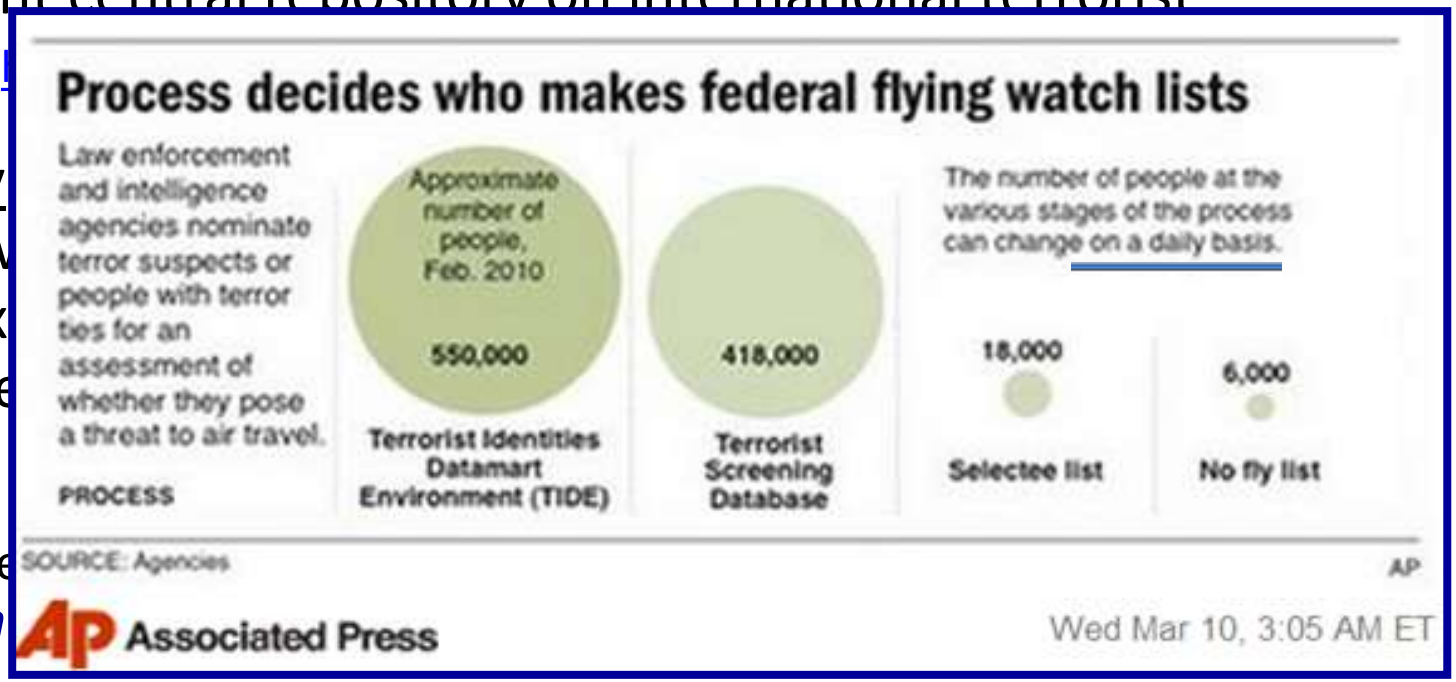


Architecture



Scenario: Back to the underwear bomber case

- TIDE - The Terrorist Identities Datamart Environment is the US Government central repository on international terrorist identities (
- "Every day their review analysts ex consolidate *this list* .
- This database List - *Abdulm*



- *Why? A guess: the fact "One way air ticket" was not connected to the fact "Is in the TIDE" - the proper "engine" was expected to run later that day.*
- *Obvious conclusion: These lists should be maintained by always running inference engine ("on a daily basis" is not enough!)*

Scenario: Identifying Suspicious Groups of Airplane Passengers

- A system validates a list of all passengers when they book tickets for air travel. Along with simple criteria such as:

  age range, gender, country of origin, legal status, ticket type, etc.

  the system may include *dynamic* characteristics such as:

  acquired certain chemical products in certain quantities,

  took certain classes at a certain educational institutions during certain time periods,

  visited certain countries during the last 3 years, 6 months, etc.

- Dynamic attributes need to be validated not just for one passenger but also for *all possible combinations of currently known passengers*

- The very fact that a passenger satisfies a certain criterion, may initiate a new request about other passengers, that can in turn initiates additional new requests and forces the system to re-evaluate already known facts

Scenario: Maintenance of User Profiles for Portfolio Balancing

- A customer may define preferences related to his/her investment strategy (conservative or moderate risk level, industry sectors, security type distributions, etc.).
- However, the dynamic nature of the constantly changing financial market requires permanent automatic and interactive adjustments to each customer's profile
- For example, a system should be able to generate questions like: *"Your positions are overly concentrated in a single market segment. Are you willing to relax position constraints?"* and make an automatic decision in each case based on a customer's preferences and the company's latest investment strategy

Crucial Functionality

- What all these scenarios have in common?
  - New facts arrive from different sources and in different times
  - New facts require immediate re-evaluation of previously analyzed facts
- What is crucial to make the described architecture work in a real-world applications?
  - An ability to add new (previously unknown) terms, facts, states, and proper processing rules on the fly (constantly enriched knowledgebase)
  - Direct involvement of subject matter experts in ongoing improvements

Further R&D Needed

- **Dealing with uncertainty**
  - Attach a “level of confidence” to facts and results
  - Rules may deal not with hard thresholds but with approximate intervals
  - Use constraint programming experience of finding solutions in uncertain situations
- **Dealing with relationships between multiple instances of the same type,** e.g. multiple passengers on the same flight
- **Fact Discovery**
  - Use of CEP
  - Use of Search Engines
  - Automatic Question Generation
  - Integration with Semantic Web (inter-ontology relationships)
- **More?**

Summary

- **CONDOTS is an experimental inference framework for creating custom "Connecting the Dots" systems**
- **Uses commonly available components:**
  - **ESB** with a Message Broker (JMS Implementation)
  - **BRMS** - Maintains Business Rules and Executes Decision Engine
  - **FSM** – Maintains Finite State Machines
  - **GUI Development**
  - Optional:
    - **Questionnaire Builder** (such as OpenRules ORD)
    - **CEP Engine**
    - **Search Engine**
- **Orientation to Subject Matter Experts with an ability to add new terms, facts, states, and processing rules on the fly**

QnA

jacobfeldman@openrules.com (+1 732 993 3131)

j.feldman@4c.ucc.ie (+353 21 4205966)