



Business Rules Management System:

Gerenciando Regras de Negócio

Edson Tirelli

Sr. Software Engineer

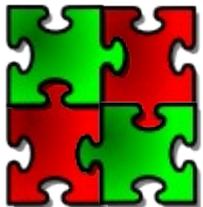




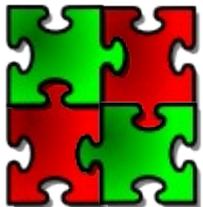
Sistema Corporativo



Décadas de 50-60: sistemas monolíticos



Décadas de 60-70: separação dos dados



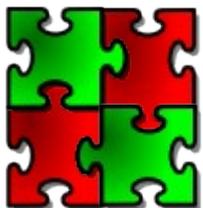
Décadas de 70-80: procedimentos reutilizáveis



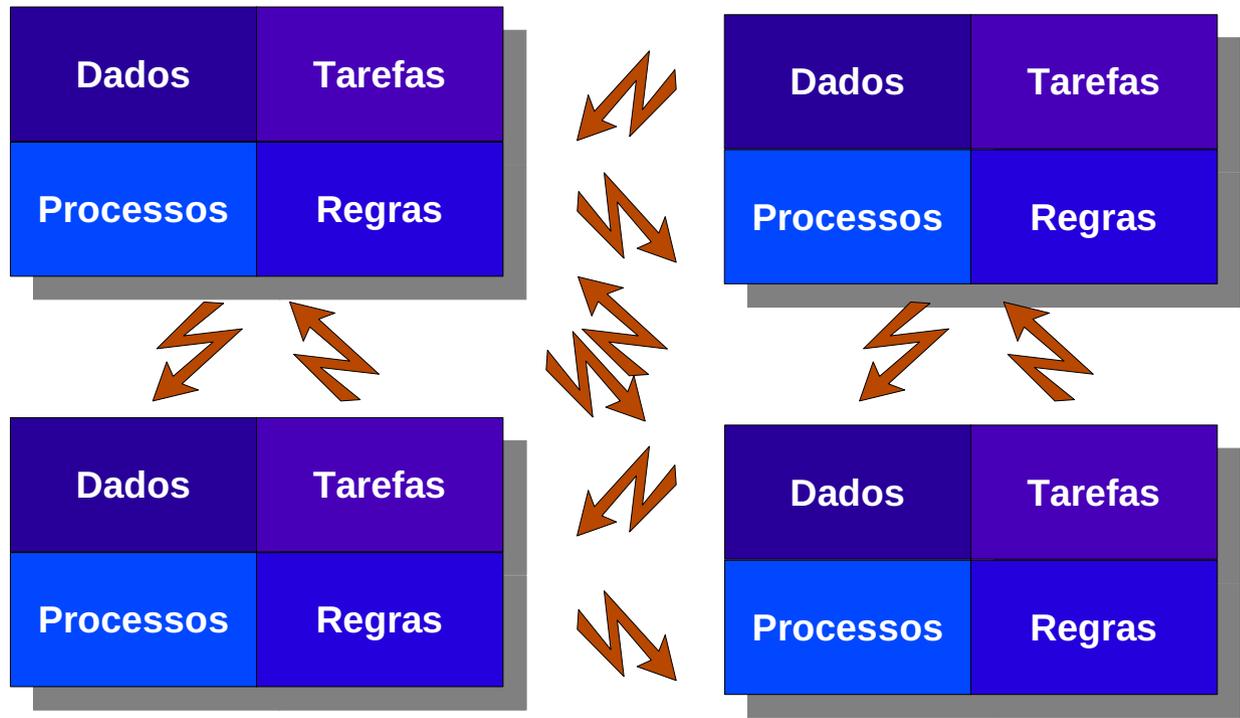
A partir de 90: separação dos processos e regras



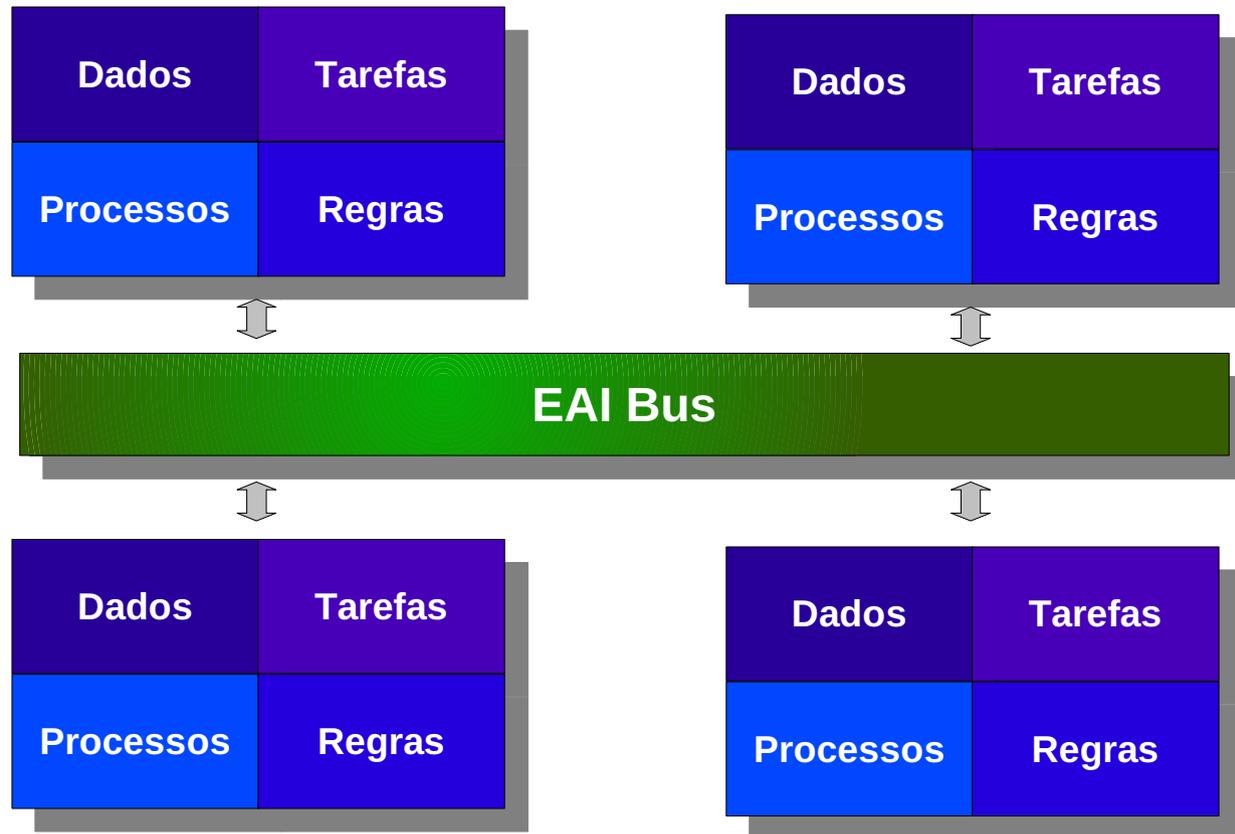
A partir de 90: separação dos processos e regras



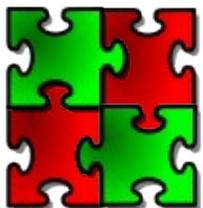
Sistemas Isolados = custo

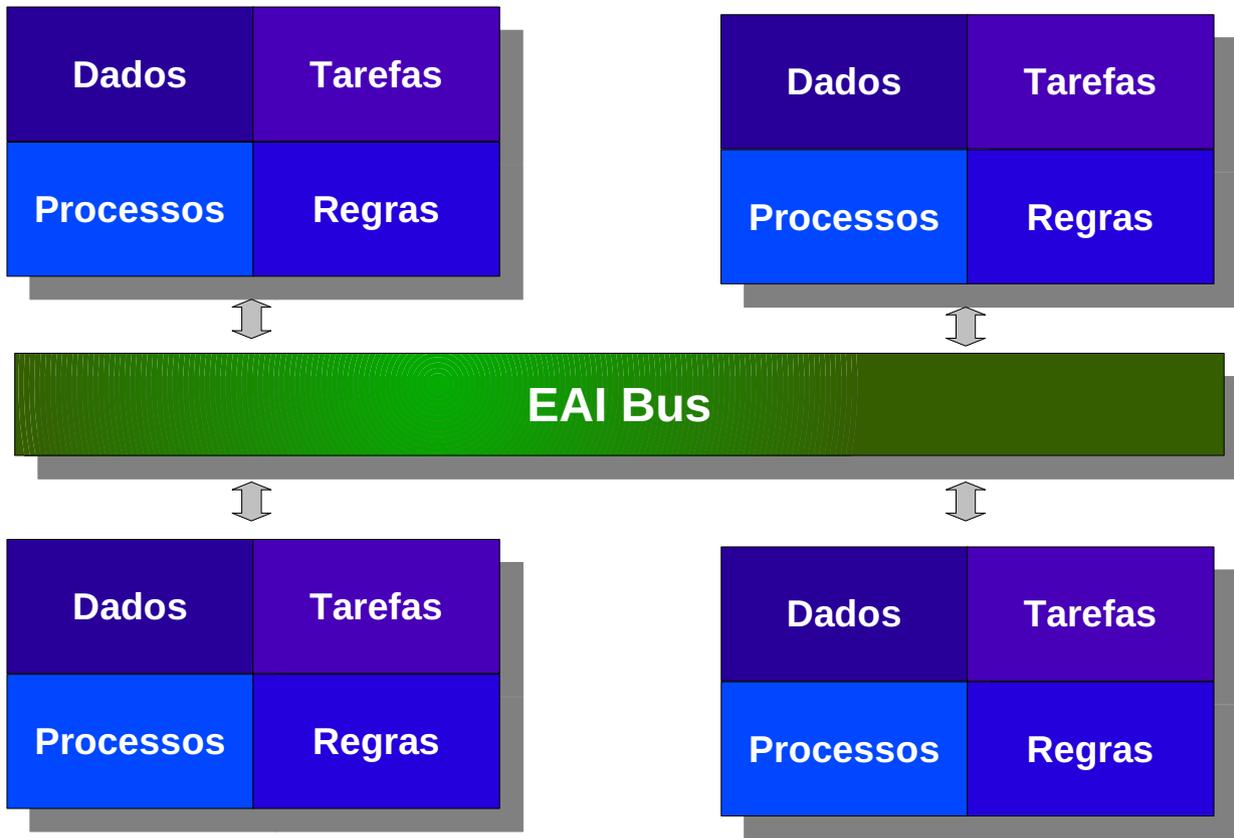


Integração ponto-a-ponto = mais custos



EAI: Solução?





Replicação de Regras e Processos

Falta de Padronização

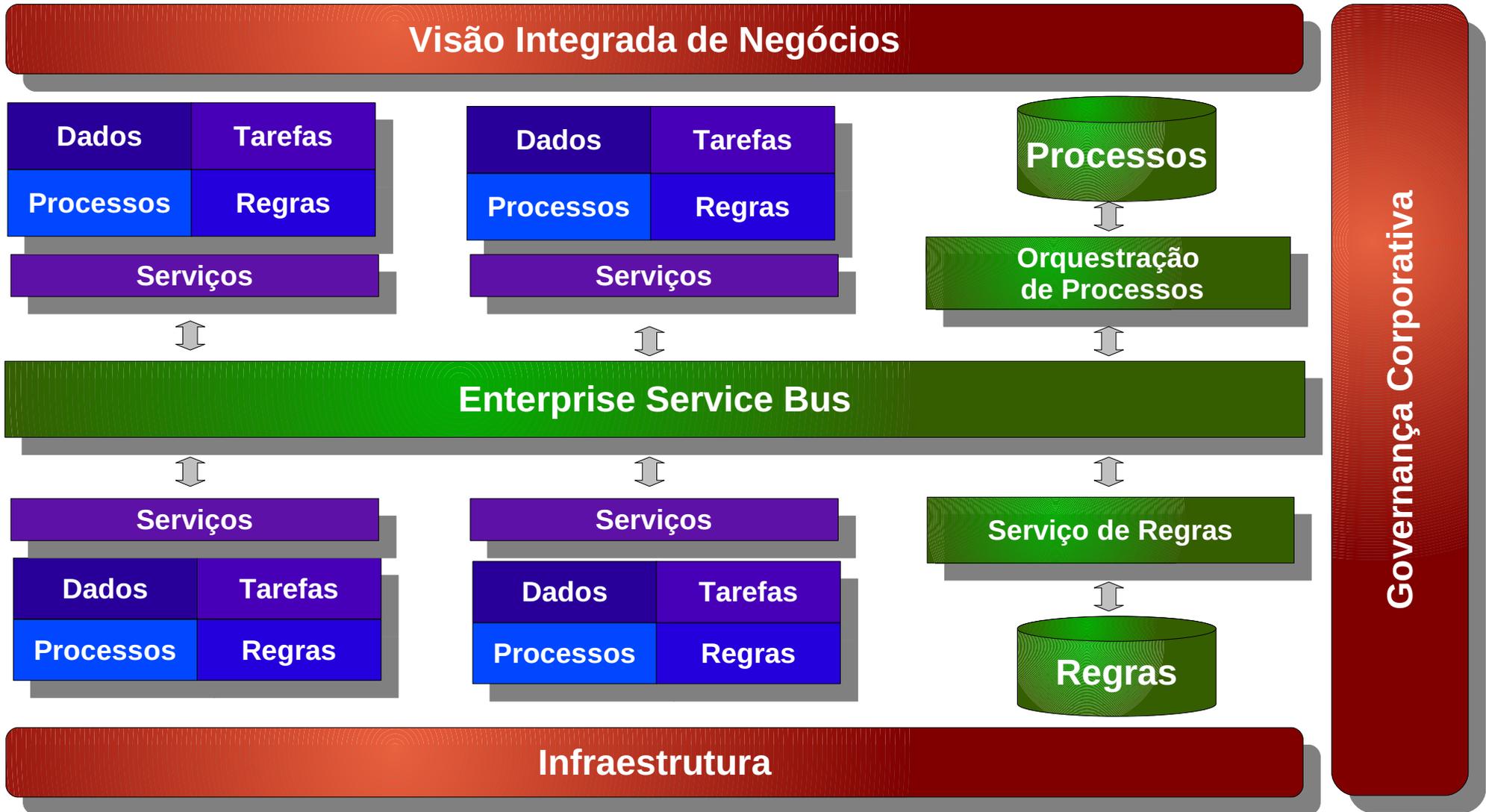
Dependência intersistêmica

Ausência de Métricas

Ausência de Governança



EAI: Solução? Não exatamente



Maximizando o reuso e a integração

■ Regra:

- Conjunto de **condições** a serem avaliadas e uma lista de **ações** a serem executadas (consequência) caso as condições sejam verdadeiras.

■ Fatos:

- **Dados** sobre os quais as regras são aplicadas.

■ Origens das regras:

- **Regulamentação legal:** “**Se** o tempo de uma **chamada telefônica** celular for inferior a 30 segundos, **então** cobre 30 segundos.”
- **Políticas da empresa:** “**Se** a **compra** for acima de R\$ 100,00, **então** aplique 10% de desconto.”
- **Conhecimento de especialistas:** “**Se** a pressão da **caldeira** estiver acima de 'n' vezes a temperatura, **então** inicie o procedimento de despressurização.”

■ **Dados:**

- DBMS: Sistemas de Bancos de Dados (Relacional, OO, Hierárquico, etc)

■ **Tarefas:**

- Linguagens Imperativas (C/C++, Java, C#, Python, etc)

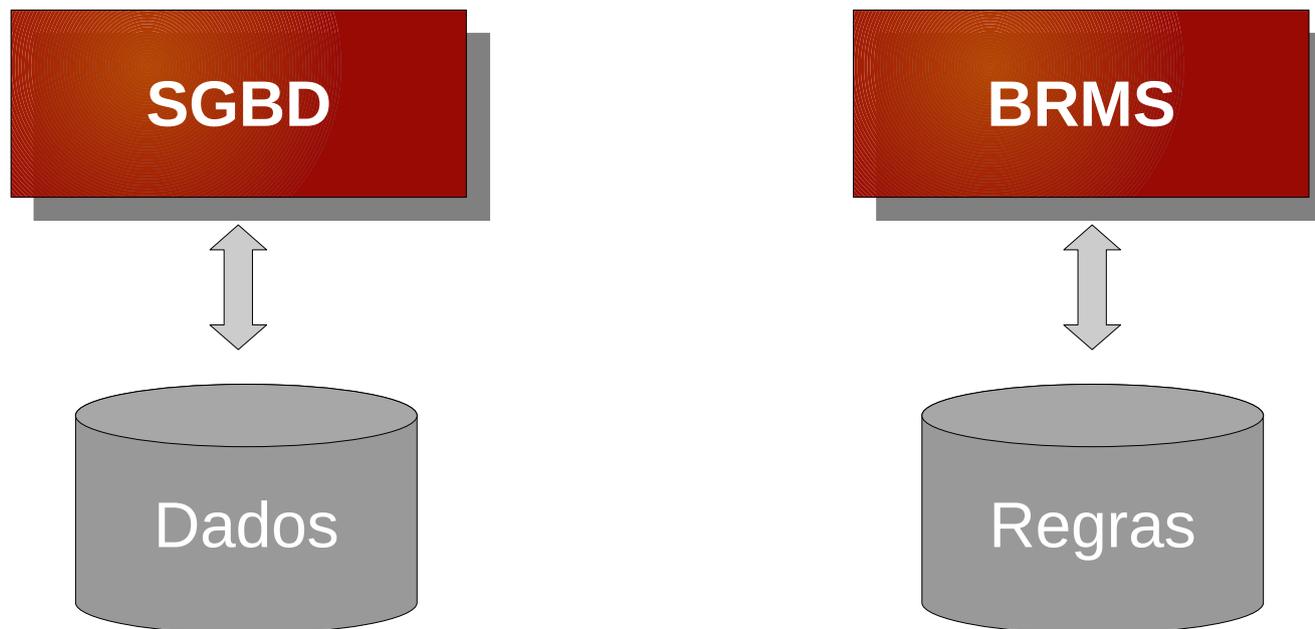
■ **Processos:**

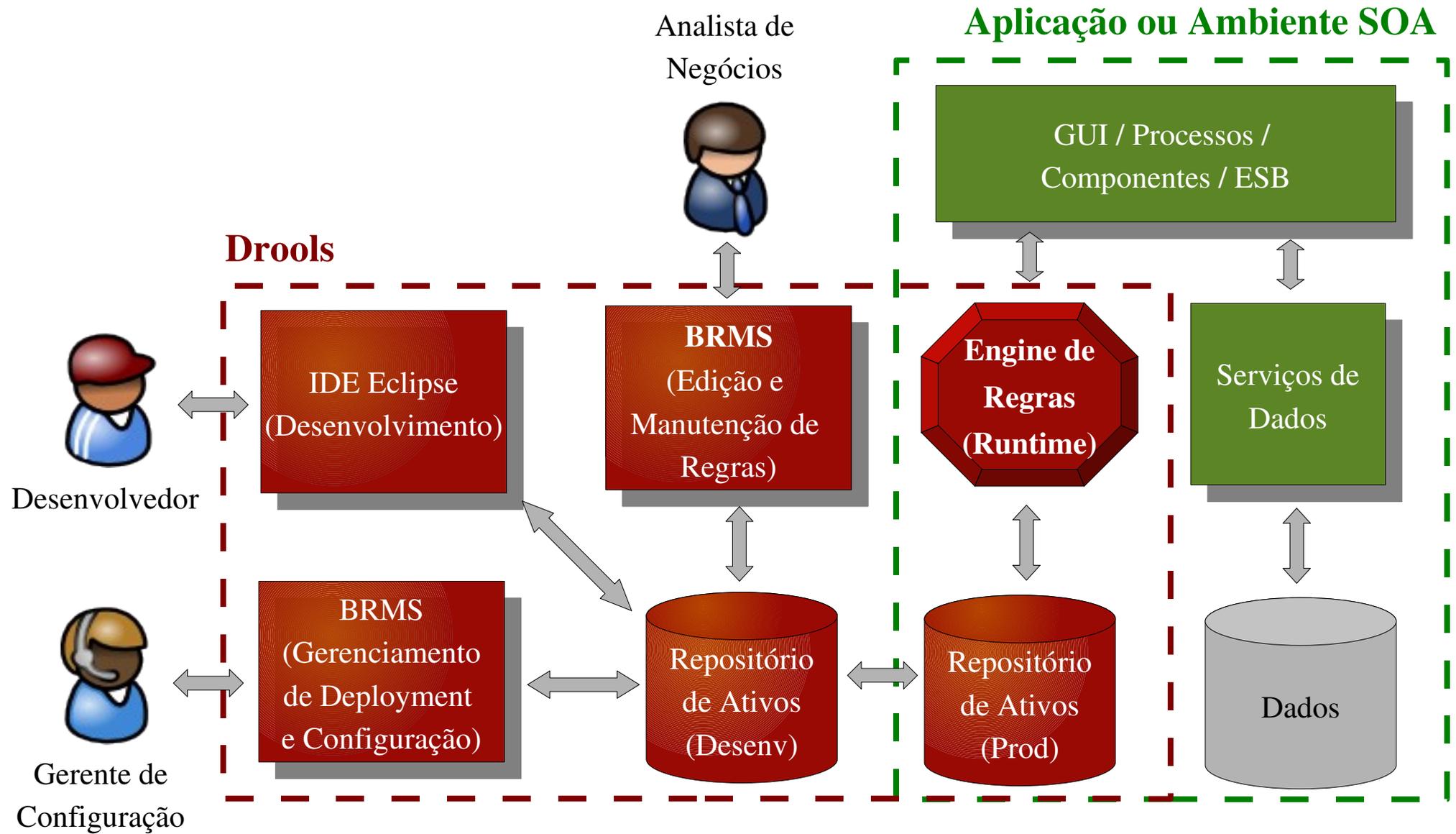
- BPMS: Engines de Processos (jBPM, WS Process Server, WLI, etc)

■ **Regras:**

- BRMS: Engines de Regras (Drools, CLIPS, iLOG JRules, Jess, etc)

- Componente de software especializado no gerenciamento e processamento de regras.
- Um BRMS está para as regras como um SGBD está para os dados.





- **Repositório de Ativos de Conhecimento**
 - Regras
 - Modelos
 - Fluxos
 - DSLs
- **Categorização e versionamento de ativos**
- **Gerenciamento de ciclo de vida**
- **Editores de Regras amigáveis para não técnicos**

Drools Insurance Company

Driver's information

Genre:

Birth Date: > (mm/dd/yyyy)

License Age: Years

Marital State:

Has Child?:

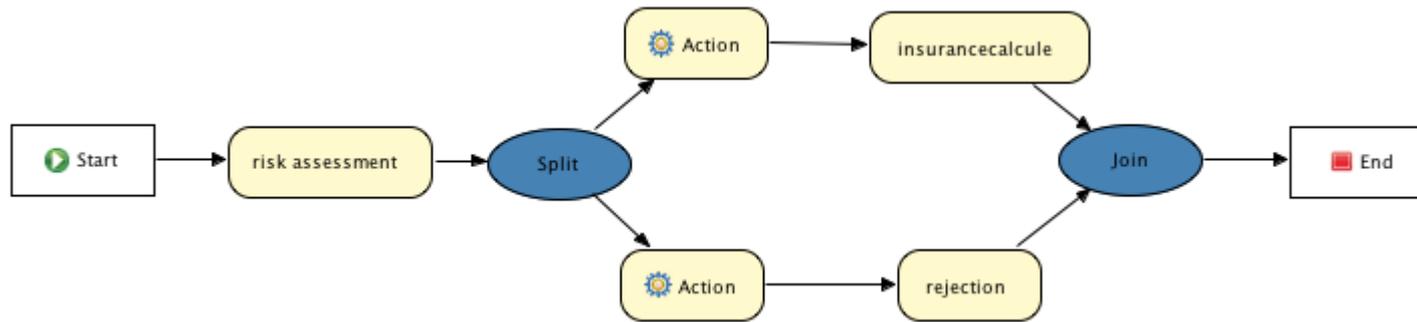
Degree:

Driver's additional info

Day vehicle place:

Night vehicle place:

Residence Status:



■ Telecom

- Qualificação de chamadas
- Tarifação de chamadas
- Detecção de fraudes

■ Seguros

- Classificação de Risco
- Cálculo do Prêmio

■ Mercado Financeiro

- Asset Management: Compliance
- Autorização de crédito

■ Segurança

- Controle e autorização de acesso
- Policy enforcement

■ Sistemas especialistas

- Construção civil
- Medicina

■ Engine de Regras:

- Forward Chaining
- Algoritmo ReteOO
- Otimizações: indexação de memórias Beta, hashing de nós Alfa, etc
- In-memory working memory

■ Arquitetura e Integração

- 100% Java, roda tanto em JSE quanto JEE
- Acesso transparente e direto ao modelo de domínio POJO
- Ferramenta de middleware, integrável à containers JEE, Aplicações Web, WebServices, jBPM, Seam, LDAP
- JSR94 compliant

■ Implementação das regras:

- Implementação declarativa
- Evita que as regras fiquem embutidas no código
- Regras podem ser alteradas sem a necessidade de recompilação
- Otimiza a execução de conjuntos de regras altamente complexos

■ Formas de definição das regras:

- DRL (Drools Rule Language): linguagem “nativa” similar ao java
- DSL (Domain Specific Language): linguagem de alto nível específica ao domínio
- BRX: Guided Editors (Eclipse e Web)
- Excel / OpenOffice: planilhas de tabelas de decisão
- XML: documentos XML com a declaração das regras
- API: para acesso direto via código Java

Debug - StateExampleUsingSalienc. drl - Eclipse SDK

File Edit Navigate Search Project Run Window Help

100%

Debug

StateExampleUsingSalienc [Drools Application]

- org.drools.examples.StateExampleUsingSalienc at localhost:4861
- Thread [main] (Suspended (breakpoint at line 8 in Rule_A_to_B_0))
 - Rule_A_to_B_0.consequence(KnowledgeHelper, State, FactHandle) line: 21
 - Rule_A_to_B_0ConsequenceInvoker.evaluate(KnowledgeHelper, WorkingMemory) line: 22
 - DefaultAgenda.fireActivation(Activation) line: not available
 - DefaultAgenda.fireNextItem(AgendaFilter) line: not available
 - ReteooWorkingMemory(AbtractWorkingMemory).fireAllRules(AgendaFilter) line: not available
 - ReteooWorkingMemory(AbtractWorkingMemory).fireAllRules() line: not available
 - StateExampleUsingSalienc.main(String[]) line: 47

Variables

Name	Value
b	State (id=1268)
changes	PropertyChangeSupport (id=1297)
name	"B"
state	1

StateExampleUsingSalienc.drl

```

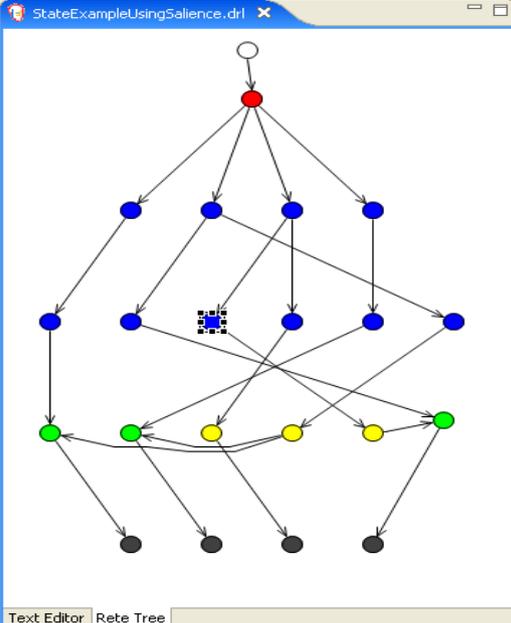
import org.drools.examples.State;

rule Bootstrap
when
    a : State(name == "A", state == State.NOTRUN )
then
    System.out.println(a.getName() + " finished" );
    a.setState( State.FINISHED );
end

rule "A to B"
when
    State(name == "A", state == State.FINISHED )
    b : State(name == "B", state == State.NOTRUN )
then
    b.setState( State.FINISHED );
    System.out.println(b.getName() + " finished" );
end

rule "B to C"
salience 10
when
    State(name == "B", state == State.FINISHED )
    c : State(name == "C", state == State.NOTRUN )
then
    System.out.println(c.getName() + " finished" );
end
    
```

StateExampleUsingSalienc.drl



Properties

Property	Value
Constraint	[LiteralConstraint fieldExtr...
Evaluator	Integer ==
Field Name	state
Name	Alpha BaseVertex
Value	1

Outline

- org.drools.examples
 - A to B
 - B to C
 - B to D
 - Bootstrap
 - org.drools.examples.State

Text Editor | Rete Tree

Text Editor | Rete Tree

Global Data View

The selected working memory has no globals defined.

A finished

Audit View

- Object asserted (1): A[NOTRUN]
- Activation created: Rule Bootstrap a=A[NOTRUN](1)
- Object asserted (2): B[NOTRUN]
- Object asserted (3): C[NOTRUN]
- Object asserted (4): D[NOTRUN]
- Activation executed: Rule Bootstrap a=A[NOTRUN](1)
 - Object modified (1): A[FINISHED]
 - Activation created: Rule A to B b=B[NOTRUN](2)
- Activation executed: Rule A to B b=B[NOTRUN](2)
 - Object modified (2): B[FINISHED]
 - Activation created: Rule B to C c=C[NOTRUN](3)
 - Activation created: Rule B to D d=D[NOTRUN](4)
- Activation executed: Rule B to C c=C[NOTRUN](3)
 - Object modified (3): C[FINISHED]
- Activation executed: Rule B to D d=D[NOTRUN](4)
 - Object modified (4): D[FINISHED]

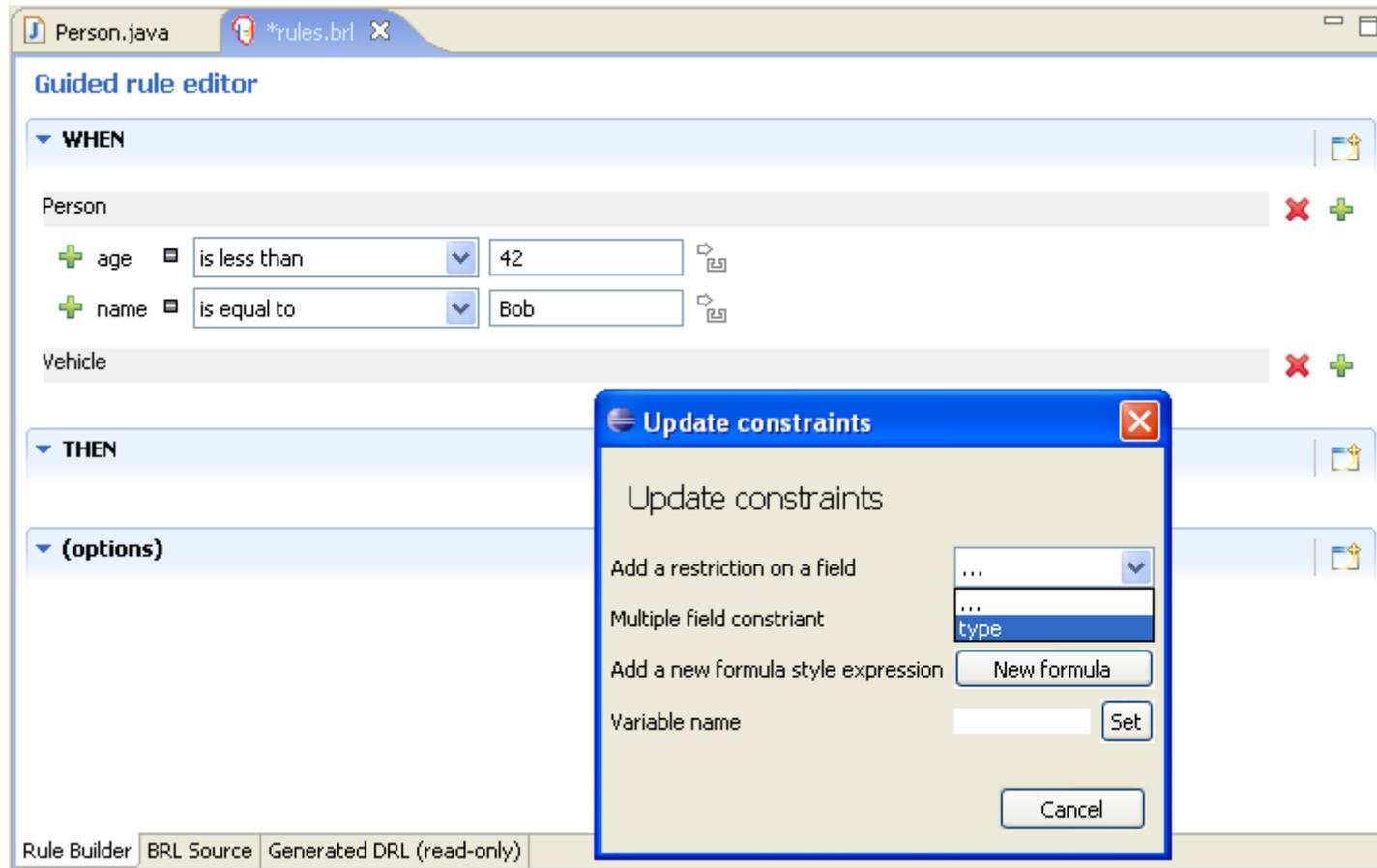
Agenda View

- MAIN[focus]= AgendaGroupImpl (id=1259)
 - [0]= AgendaItem (id=1262)
 - ruleName= "B to C"
 - c= State (id=1269)
 - [1]= AgendaItem (id=1263)
 - ruleName= "B to D"
 - d= State (id=1270)

Working Memory View

- [0]= State (id=1268)
- [1]= State (id=1269)
 - FINISHED= 1
 - NOTRUN= 0
 - changes= PropertyChangeSupport (id=1294)
 - name= "C"
 - state= 0
- [2]= State (id=1270)
- [3]= State (id=1271)

Writable Insert 21 : 1



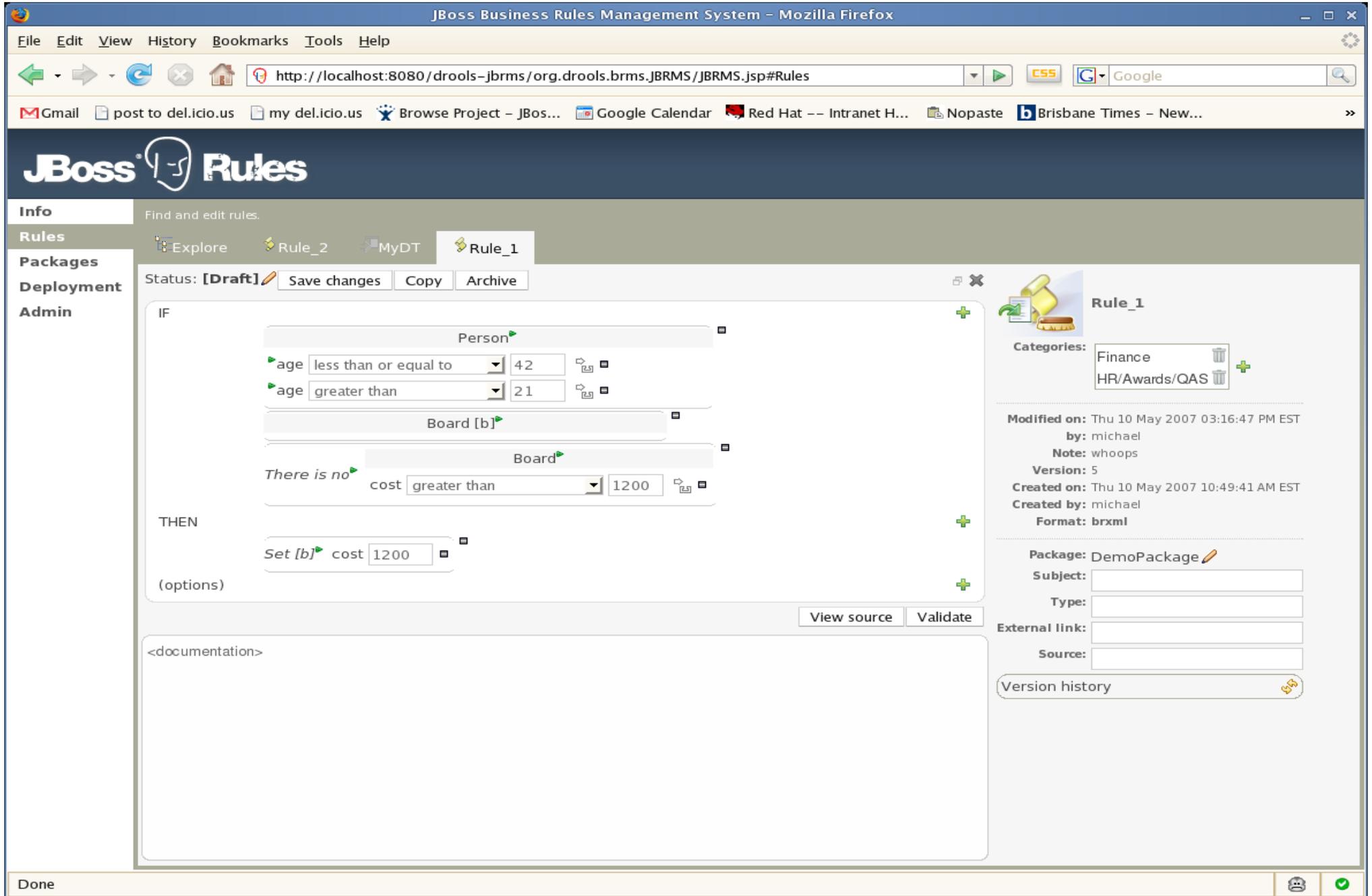
The screenshot shows the Eclipse IDE's Guided Rule Editor. The main window has tabs for 'Person.java' and '*rules.brl'. The editor is divided into sections: 'WHEN', 'THEN', and '(options)'. Under the 'WHEN' section, there are two constraints for the 'Person' object: 'age is less than 42' and 'name is equal to Bob'. A dialog box titled 'Update constraints' is open in the foreground, allowing the user to modify these constraints. The dialog includes a dropdown menu for selecting a field, a text input for a value, a 'New formula' button, and a 'Set' button for a variable name. The 'Generated DRL (read-only)' tab is visible at the bottom of the editor.

```
rule "Driver in unsafe area for marginal age"  
  when  
    Policy type is 'COMPREHENSIVE'  
    Driver is less than 25 years old  
    Driver has a location risk profile of 'HIGH'  
  then  
    <> Driver has a location risk profile of '{risk}'  
    <> Driver has an age of at least {age}  
    <> Driver has had more than {prior} prior claims  
  end  
rule "Driver unsafe for marginal age driver in high risk area"  
  when  
    <> Driver has had {number} prior claims  
    <> Driver is between {lower} and {upper} years old  
    <> Driver is greater than {age} years old  
    <> Driver is less than {age} years old  
    Policy type is 'MED'  
  then  
    Reject Policy with explanation : 'Driver in that area is too risky -  
  end  
rule "Driver unsafe for third party"  
  when  
    Policy type is 'THIRD_PARTY'  
    Driver has had more than 2 prior claims  
  ..  
  ..  
  ..
```

Text Editor | Rete Tree

Decision Tables (Excel/OpenOffice)

	B	C	D	E	F	G	H
1							
4							
9	Base pricing rules	Age Bracket	Location risk profile	Number of prior claims	Policy type applying for	Base \$ AUD	Record Reason
10	Young safe package	18, 24	LOW	1	COMPREHENSIVE	450	
11			MED		FIRE_THEFT	200	Priors not relevant
12			MED	0	COMPREHENSIVE	300	
13			LOW		FIRE_THEFT	150	
14			LOW	0	COMPREHENSIVE	150	Safe driver discount
15	Young risk	18,24	MED	1	COMPREHENSIVE	700	
16		18,24	HIGH	0	COMPREHENSIVE	700	Location risk
17		18,24	HIGH		FIRE_THEFT	550	Location risk
18	Mature drivers	25,30		0	COMPREHENSIVE	120	Cheapest possible
19		25,30		1	COMPREHENSIVE	300	
20		25,30		2	COMPREHENSIVE	590	
21		25,35		3	THIRD PARTY	800	High risk



The screenshot shows the JBoss Business Rules Management System (BRMS) interface in a Mozilla Firefox browser. The browser's address bar shows the URL: `http://localhost:8080/drools-jbrms/org.drools.brms.JBRMS/JBRMS.jsp#Rules`. The page title is "JBoss Rules".

The interface is divided into several sections:

- Info:** Find and edit rules.
- Rules:** Includes tabs for "Explore", "Rule_2", "MyDT", and "Rule_1".
- Packages, Deployment, Admin:** A vertical sidebar on the left.
- Rule Editor:** The main area for editing rules. It shows the status as "[Draft]" and buttons for "Save changes", "Copy", and "Archive".
- Rule Definition:** The rule is defined with an "IF" condition and a "THEN" action.
 - IF:** A "Person" object with "age" less than or equal to 42 and "age" greater than 21. A "Board [b]" object with "cost" greater than 1200. A note: "There is no Board".
 - THEN:** "Set [b] cost 1200".
 - (options):** An empty section.
- Metadata:** A sidebar on the right for "Rule_1" with fields for Categories (Finance, HR/Awards/QAS), Modified on (Thu 10 May 2007 03:16:47 PM EST), by (michael), Note (whoops), Version (5), Created on (Thu 10 May 2007 10:49:41 AM EST), Created by (michael), Format (brxml), Package (DemoPackage), Subject, Type, External link, and Source.
- Buttons:** "View source" and "Validate" buttons are located below the rule definition.
- Documentation:** A text area at the bottom of the rule editor containing the placeholder text "<documentation>".

The bottom status bar of the browser shows "Done" and a green checkmark icon.

The screenshot displays the Eclipse IDE interface for editing a Drools RuleFlow. The main window shows a flowchart for a number-guessing game. The flow starts with a 'Start' node, leading to a 'More guesses Join' node. This is followed by a 'Guess' node, then a decision node 'Guess correct?'. If correct, it leads to 'Guess Correct' and then 'No more guesses Join', which leads to 'End'. If incorrect, it branches into 'Too High' and 'Too Low', leading to an 'Incorrect guess' node, then 'Guess incorr...', and finally 'More Guesses?', which loops back to 'More guesses Join'. A 'Components' sidebar on the left lists various flow elements like 'Start', 'End', 'RuleFlowGroup', 'Split', 'Join', 'Milestone', 'SubFlow', and 'Action'. The right pane shows the corresponding DRL code for the ruleflow, including rules for recording highest/lowest guesses and retracting incorrect guesses. The bottom of the IDE shows the Package Explorer and Outline views.

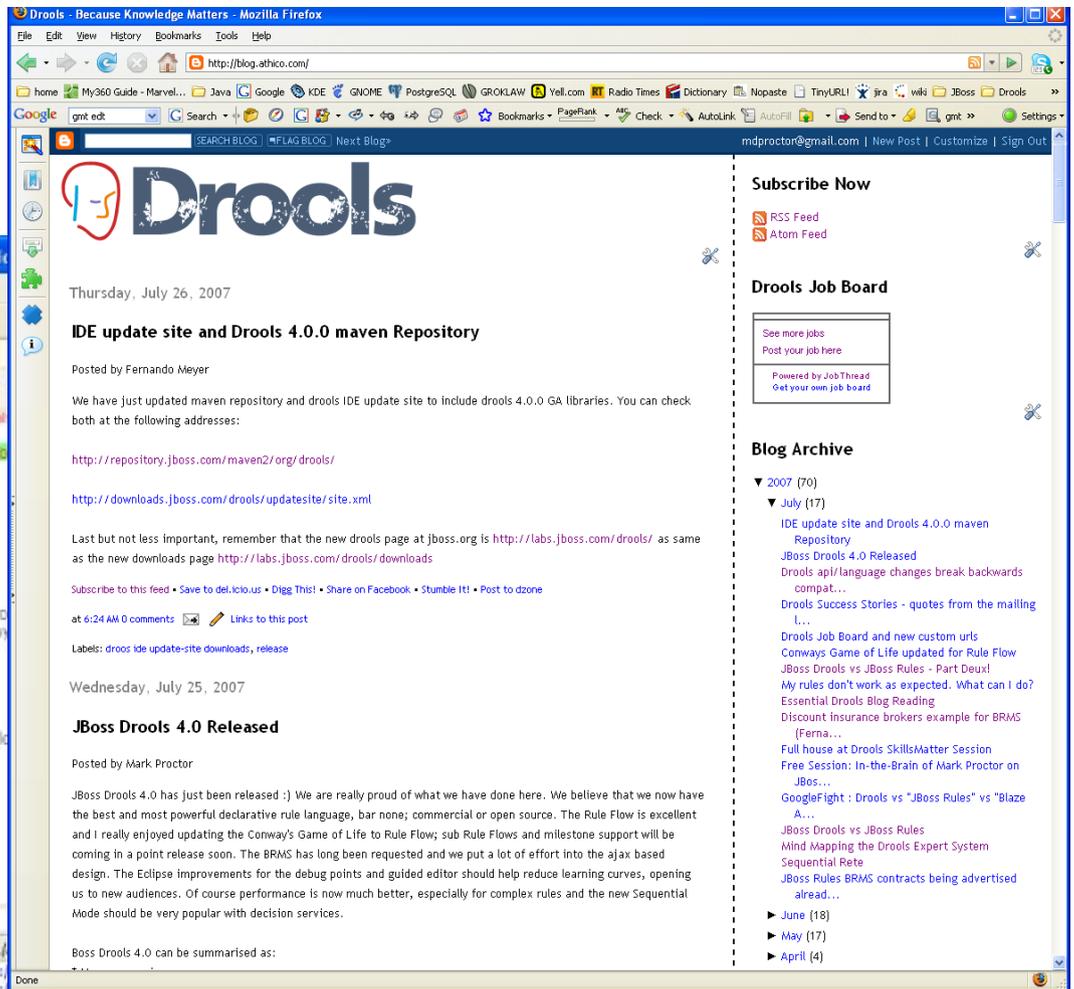
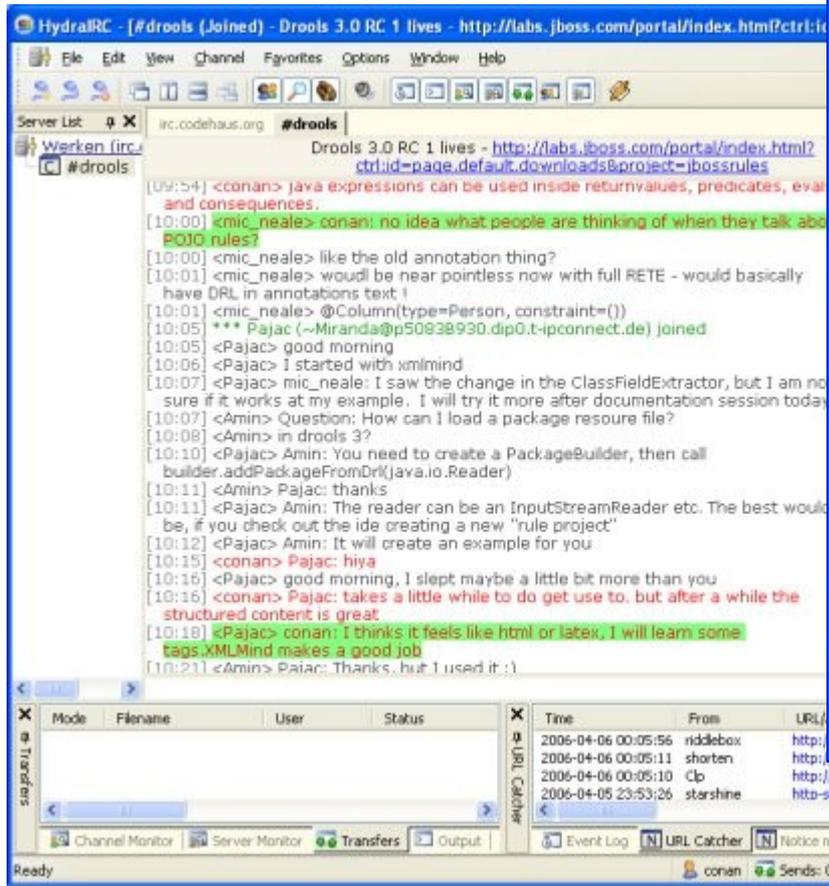
```
26     insert( new Guess( i ) );
27 end
28
29 rule "Record the highest Guess"
30     ruleflow-group "Too High"
31     no-loop
32     when
33         game : Game( biggestGuess : biggest )
34         Guess( $value : value > biggestGuess )
35     then
36         modify ( game ) { biggest = $value };
37     end
38
39 rule "Record the lowest Guess"
40     ruleflow-group "Too Low"
41     no-loop
42     when
43         Game( smallestGuess : smallest )
44         Guess( $value : value < smallestGuess )
45     then
46         modify ( game ) { smallest = $value };
47     end
48
49 rule "Guess incorrect, retract Guess"
50     ruleflow-group "Guess incorrect"
51     when
52         guess : Guess()
53     then
54         retract( guess );
55     end
56
57
58 rule "No more Guesses notification"
```

- ◆ Activation executed: Rule Start Clinical Pathway X if diagnosed d=Diagnose: Diagnose disease X: Type unknown(2)
 - Object removed (2): Diagnose: Diagnose disease X: Type unknown
 - ↳ Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-16-17
 - ↳ Activation cancelled: Rule Remove old diagnose d=Diagnose: Diagnose disease X: Type unknown(2)
 - ↳ Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-12
 - 🔗 RuleFlowGroup activated: Examinations[size=2]
 - 🔗 RuleFlow started: ClinicalPathwayX[org.drools.examples.cdss.ClinicalPathwayX]
- ◆ Activation executed: Rule Examination1
- ◆ Activation executed: Rule Examination2
- 🔗 RuleFlowGroup deactivated: Examinations[size=0]
- 🔗 RuleFlowGroup activated: AdditionalExaminations[size=2]
- Object inserted (2): Diagnose: Diagnose disease X: Type unknown
 - ⇒ Activation created: Rule Start Clinical Pathway X if diagnosed d=Diagnose: Diagnose disease X: Type unknown(2)
 - ⇒ Activation created: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-16-17
 - ⇒ Activation created: Rule Remove old diagnose d=Diagnose: Diagnose disease X: Type unknown(2)
 - ⇒ Activation created: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-12
- ◆ Activation executed: Rule Remove old diagnose d=Diagnose: Diagnose disease X: Type unknown(2)
 - Object removed (2): Diagnose: Diagnose disease X: Type unknown
 - ↳ Activation cancelled: Rule Start Clinical Pathway X if diagnosed d=Diagnose: Diagnose disease X: Type unknown(2)
 - ↳ Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-16-17
 - ↳ Activation cancelled: Rule RuleFlow-org.drools.examples.cdss.ClinicalPathwayX-12
 - ◆ Activation executed: Rule Examination3
 - 🔗 RuleFlowGroup deactivated: AdditionalExaminations[size=0]
 - 🔗 RuleFlow completed: TreatmentY[org.drools.examples.cdss.TreatmentY]
 - 🔗 RuleFlow started: TreatmentY[org.drools.examples.cdss.TreatmentY]
 - 🔗 RuleFlow completed: ClinicalPathwayX[org.drools.examples.cdss.ClinicalPathwayX]
- Object inserted (2): Diagnose: Diagnose disease X: Type 2

Comprehensive Blog

Open Mailing lists

IRC





- **Dave Bowman**: All right, HAL; I'll go in through the emergency airlock.
- **HAL**: Without your space helmet, Dave, you're going to find that rather difficult.
- **Dave Bowman**: HAL, I won't argue with you anymore! Open the doors!
- **HAL**: Dave, this conversation can serve no purpose anymore. Goodbye.

Joshua: Greetings, Professor Falken.

Stephen Falken: Hello, Joshua.

Joshua: A strange game. The only winning move is not to play. How about a nice game of chess?



Edson Tirelli

JBoss Sr. Software Engineer / Drools Core Dev
etirelli@redhat.com