

Isaac Simões Araújo Pereira

Drools Situation: uma Abordagem Baseada em Regras para Detecção de Situações

Monografia apresentada para obtenção do grau de Bacharel em Ciência da Computação pela Universidade Federal do Espírito Santo.

Orientadora:

Prof^a. Dr^a. Patrícia Dockhorn Costa

Agenda

- Motivação e Objetivos
- Fundamentação Conceitual
- *JBoss Drools*
- Proposta de Modelo de Motor de Situações
- *Drools Situation*
- Padrões de Situação
- Conclusões e Trabalhos Futuros

Motivação

- Sensibilidade ao Contexto e Computação Pervasiva
- Pontos de Reatividade vs. Complexidade do Domínio
- Suporte Conceitual:
 - *Situation Specification and Realization in Rule-Based Context-Aware Applications* (COSTA, ALMEIDA, et al. 2006),
 - *Architectural Support for Context-Aware Applications - From Context Models to Services Platforms* (COSTA, 2007),
 - *Situations in Conceptual Modeling of Context* (COSTA, ALMEIDA, et al. 2007)

Motivação

- Carência de Ferramentas para:
 - Descrever,
 - Identificar,
 - E Gerenciar estados que exigem reação do sistema;
- *JBoss Drools*
 - Regras com suporte a eventos

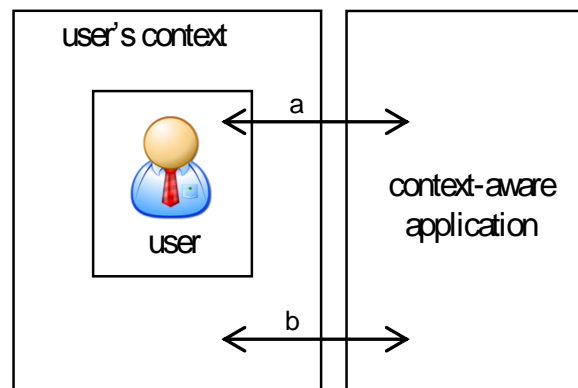
Objetivos

- Desenvolver uma ferramenta de gerência de situações
- Avaliar sua expressividade em relação a modelos de situação existentes.

Fundamentação Conceitual

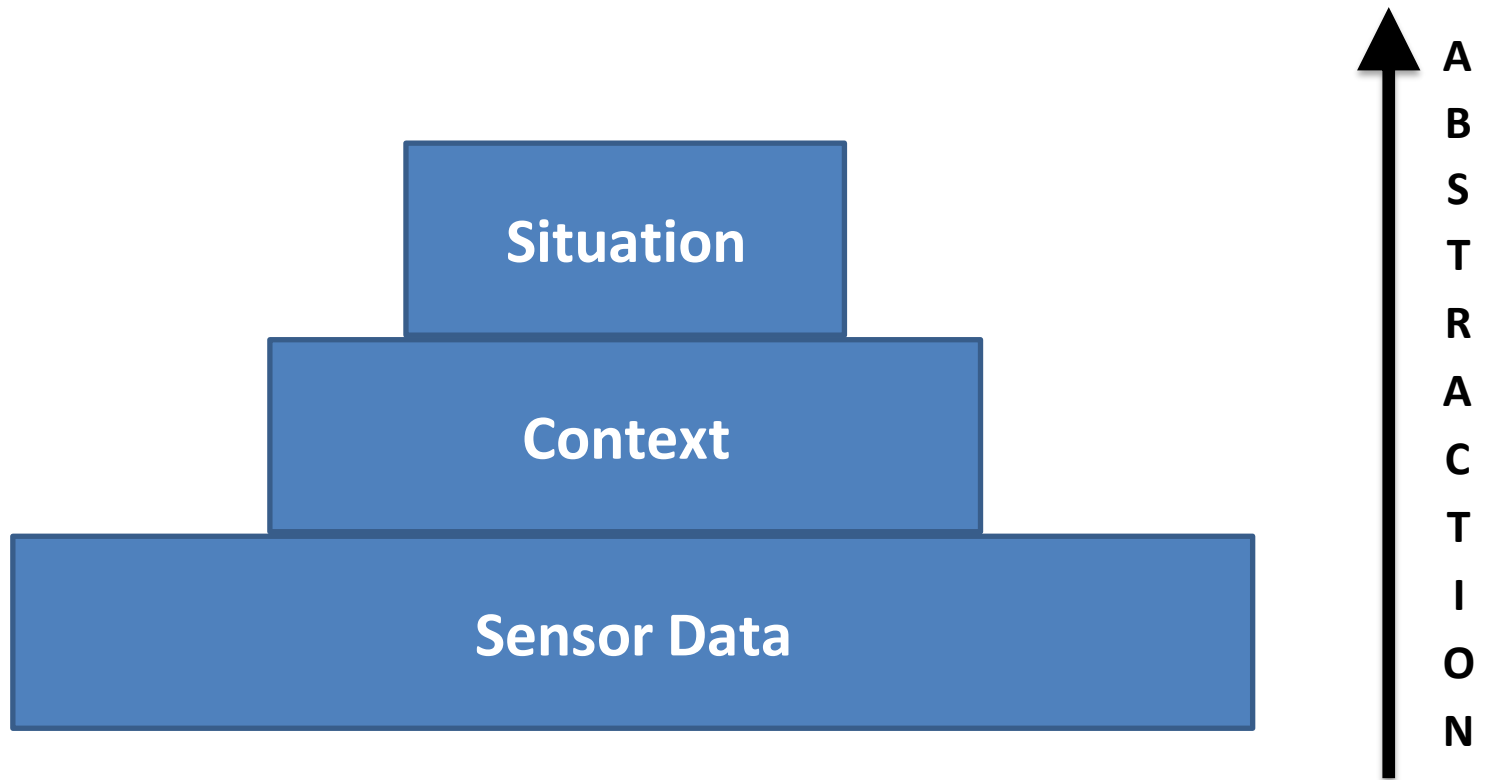
Sensibilidade ao Contexto

*“Context is any information that can be used to **characterize** the situation of an **entity**. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”*
(DEY, 1999)



Fundamentação Conceitual

Composição de Contexto



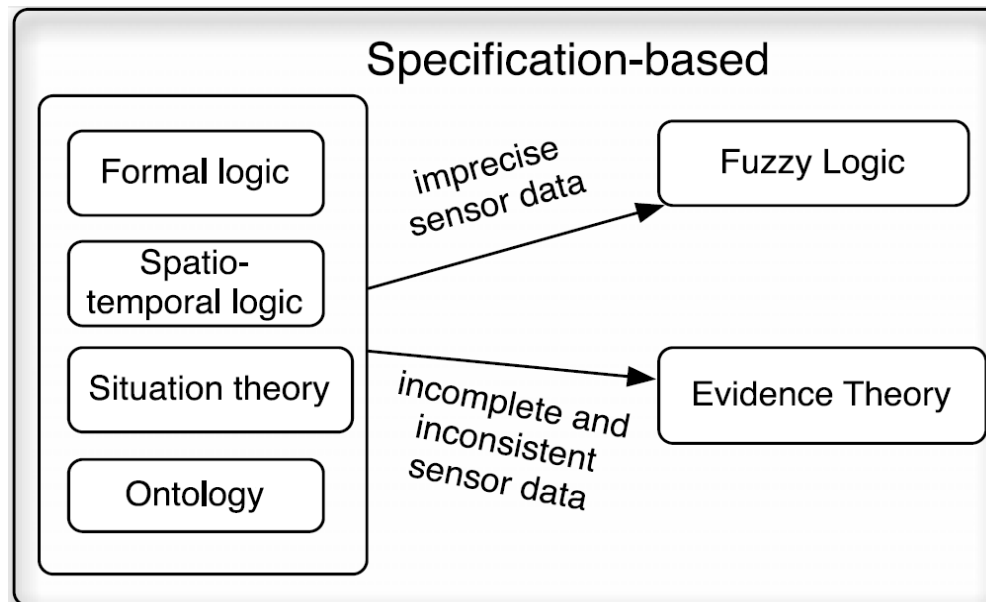
Fundamentação Conceitual

Situações

*“particular **state-of-affairs** that is of interest to applications. A situation is a composite concept whose constituents may be (a combination of) entities and their context conditions.”*
(COSTA, 2007)

Fundamentação Conceitual

Técnicas de Identificação de Situações



Fundamentação Conceitual

Paradigma Orientado a Regras

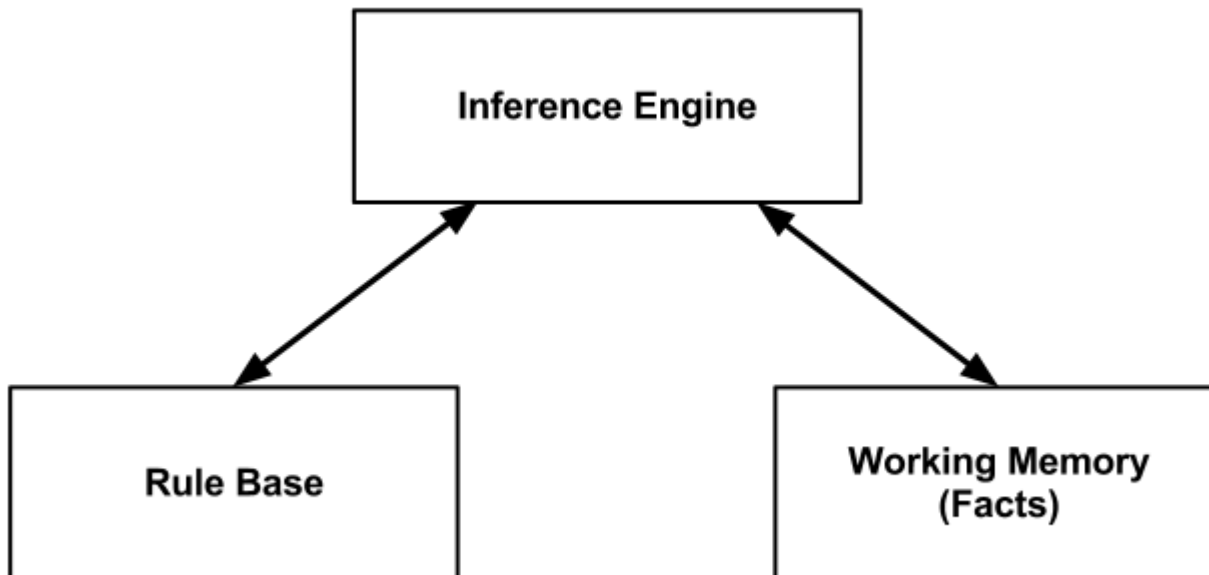
Regra: diretriz que aponta um curso usual, costumeiro, ou generalizado de ação ou comportamento sob certas condições.

se < premissa > ,

então < consequência > .

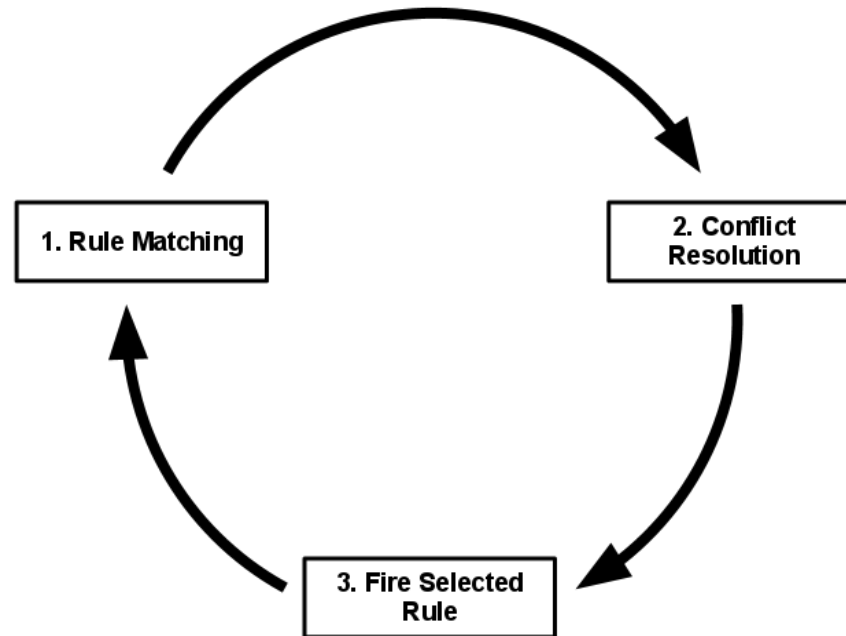
Fundamentação Conceitual

Sistemas Baseados em Regras



Fundamentação Conceitual

Match-Resolve-Act



Trabalhos Relacionados

- *A Language and an Execution Model for the Detection of Active Situations (ADI, 2002)*
- *Situation Inference for Mobile Users: a Rule Based Approach (GOIX et al., 2007)*



JBoss Drools



Guvnor 

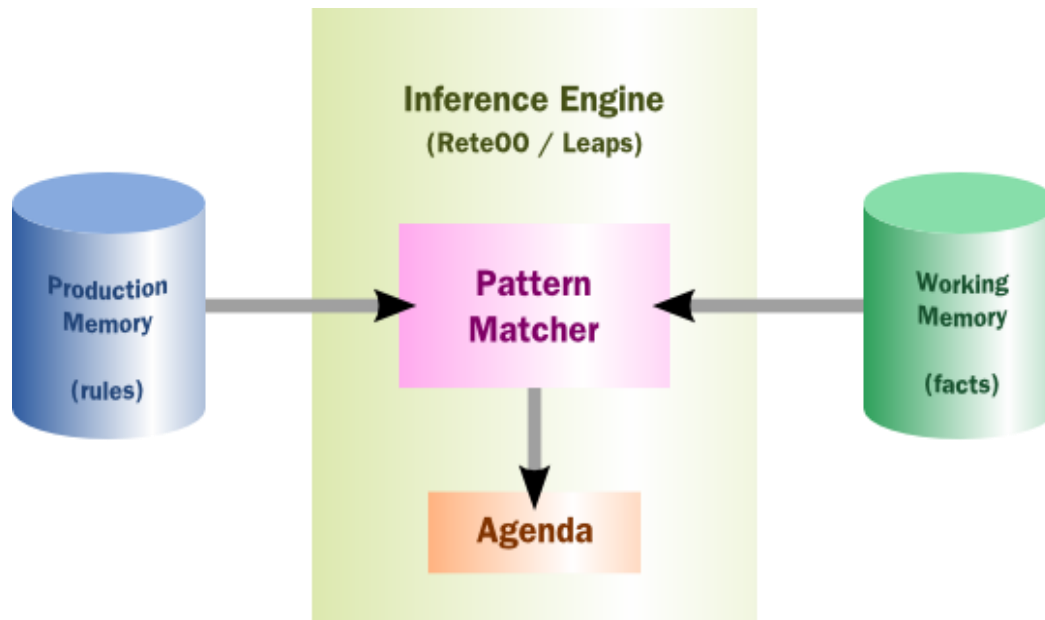
Expert 

Fusion 

Flow 

JBoss Drools

Expert



JBoss Drools

Drools Rule Language (DRL)

```
1 rule "Rule Label"  
2     when  
3         Pattern(restriction_1, ..., restriction_N)  
4     then  
5         //Action in Java Code  
6 end
```

Exemplo:

```
1 rule "Febre"  
2     when  
3         Paciente(temperatura > 37)  
4     then  
5         System.out.println("Paciente apresenta febre");  
6 end
```


JBoss Drools

Fusion

- “**Event**, is a record of a significant change of state in the application domain.”;
- *Características:*
 - *Point-in-Time* ou *Interval-based*;
 - **Imutáveis**

JBoss Drools

Fusion

Declaração de Evento

```
1 declare LigacaoTelefonica
2 //meta-atributos
3     @role(event)
4     @timestamp(inicio)
5     @duration(duracao)
6     @expires(24h)
7
8 end
```

JBoss Drools

Relações Temporais entre Eventos

	Point - Point	Point - Interval	Interval - Interval
A Before B B After A	A ● B ●	A ●—● B ●	A ●—● B ●—●
A Meets B B Met by A		A ●—● ⋮ ● B	A ●—● ⋮ B ●—●
A Overlaps B B Overlapped by A			A ●—● B ●—●
A Finishes B B Finished by A		B ●—● ⋮ A ●	B ●—● ⋮ A ●—●
A Includes B B During A		A ●—● B ●	A ●—● B ●—●
A Starts B B Started by A		A ● ⋮ B ●—●	A ●—● ⋮ B ●—●
A Coincides B	A ● ⋮ B ●		A ●—● ⋮ B ●—●



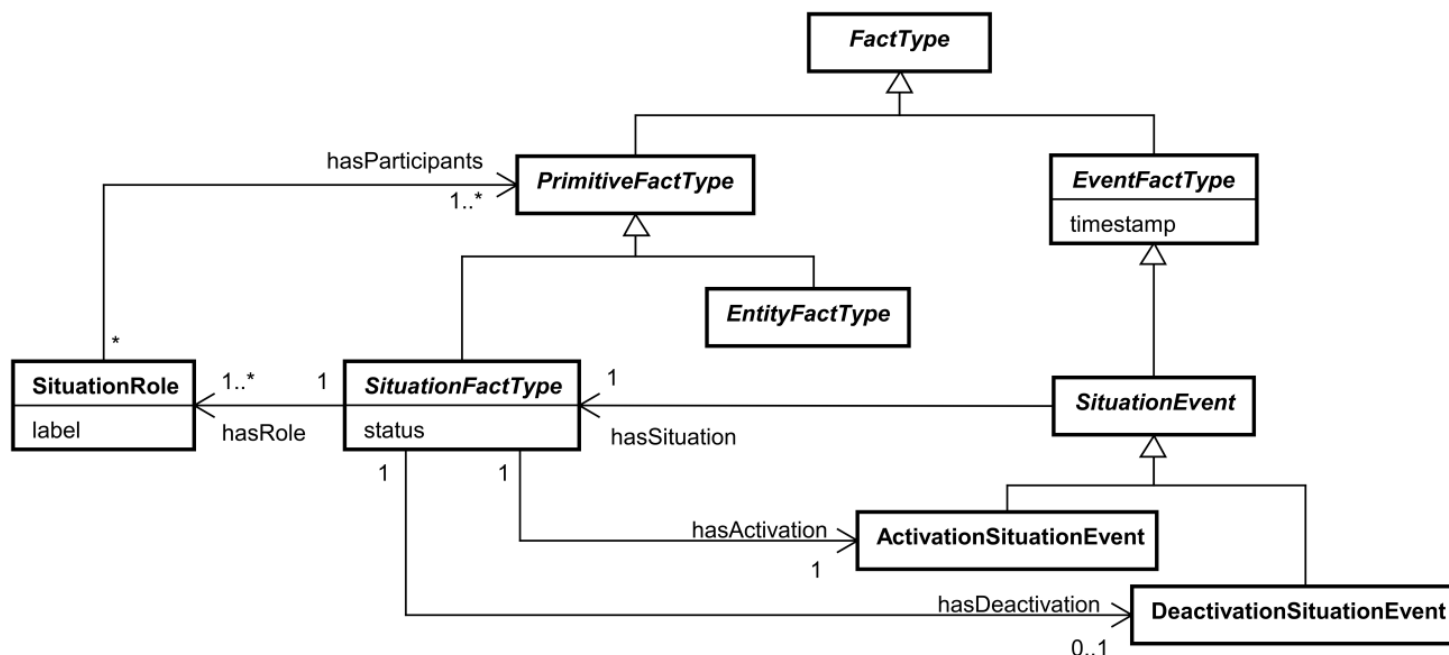
Modelo de Motor de Situações

Conceito

Entende-se como ***tipo de situação*** (*Situation Type*) um padrão que caracteriza um estado de interesse manifestável dentro do escopo de um determinado domínio de discurso. Caracterização esta, de natureza declarativa, que discursa sobre propriedades inerentes a, e relações (contextos) entre, entidades deste universo, restringindo-as.

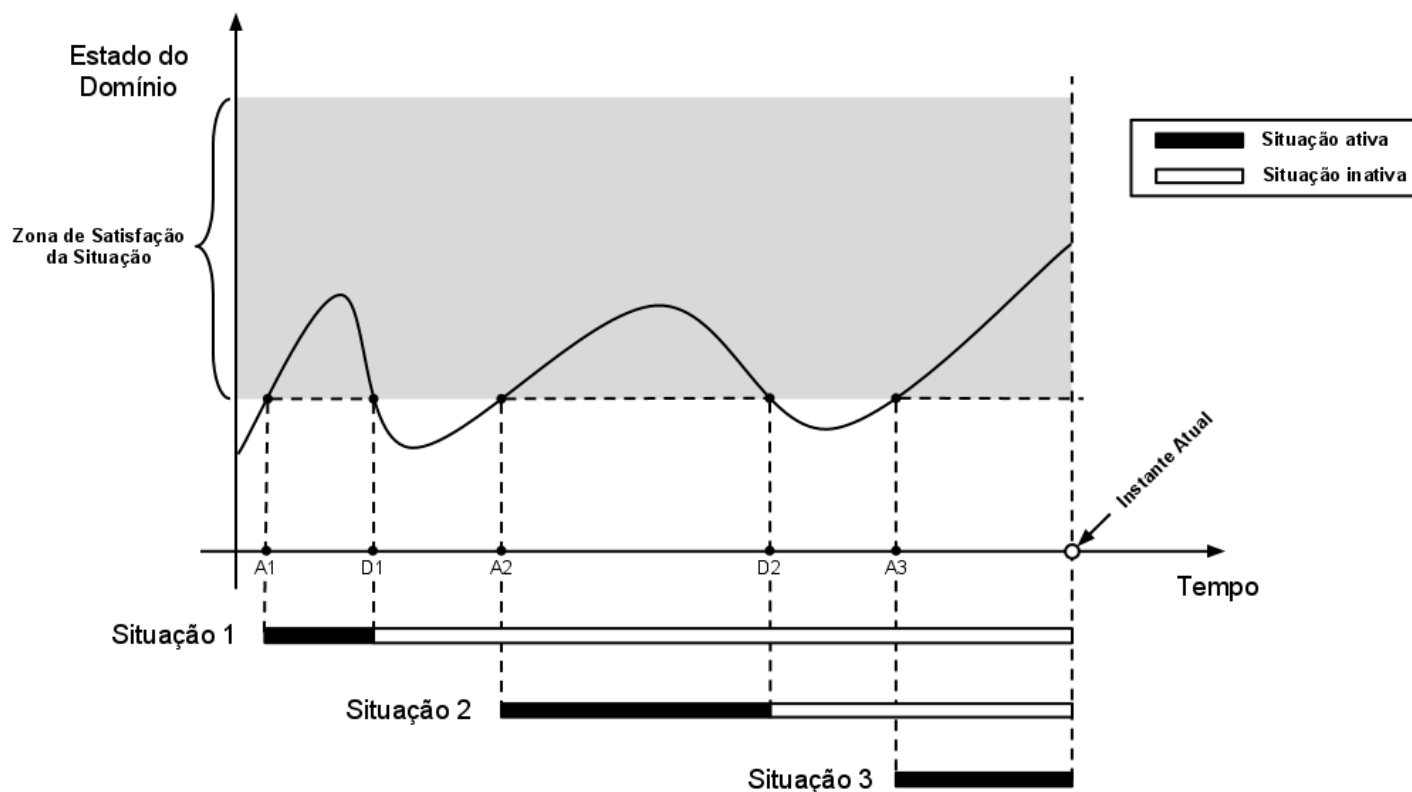
Modelo de Motor de Situações

Aspecto Estrutural



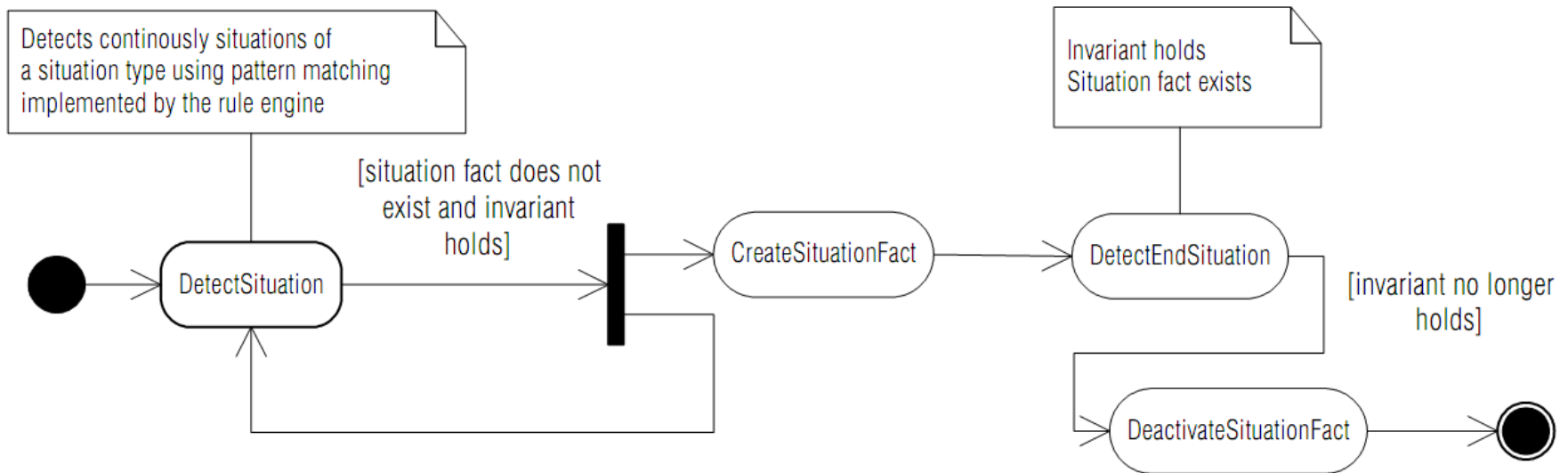
Modelo de Motor de Situações

Instância de Situação



Modelo de Motor de Situações

Identificação e Ciclo de Vida































Modelo de Motor de Situações

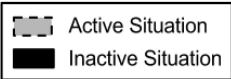
Identificação e Ciclo de Vida

- Desacoplamento entre Identificação e Gerência do Ciclo de Vida:
 - (Identificação) Regras de Situação
 - (Ciclo de Vida) Regras de Ativação e Desativação

Modelo de Motor de Situações

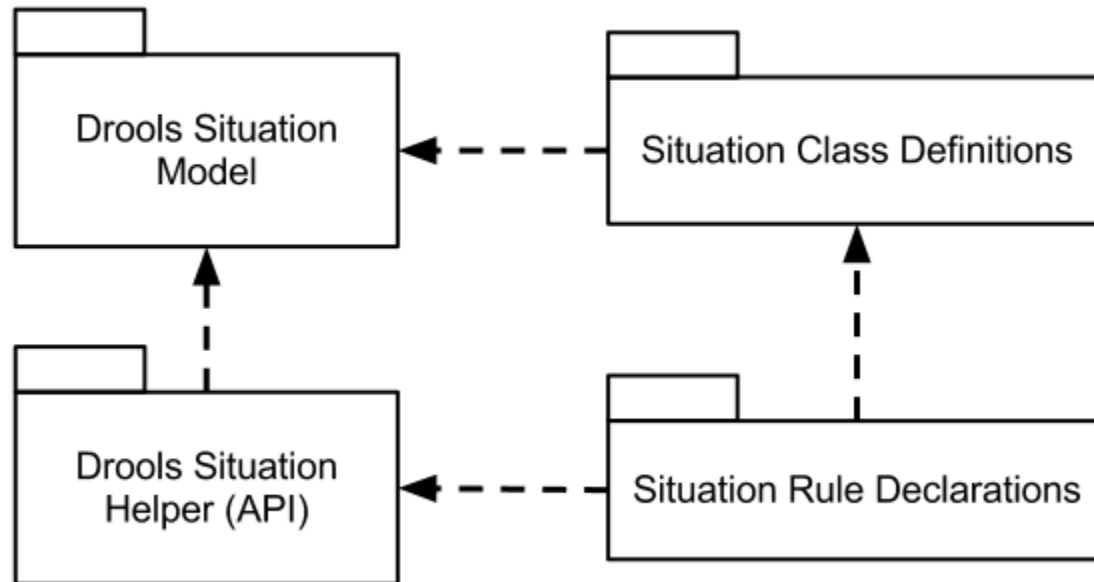
Relações Temporais entre Situações

	Active - Active	Active - Inactive	Inactive - Inactive
A Before B B After A		A  B 	A  B 
A Meets B B Met by A		A  B 	A  B 
A Overlaps B B Overlapped by A	A  B 	A  B 	A  B 
A Finishes B B Finished by A			A  B 
A Includes B B During A		A  B 	A  B 
A Starts B B Started by A	A  B 	A  B 	A  B 
A Coincides B			A  B 



Drools Situation

Módulo de Gerência de Situações

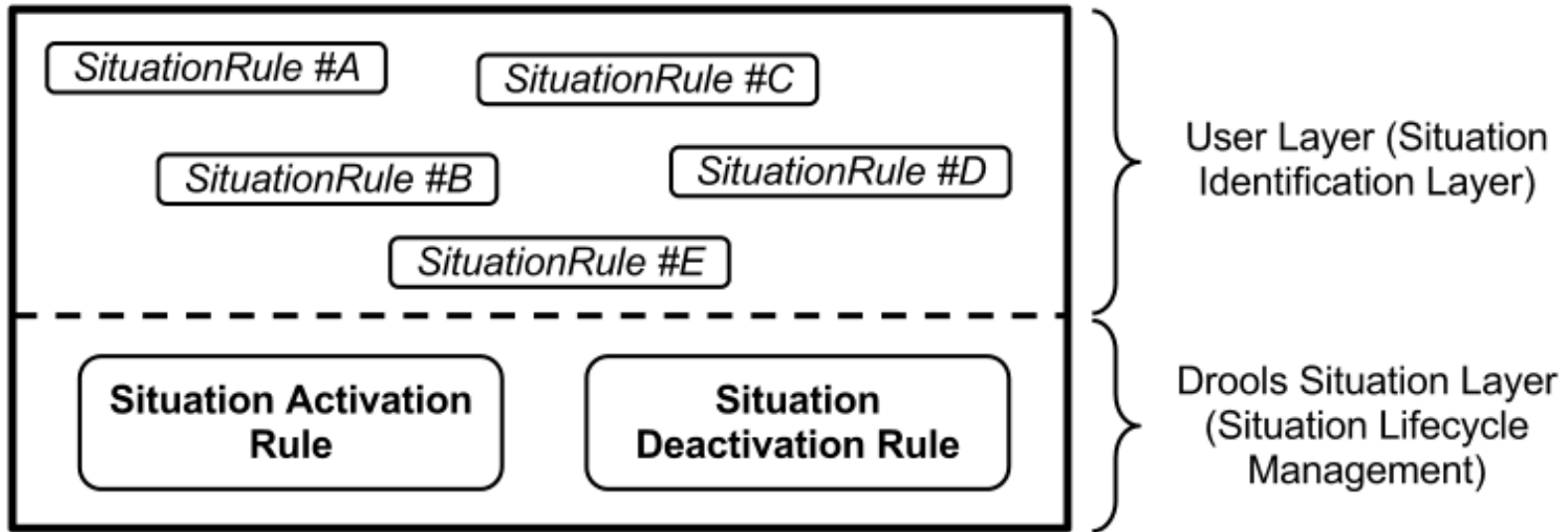


Drools Situation

SituationType

```
1 public abstract class SituationType {  
2     private Boolean          active;  
3     private ActivateSituationEvent activation;  
4     private DeactivateSituationEvent deactivation;  
5     //GETTERS AND SETTERS  
6 }
```


Drools Situation



Drools Situation General Rule Base

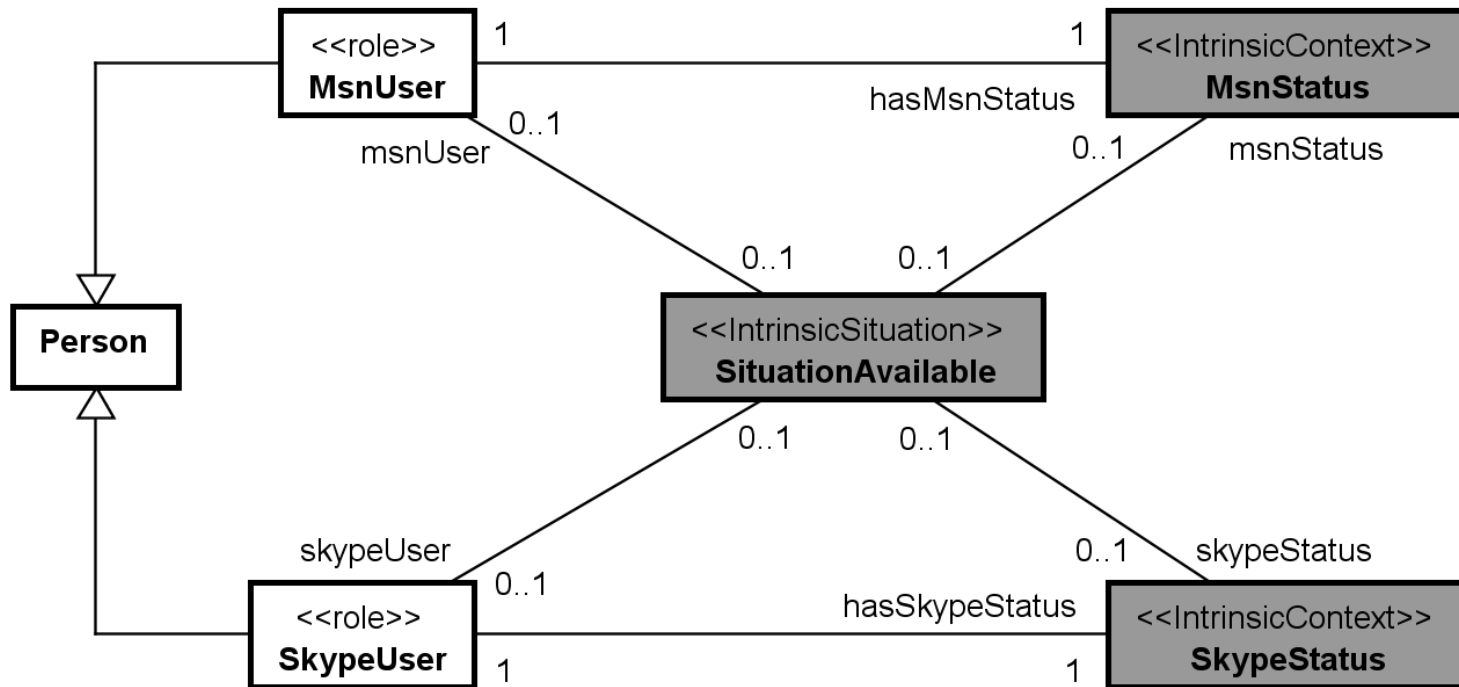
Drools Situation

Regra *Situation Deactivation*

```
1 rule "SituationDeactivation"  
2   when  
3     $sit: SituationType(active==true)  
4     not (exists CurrentSituation(situation == $sit))  
5   then  
6     SituationHelper.deactivateSituation(drools, (Object) $sit);  
7 end
```

Situation Patterns

Situação Intrínseca



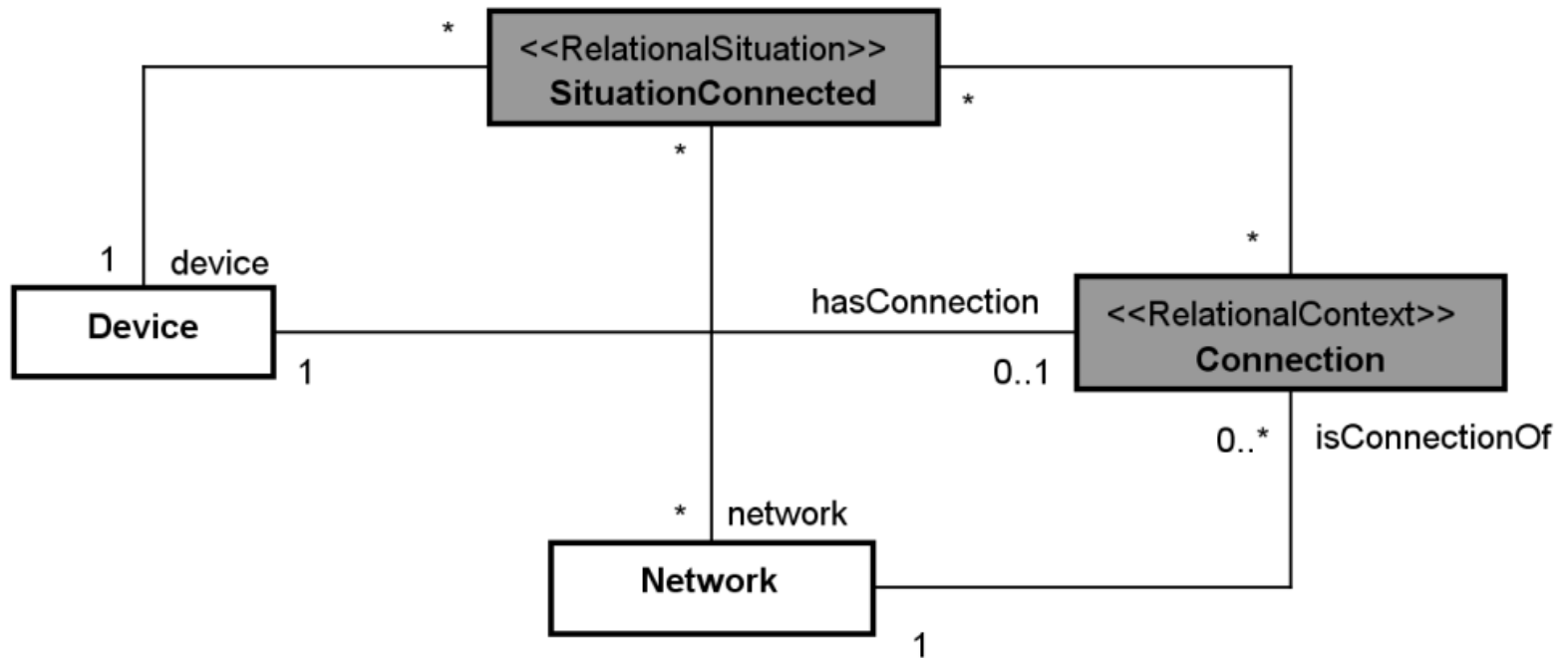
Situation Patterns

Situação Intrínseca

O C L	<pre>{Context SituationAvailable inv: (skypeUser = msnUser) AND ((not skypeUser.oclIsUndefined()) AND (skypeUser.skypeStatus = skypeStatus) AND ((skypeStatus.value = "Online") OR (skypeStatus.value = "SkypeMe"))) OR ((not msnUser.oclIsUndefined()) AND (msnUser.msnStatus = msnStatus) AND ((msnStatus.value = "Online") OR (msnStatus.value = "BeRightBack")))} }</pre>
D R L	<pre>Rule "SituationAvailable" when \$MSNUser: MSNUser(\$MSNStatus: MSNStatus) \$SkypeUser: SkypeUser(this==\$MSNUser, \$SkypeStatus: SkypeStatus) (MSNStatus(this==\$MSNStatus, value(=="OnLine" =="BeRightBack")) or SkypeStatus(this==\$SkypeStatus, value(=="OnLine" =="SkypeMe"))) then SituationHelper.situationDetected(drools, kcontext, SituationAvailable.class) end</pre>

Situation Patterns

Situação Relacional



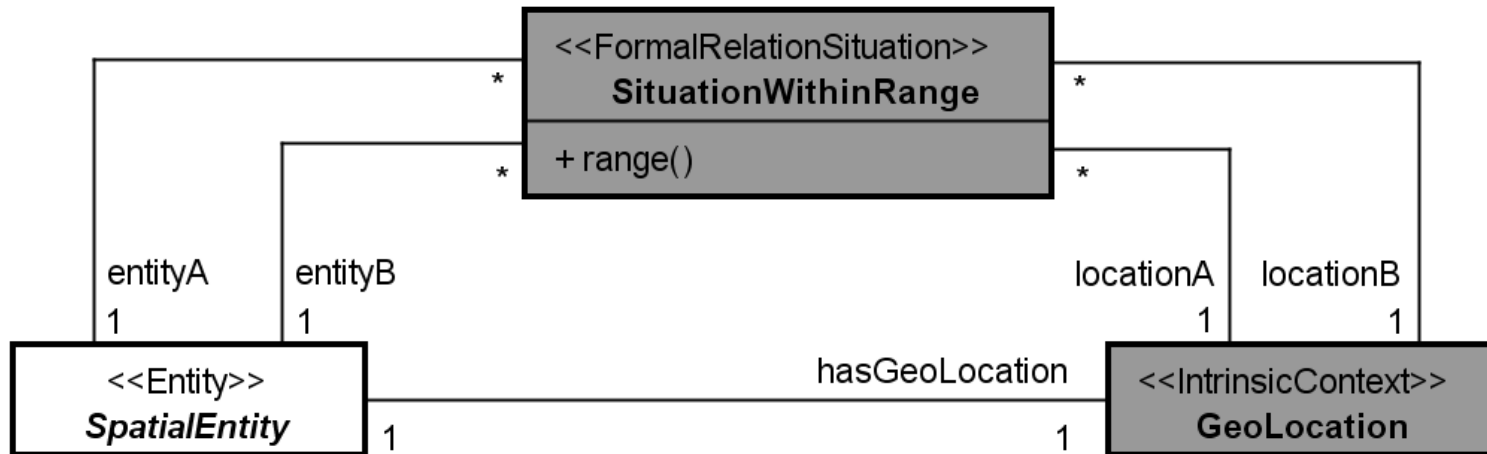
Situation Patterns

Situação Relacional

O C L	<pre>{Context SituationConnected inv: not device.hasConnection.oclIsUndefined()}</pre>
D R L	<pre>rule "SituationConnected" when \$network: Network() \$connection: Connection(network==\$network) \$device: Device(connection==\$connection) then <i>SituationHelper</i>.situationDetected(drools, kcontext, SituationConnected.class) end</pre>

Situation Patterns

Situação de Relação Formal



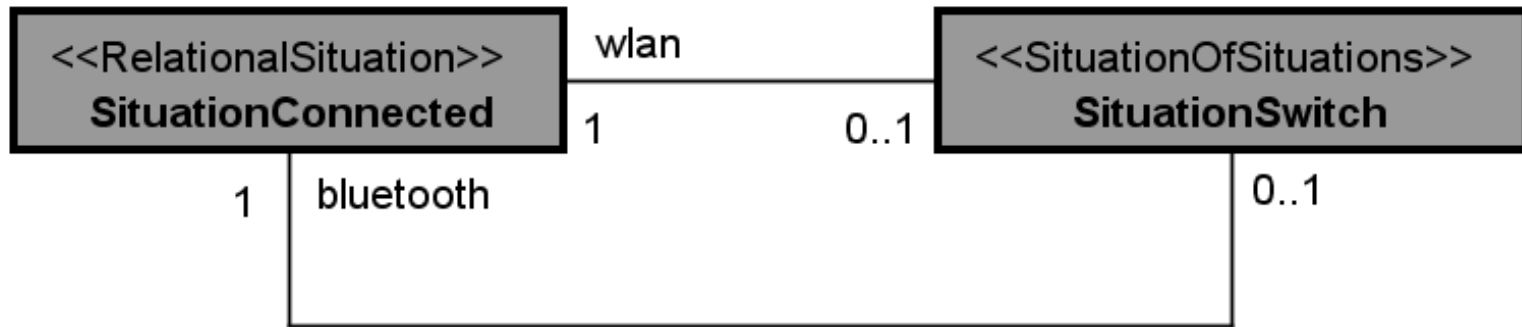
Situation Patterns

Situação de Relação Formal

O C L	<pre>{Context SituationWithinRange inv: entityA.hasGeoLocation = locationA AND entityB.hasGeoLocation = locationB AND locationA.value->distance(locationB.value) < range}</pre>
D R L	<pre>rule "SituationWithinRange" when \$entityA: SpatialEntity (\$locationA: location) \$entityB: SpatialEntity (this!=\$entityA, \$locationB: location) eval(\$locationA.distance(\$locationB) < range) then <i>SituationHelper</i>.situationDetected(drools, kcontext, SituationWithinRange.class) end</pre>

Situation Patterns

Situação de Situações



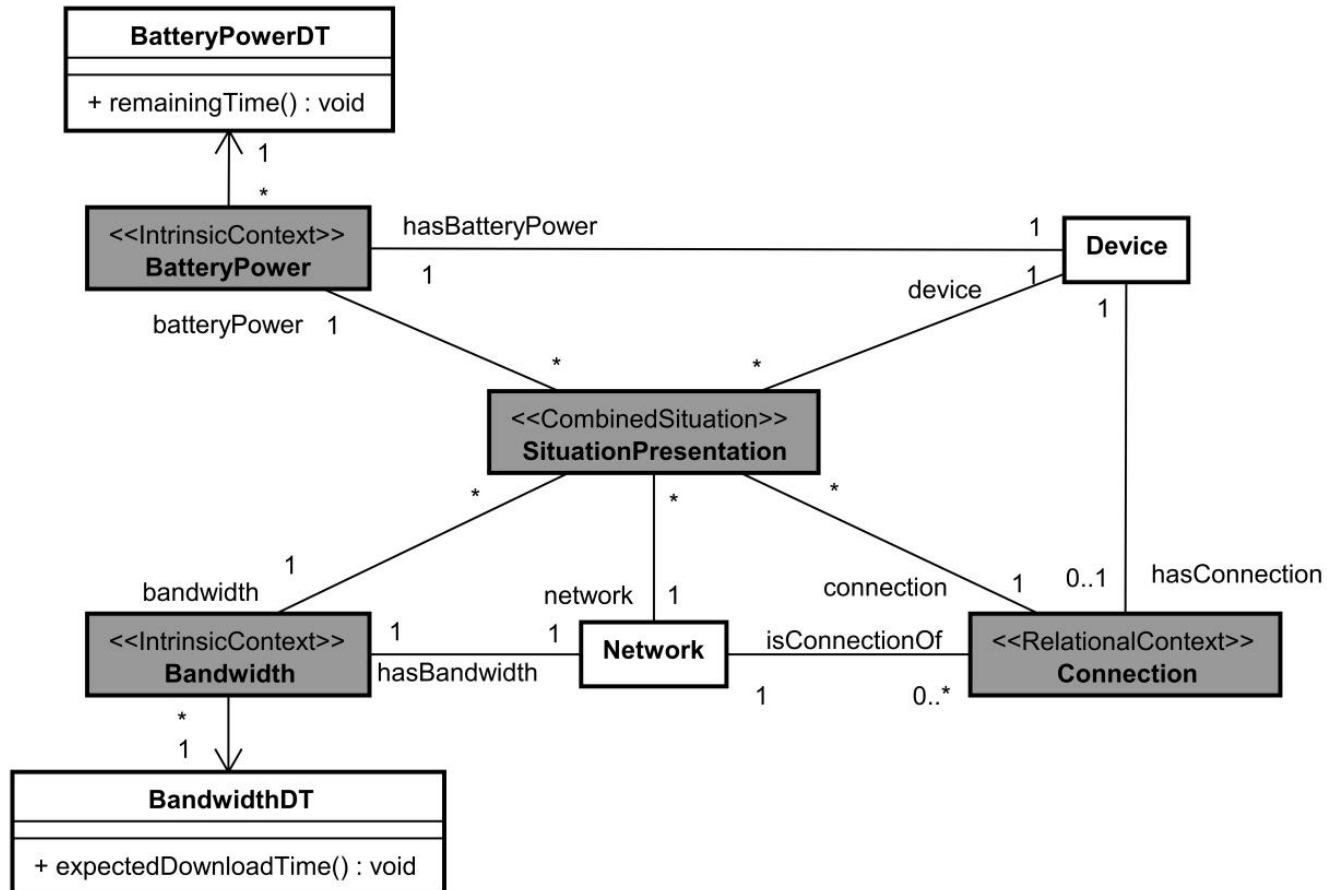
Situation Patterns

Situação de Situações

O C L	<pre>{ Context SituationSwitch inv: (wlan.device = bluetooth.device) AND (wlan.device.hasConnection.network.oclIsTypeOf(WLAN)) AND (bluetooth.device.hasConnection.network.oclIsTypeOf(Bluetooth)) AND (bluetooth.initialtime - wlan.finaltime < 1)}</pre>
D R L	<pre>rule "SituationSwitch" when \$wlan: SituationConnected(\$device: device, network.type==WLAN) \$bluetooth: SituationConnected(device==\$device, network.type==BLUETOOTH, this metby[1s] \$wlan) then SituationHelper.situationDetected(drools, kcontext, SituationSwitch.class) end</pre>

Situation Patterns

Combinação de Situações



Situation Patterns

Combinação de Situações

O C L	<pre>{ Context SituationPresentation inv: (not device.hasConnection.isOclUndefined()) AND (device.hasConnection = connection) AND (connection.network = network) AND (network.hasBandwidth = bandwidth) AND (device.hasBatteryPower = batterypower) AND ((bandwidth.value->expectedDownloadTime(<i>presentationsize</i>) + <i>presentationduration</i>) < batterypower.value->remainingTime())}</pre>
D R L	<pre>rule "SituationPresentation" when \$network: Network(\$bandwidth: bandwidth) \$connection: Connection(network==\$network) \$device: Device(connection==\$connection, \$batterypower: batterypower) eval((\$bandwidth.value.expectedDownloadTime(<i>presentationsize</i>) + <i>presentationduration</i>) < \$batterypower.value.remainingTime) then <i>SituationHelper</i>.situationDetected(drools, kcontext, SituationPresentation.class) end</pre>

Conclusões e Trabalhos Futuros

- Simplicidade na especificação de *regras de situação*;
- Possível impacto da arquitetura na performance da engine *Drools*;
- Expressividade compatível com os padrões situacionais de (COSTA, 2007)
 - Exceção: *Situação de Situações*

Conclusões e Trabalhos Futuros

Futuro

- Cobrir completamente a especificação de *situações de situações*;
- Analisar cenários complexos a fim de determinar a razoabilidade da performance;
- Efetuar melhorias no editor de regras para suportar sintaticamente as alterações causadas pelo *Drools Situation*.

