

Exame de Sistemas Baseados em Conhecimento

Época Normal 19/02/2021

Mestrado em Engenharia Informática do ISEP

Exame com consulta

Duração: 1 hora

Considere um sistema de transporte de energia eléctrica constituído por diferentes tipos de componentes. Estes componentes dispõem de dispositivos capazes de gerar e enviar mensagens descritivas de eventos que ocorrem nesses componentes. As mensagens contêm uma descrição do evento, assim como informação temporal relativa ao instante de ocorrência do evento. Todas as mensagens geradas no sistema são encaminhadas para um centro de controlo e usadas para alimentar um Sistema Baseado em Conhecimento (SBC) que tem por objetivo interpretar o estado de funcionamento do sistema, assim como diagnosticar eventuais avarias.

1. [20%] A tarefa de diagnóstico de avarias, que é função do Sistema Baseado em Conhecimento, é realizada com base na identificação de sequências estereotipadas de mensagens geradas no sistema de transporte de energia eléctrica e na verificação de restrições temporais entre mensagens presentes nas sequências.

Considerando os diferentes tipos de conhecimento estudados, caracterize o conhecimento que será utilizado pelo sistema a desenvolver e identifique, justificando, o formalismo de representação de conhecimento mais adequado para a construção da base de conhecimento.

2. [20%] Considere que pretendemos implementar o SBC descrito em Drools e que as mensagens descritivas de eventos são representadas na memória de trabalho através de factos com a designação *ScadaMessage*. Estes factos possuem os atributos *type*, *plant*, *panel*, *plant2* e *nl*. Todos estes atributos são do tipo *string*. Apresente o código necessário para declarar este tipo de facto em Drools.

3. [20%] Considere o cenário em que a memória de trabalho do SBC contém todos os factos necessários para que ambas as regras que se apresentam a seguir possam disparar.

```
rule "d1 - Disparo trifasico com disjuntor em movimento"
when
    ScadaMessage( type == ScadaMessageType.DISPAR001, $plant1 : plant,
                  $panel1 : panel, $plant2 : plant2, $nl : nl)
    ScadaMessage( type == ScadaMessageType.DISJUNTOR00, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
    ScadaMessage( type == ScadaMessageType.DISJUNTOR10, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
then
    Conclusion c = new Conclusion($plant1, $panel1, $plant2, $nl,
                                ConclusionType.DISP_TRIF_DISJ_MOV);
end

rule "d2 - Disparo tipo nao identificado"
when
    ScadaMessage( type == ScadaMessageType.DISPAR001, $plant1 : plant,
                  $panel1 : panel, $plant2 : plant2, $nl : nl)
then
    Conclusion c = new Conclusion($plant1, $panel1, $plant2, $nl,
                                ConclusionType.DISP_N_IDENT);
end
```

Considere que o motor de inferência do Drools foi adaptado para usar um mecanismo de resolução de conflitos por especificidade. Neste cenário, indique como é que o motor de inferência deve atuar. Justifique a sua resposta.

4. [20%] Considere as seguintes regras:

```
rule "d1 - Disparo trifasico com disjuntor em movimento"
when
    ScadaMessage( type == ScadaMessageType.DISPAR001, $plant1 : plant,
                  $panel1 : panel, $plant2 : plant2, $nl : nl)
    ScadaMessage( type == ScadaMessageType.DISJUNTOR00, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
    ScadaMessage( type == ScadaMessageType.DISJUNTOR10, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
then
    Conclusion c = new Conclusion($plant1, $panel1, $plant2, $nl,
                                ConclusionType.DISP_TRIF_DISJ_MOV);
end

rule "d3 - Religacao trifasica"
when
    Conclusion($plant1 : plant, $panel1 : panel, $plant2 : plant2,
              $nl : nl, conclusion == ConclusionType.DISP_TRIF_DISJ_MOV)
    ScadaMessage( type == ScadaMessageType.DISJUNTOR01, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
then
    Conclusion c = new Conclusion($plant1, $panel1, $plant2, $nl,
                                ConclusionType.RELIG_TRIF);
end
```

Considere ainda que a regra *d1* dispara e que na memória de trabalho existe um facto que permite validar a segunda condição da regra *d3* (*ScadaMessage()*), com os mesmos valores para os atributos *plant*, *panel*, *plant2* e *nl* que permitiram o disparo da regra *d1*.

Nas circunstâncias descritas, indique se a regra *d3* dispara. Justifique a sua resposta.

5. [20%] Considere as seguintes regras:

```
rule "d1 - Disparo trifasico com disjuntor em movimento"
when
    ScadaMessage( type == ScadaMessageType.DISPAR001, $plant1 : plant,
                  $panel1 : panel, $plant2 : plant2, $nl : nl)
    ScadaMessage( type == ScadaMessageType.DISJUNTOR00, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
    ScadaMessage( type == ScadaMessageType.DISJUNTOR10, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
then
    Conclusion c = new Conclusion($plant1, $panel1, $plant2, $nl,
                                ConclusionType.DISP_TRIF_DISJ_MOV);
    insertLogical(c);
end

rule "d2 - Religacao trifasica"
when
    ScadaMessage( type == ScadaMessageType.DISJUNTOR01, plant == $plant1,
                  panel == $panel1, plant2 == $plant2, nl == $nl)
then
    Conclusion c = new Conclusion($plant1, $panel1, $plant2, $nl,
                                ConclusionType.RELIG_TRIF);
    insertLogical(c);
end
```

Considere ainda que os factos do tipo *ScadaMessage()* presentes na memória de trabalho em dois instantes *T1* e *T2*, sendo *T1* anterior a *T2*, são os seguintes:

Instante *T1*:

```
ScadaMessage( ScadaMessageType.DISPAR001, "PlantA", "PanelA", "PlantB", "1")
ScadaMessage( ScadaMessageType.DISJUNTOR00, "PlantA", "PanelA", "PlantB", "1")
ScadaMessage( ScadaMessageType.DISJUNTOR10, "PlantA", "PanelA", "PlantB", "1")
```

Instante *T2*:

```
ScadaMessage( ScadaMessageType.DISPAR001, "PlantA", "PanelA", "PlantB", "1")
ScadaMessage( ScadaMessageType.DISJUNTOR01, "PlantA", "PanelA", "PlantB", "1")
```

Indique quais são os factos do tipo *Conclusion()* presentes na memória de trabalho logo após o instante *T1* e logo após o instante *T2*.

Nota: Entre os instantes *T1* e *T2* foram removidos os factos *ScadaMessage(DISJUNTOR00,...)* e *ScadaMessage(DISJUNTOR10,...)* e foi inserido o facto *ScadaMessage(DISJUNTOR01,...)*.

1->- Usar conhecimento procedimental, pois permite definir sequências e serve para lidar com situações distintas.
Ver resposta do exame normal 21-22

2->- Criar a classe em java, associar ao drools através de um declare ScadaMessage

- Nas regras utilizar \$Message para utilizar as strings (\$message : ScadaMessage(type == "Alerta", plant == "Planta A"))
- i dunno

3->- Neste caso, o drools vai disparar a primeira regra, pois é a que possui mais detalhe (maior nível de especificidade/mais condições). Regras 1 e 2 possuem uma condição em semelhança, no entanto a 1 é mais completa.

4->

- Sim, a regra dispara, pois todas as suas condições estão satisfeitas. Como a regra d1 já disparou e criou a conclusão que d3 verifica, não haverá problemas.

5->

- Logo após o T1 temos esta conclusão:

```
Conclusion c = new Conclusion ( $plant1 , $panel1 , $plant2 , $nl ,  
ConclusionType . DISP_TRIF_DISJ_MOV );
```

- logo após T2 temos:

```
Conclusion(plant="PlantA", panel="PanelA", plant2="PlantB", nl="1", conclusion=DISP_TRIF_DISJ_MOV) e  
Conclusion(plant="PlantA", panel="PanelA", plant2="PlantB", nl="1", conclusion=RELIG_TRIF)
```

