

Safe Composition of Agent-oriented Software Product Lines

Elder Cirilo,
Leopoldo Teixeira, et. al.

Problema

Hello World SPL ...

```
public class HelloWorldSPL {  
    public static void main(String args[]) {  
        Console.print(args);  
  
        Email.print(args);  
  
        PDF.print(args);  
    }  
}
```



Console



Email



PDF

Hello World SPL ...

```
public class HelloWorldSPL {  
    public static void main(String args[]) {  
        Console.print(args);  
        Email.print(args);  
        PDF.print(args);  
    }  
}
```



CLASS

Console



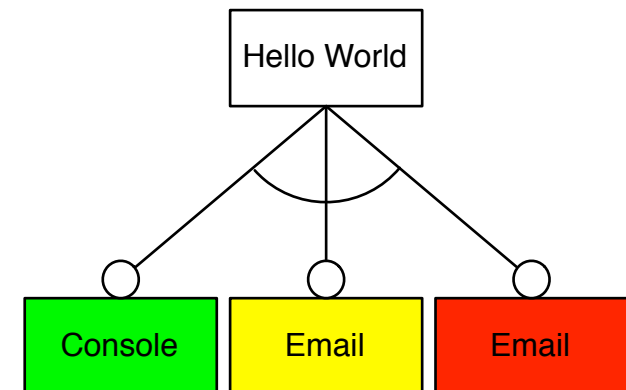
CLASS

Email



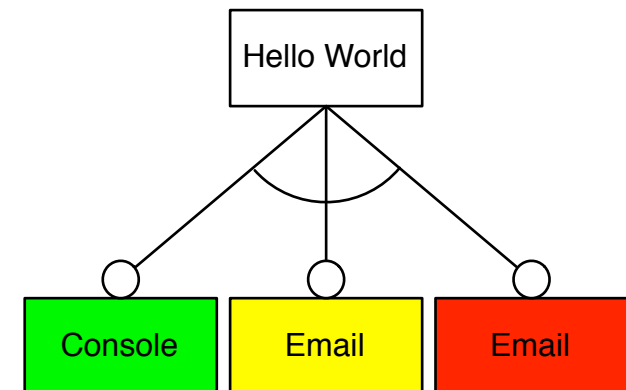
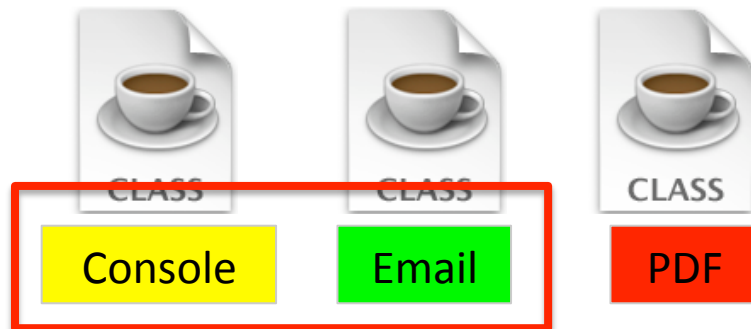
CLASS

PDF



Hello World SPL ...

```
public class HelloWorldSPL {  
    public static void main(String args[]) {  
        Console.print(args);  
        Email.print(args);  
        PDF.print(args);  
    }  
}
```

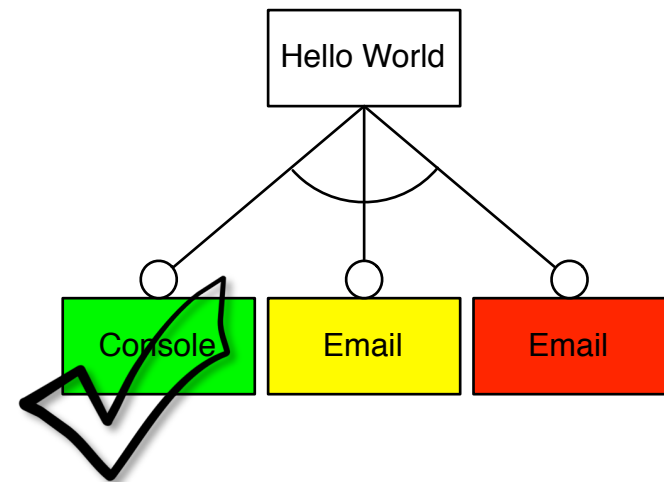


Hello World SPL ...

```
public class HelloWorldSPL {  
    public static void main(String args[]) {  
         Console.print(args);  
    }  
}
```



Email



Pode ser pior?

Spring Hello World SPL ...

```
public class HelloWorldBean {  
    public HelloWorldBean(Printer printer) { ... }  
    public static void print(String args[]) {  
        Printer.print(args);  
    }  
}
```

```
<beans>  
    <bean id="HWB" class="HelloWorldBean">  
        <constructor-args name="Printer">  
        </bean>  
  
    <bean id="Printer" class="ConsolePrinter"/>  
    <bean id="Printer" class="EmailPrinter"/>  
    <bean id="Printer" class="PDFPrinter"/>  
</beans>
```



ConsolePrinter



EmailPrinter

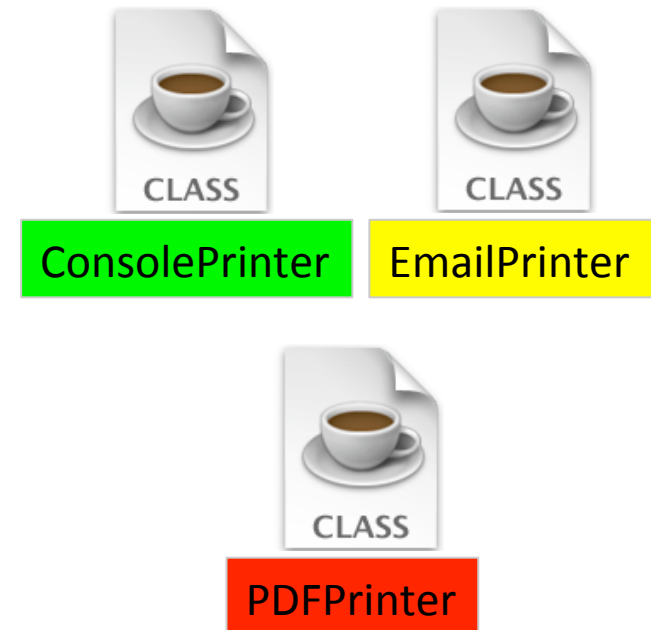
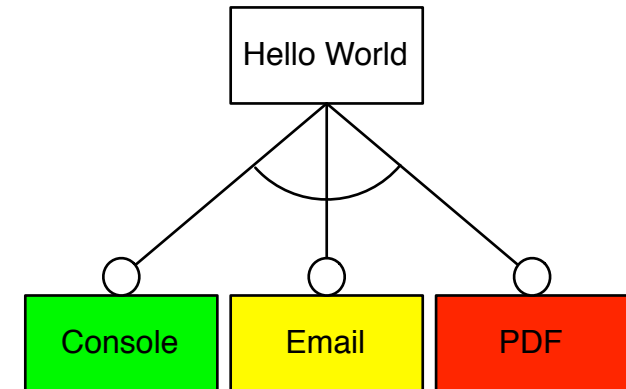


PDFPrinter

Spring Hello World SPL ...

```
public class HelloWorldBean {  
    public HelloWorldBean(Printer printer) { ... }  
    public static void print(String args[]) {  
        Printer.print(args);  
    }  
}
```

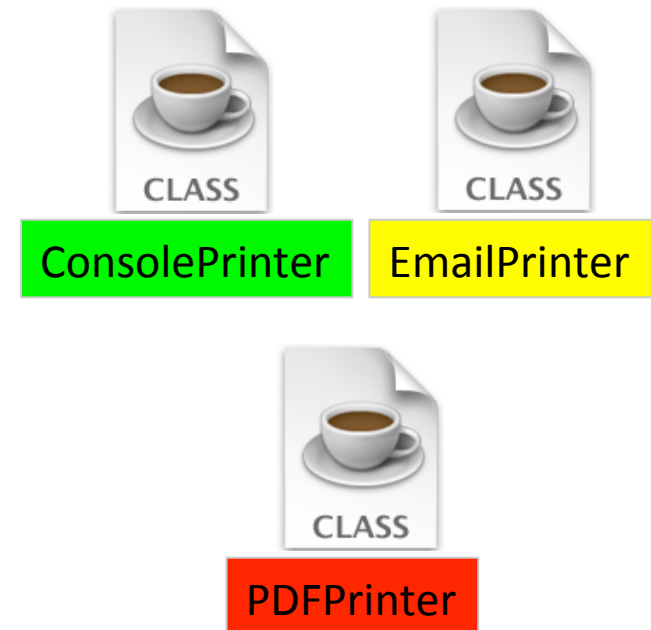
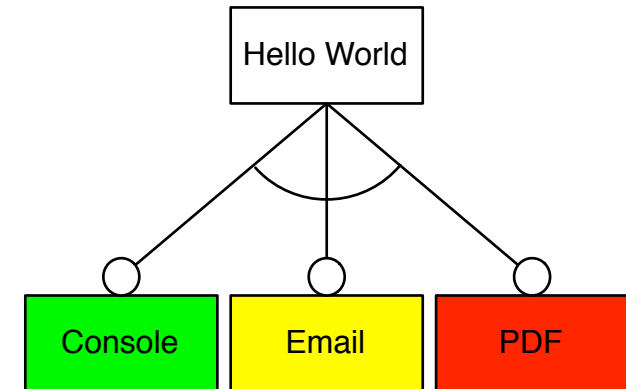
```
<beans>  
    <bean id="HWB" class="HelloWorldBean">  
        <constructor-args name="Printer">  
        </bean>  
  
    <bean id="Printer" class="ConsolePrinter"/>  
    <bean id="Printer" class="EmailPrinter"/>  
    <bean id="Printer" class="PDFPrinter"/>  
</beans>
```



Spring Hello World SPL ...

```
public class HelloWorldBean {  
    public HelloWorldBean(Printer printer) { ... }  
    public static void print(String args[]) {  
        Printer.print(args);  
    }  
}
```

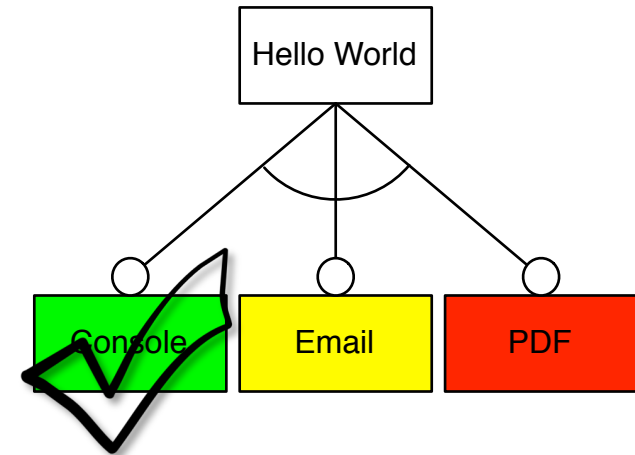
```
<beans>  
    <bean id="HWB" class="HelloWorldBean">  
        <constructor-args name="Printer">  
            </bean>  
        </bean>  
        <bean id="Printer" class="ConsolePrinter"/>  
        <bean id="Printer" class="EmailPrinter"/>  
        <bean id="Printer" class="PDFPrinter"/>  
    </beans>
```



Produto mal formado ...

```
public class HelloWorldBean {  
    public HelloWorldBean(Printer printer) { ... }  
    public static void print(String args[]) {  
        Printer.print(args);  
    }  
}
```

```
<beans>  
    <bean id="HWB" class="HelloWorldBean">  
        <constructor-args name="Printer">  
        </bean>  
  
    ✖ <bean id="Printer" class="EmailPrinter"/>  
</beans>
```



ConsolePrinter

Agent Hello World SPL ...

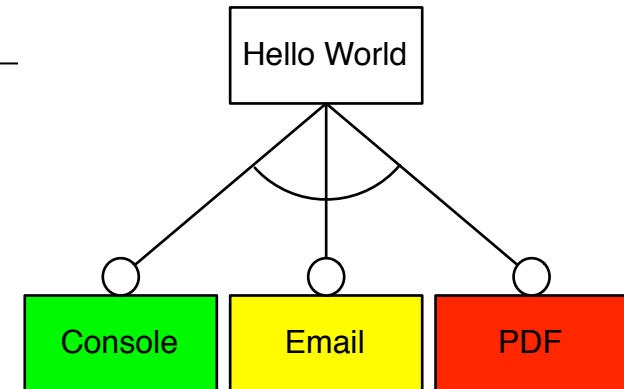
```
<agent name="HelloWorld">
  <beliefs>
    <belief name="ConsolePrinter" />
    <belief name="EmailPrinter" />
  </beliefs>

  <goals>
    <goal name="PrintConsole"/>
    <goal name="PrintEmail"/>
  </goals>

  <plans>
    <plan name="Print">
      <body class="HelloWorld"/>
      <trigger>
        <achievegoal name="PrintConsole"/>
        <achievegoal name="PrintEmail"/>
      </trigger>
    </plan>
  </plans>
</agent>
```

Agent Hello World SPL ...

```
<agent name="HelloWorld">
  <beliefs>
    <belief name="ConsolePrinter" />
    <belief name="EmailPrinter" />
  </beliefs>
  <goals>
    <goal name="PrintConsole"/>
    <goal name="PrintEmail"/>
  </goals>
  <plans>
    <plan name="Print">
      <body class="HelloWorld"/>
      <trigger>
        <achievegoal name="PrintConsole"/>
        <achievegoal name="PrintEmail"/>
      </trigger>
    </plan>
  </plans>
</agent>
```



Agent Hello World SPL ...

```
<agent name="HelloWorld">
```

```
<beliefs>
```

```
<belief name="ConsolePrinter" />
```

```
<belief name="EmailPrinter" />
```

```
</beliefs>
```

```
<goals>
```

```
<goal name="PrintConsole"/>
```

```
<goal name="PrintEmail"/>
```

```
</goals>
```

```
<plans>
```

```
<plan name="Print">
```

```
<body class="HelloWorld"/>
```

```
<trigger>
```

```
<achievegoal name="PrintConsole"/>
```

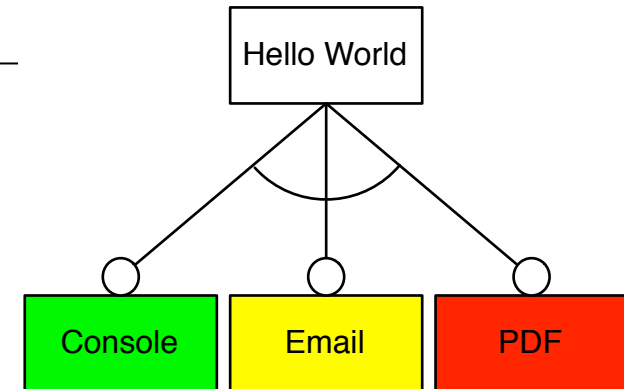
```
<achievegoal name="PrintEmail"/>
```

```
</trigger>
```

```
</plan>
```

```
</plans>
```

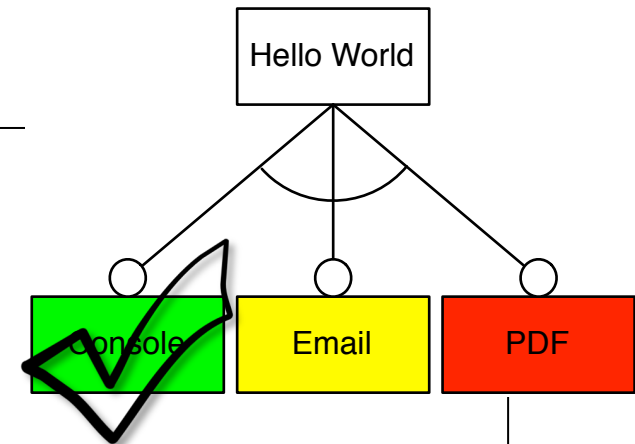
```
</agent>
```



Agent Hello World SPL ...

```
<agent name="HelloWorld">
  <beliefs>
    <belief name="EmailPrinter" />
  </beliefs>

  <goals>
    <goal name="PrintConsole"/>
  </goals>
  <plans>
    <plan name="Print">
      <body class="HelloWorld"/>
      <trigger>
        <achievegoal name="PrintConsole"/>
      </trigger>
    </plan>
  </plans>
</agent>
```



Onde está o erro?

Agent Hello World SPL ...

```
<agent name="HelloWorld">
  <beliefs>
    <belief name="ConsolePrinter" />
    <belief name="EmailPrinter" />
  </beliefs>

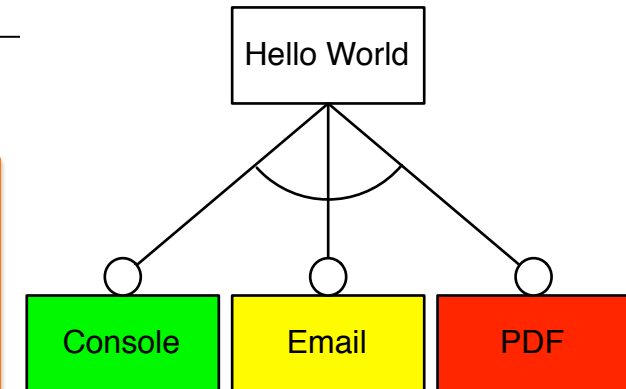
  <goals>
    <goal name="PrintConsole"/>
    <goal name="PrintEmail"/>
  </goals>

  <plans>
    <plan name="Print">
      <body>
        <trig
          </trig
        </plan>
      </plans>
    </agent>
```

```
public class HelloWorld extends Agent {
  public void body() {
    ConsolePrinter printer = getBeliefbase()
      .getBelief("ConsolePrinter").getFact();

    ConsolePrinter printer = getBeliefbase()
      .getBelief("EmailPrinter").getFact();

    ...
  }
}
```

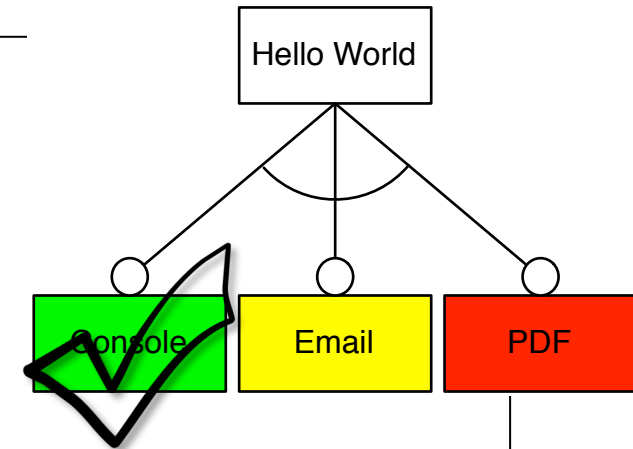


Agent Hello World SPL ...

```
<agent name="HelloWorld">
  <beliefs>
    <belief name="EmailPrinter" />
  </beliefs>

  <goals>
    <goal name="PrintConsole"/>
  </goals>

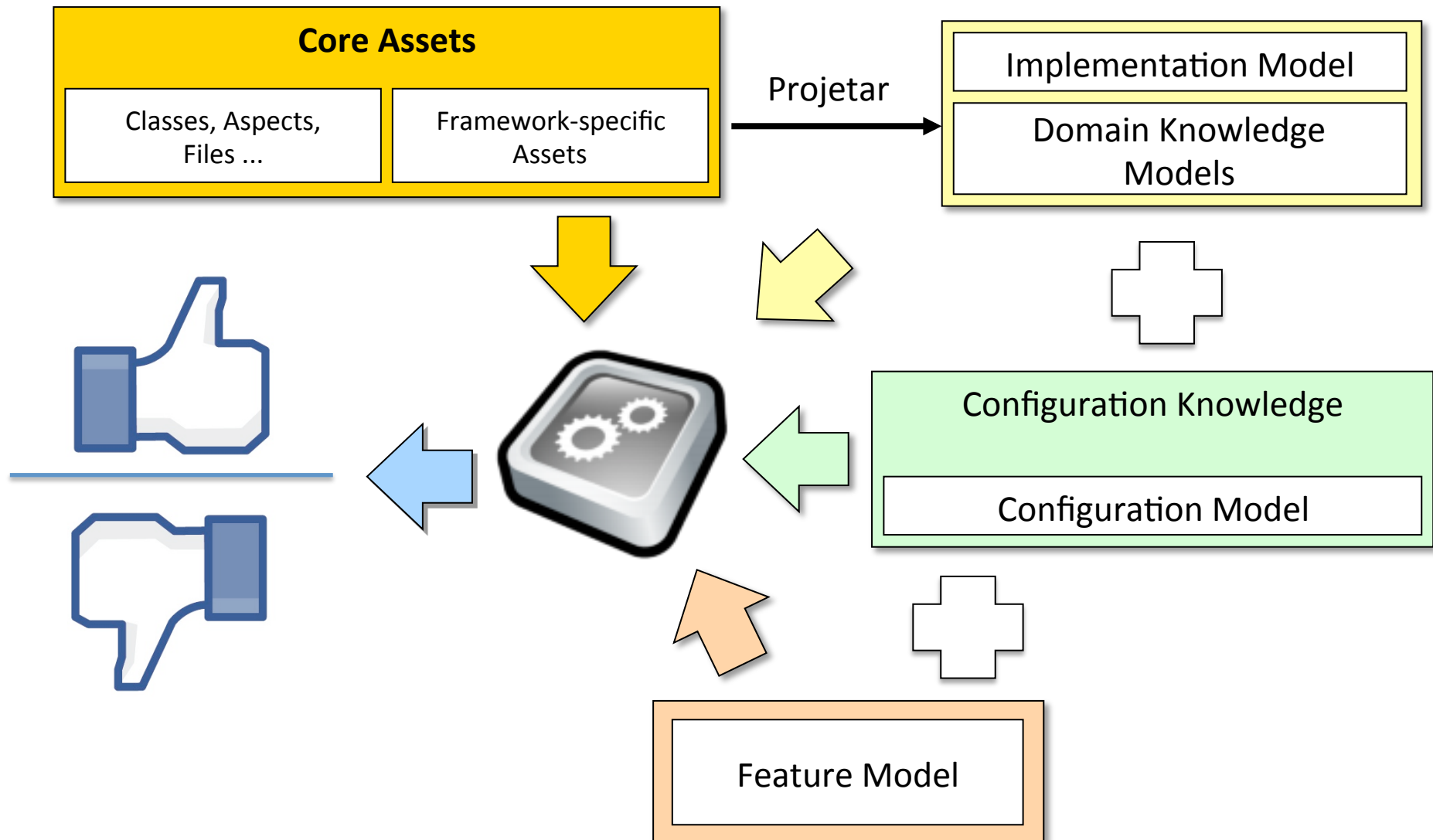
  <plans>
    <plan name="Print">
      <body class="HelloWorld"/>
      <trigger>
        <achievegoal name="PrintEmail"/>
      </trigger>
    </plan>
  </plans>
</agent>
```



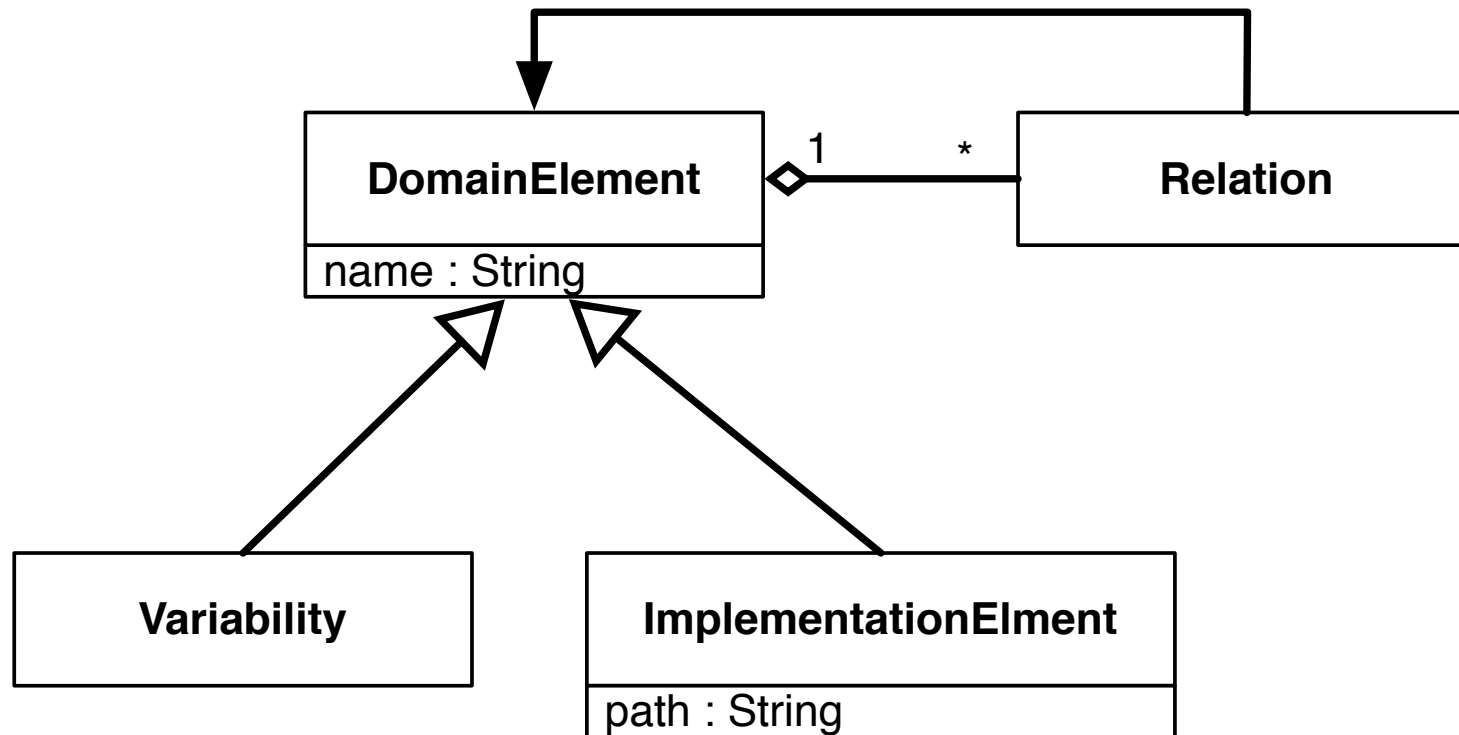
```
public class HelloWorld extends Agent {
  public void body() {
    ✖ ConsolePrinter printer = getBeliefbase()
      .getBelief("ConsolePrinter").getFact();
  }
}
```

Solução

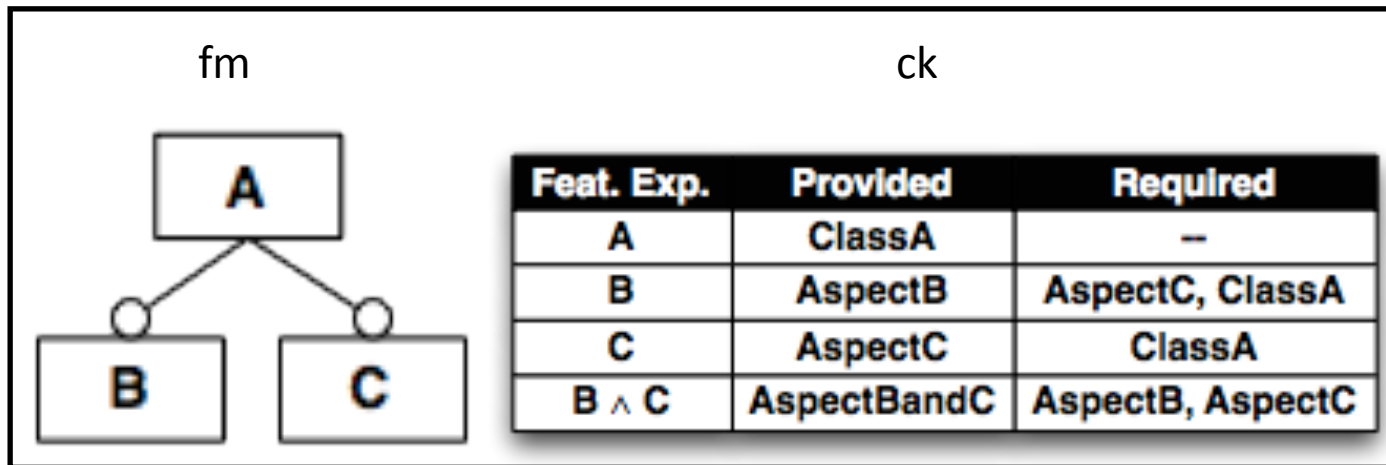
Solução ...



DKML ...



Solução baseada em *Provided* e *Required*



$\forall \text{ product} \in [[\text{fm}]] \mid$
 $\text{required}(\text{product}, \text{ck}) \subseteq \text{provided}(\text{product}, \text{ck})$

$\text{semanticsFM} \Rightarrow \text{constraintsCK}$

Solução baseada em Constraint Programming...

- Conjunto de Variáveis cujo o valor pode ser definido dentro de um domínio
- Conjunto de Constraints
- Solução
 - Atribuição de valores as variáveis obedecendo todas as constraints ...

Definição dos modelos ...

- **Feature Model:**
 - set of features F
- **Architecture Model:**
 - set of model elements E_i
- **Domain Knowledge Model**
 - set of model elements E_a
 - and a set of relation $R_a \subseteq E_a \times E_a$ and $R_i \subseteq E_i \times E_i$
- **Configuration Model**
 - set of $R_C \subseteq F_E \times E_A \cup E_i$

Variáveis ...

- set of variables V where each variable $v \in V$ represents a feature $f \in F$.
- set of variables M representing elements E_A
- And variables $a \in A$ represent elements from the set E_i

Constraints ...

- Mapeamentos de **Features** para **Assets**
 - $(b = 1) \Leftrightarrow (w = 1)$
- Relação entre elementos do **Espaço Solução**
 - *Realization* - $(m_{eA} = 1) \Leftrightarrow (a_{el} = 1)$
 - *Hierarchical* - $(m_{eA} = 1) \Leftrightarrow (a_{el} = 1)$
 - *Referential* - $(m_{eA} = 1) \Leftrightarrow (a_{el} = 1)$

Safe Composition of Framework-based Software Product Lines

Elder Cirilo,
Leopoldo Teixeira, et. al.

Action-based Hello World

```
<struts>
  <action name="Printer" class="printer">
    <result name="success">ConsolePrint.jsp</result>

    <result name="success" type="redirect-action">
      <param name="actionName">EmailPrinter</param>
      <param name="namespace">/</param>
      <param name="email">XYZ@gmail.com</param>
    </result>
  </action>

  <action name="EmailPrinter" class="emailPrinter">
    <result name="success">EmailPrint.jsp</result>
  </action>
</struts>
```

