Copyright law: temporal versions

- US “Digital Millenium Act” and modifications
- goal: in t_x calculate the proper *statutory damage* in case of violation of the copyright taking in consideration all the exceptions and the modifications respect an fact.

17 USC Sec. 504

Remedies for infringement: Damages and profits

Interval of efficacy of the norm	Statutory Damages
[Jan. 1, 1978, March 1, 1989 [	**$250** <= statutoryDamages <= **$10,000**
[March 1, 1989, Dec. 9, 1999 [	**$500** <= statutoryDamages <= **$20,000**
[Dec. 9, 1999, ∞	**$750** <= statutoryDamages <= **$30,000**

Version 1
[Jan. 1, 1978, March 1, 1989 [

(c) **Statutory Damages**. -

The copyright owner may elect an award of statutory damages for infringements in a sum of not less than **$250** or more than **$10,000** as the court considers just.

Version 2
[March 1, 1989, Dec. 9, 1999 [

(c) **Statutory Damages**. -

The copyright owner may elect an award of statutory damages for infringements in a sum of not less than **$500** or more than **$20,000** as the court considers just.

Version 3
[Dec. 9, 1999, ∞

(c) **Statutory Damages**. -

The copyright owner may elect an award of statutory damages for infringements in a sum of not less than **$750** or more than **$30,000** as the court considers just.

Reaction RuleML Representation
Statutory Damage Statute (1)

```
<Atom  key="#StatutoryDamage Statute1"> <!– statute 1-->
    <!-- descriptive  metdata -->
    <meta>  <Atom> <Rel>src</Rel><Ind>CopyrightLaw </Ind>   </meta>
    <!-- qualification  metdata -->
    <qualification>

          <!– note: simplified version of efficacy;  could be also modeled in Reaction RuleML as a
              fluent situation (a changeable  truth property of the world) -->
         <Atom>
            <Rel iri="lrml:efficacy">efficacyDamageStatute</Rel>
            <Interval>
                    < Time type="ruleml:TimeInstant"><Data>1978-01-01</Data></Time>
                    < Time type="ruleml:TimeInstant"><Data>1989-03-01</Data></Time>
             </Interval>
         </Atom>

    </qualification>
    <!– instance -->
    <Rel>statute</Rel>
     <Interval>
            <Ind type="lkif:fee">$250</Ind><Ind type="lkif:fee">$10,000</Ind>
    </Interval>

</Atom>
```

Statutory Damage Statute (2,3)

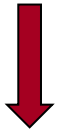
```
<Atom  key="#StatutoryDamage Statute2"> <!– statute 2-->
    <qualification>
          <Atom><Rel> efficacyDamageStatute </Rel>
             <Interval>
                    < Time ><Data> 1989-03-01 </Data></Time>
                    < Time><Data>1999-12-09</Data></Time>
             </Interval>
          </Atom>
     </qualification>
    <Rel>statute</Rel>
     <Interval>
          <Ind type="lkif:fee">$500</Ind><Ind type="lkif:fee">$20,000</Ind>
          <Var/>
    </Interval>
</Atom>
<Atom  key="#StatutoryDamage Statute3"> <!– statute 3-->
    <qualification>
          <Atom><Rel> efficacyDamageStatute </Rel>
             <Interval type="ruleml:LeftClosedInterval">
                    < Time ><Data> 1999-12-09 </Data></Time>
             </Interval>
          </Atom>
     </qualification>
    <Rel>statute</Rel>
     <Interval>
          <Ind type="lkif:fee">$750</Ind><Ind type="lkif:fee">$30,000</Ind>
    </Interval></Atom>
```

RRML: Satutory Damage Rule ...

```
<Rule  style="reasoning"  key="#StatutoryDamagesRule"  >
  <if> <And>
      <Atom><Rel iri="lkif:Infringement">infringes</Rel>
          <Var type="lkif:infringer">Infringer</Var> <Ind>copyright</Ind>
      </Atom>
      <Atom> <!-- note: could be also modeled as an event-->
          <Rel iri="lkif:electStatutoryDamages">electStatutoryDamages </Rel>
          <Var type="lkif:copyrightOwner">CopyrightOwner</Var>
          <Var type="ruleml:TimeInstant">ElectTime</Var>
      </Atom>
      <Atom>
          <scope> <!– scope definition: select knowledge  with metadata  efficacyDamageStatute-->
            <Atom>
              <Rel iri="lrml:efficacy"> efficacyDamageStatute </Rel>
              <Var type="ruleml:TimeInterval">ValidityInterval</Var> <!– binds validity -->
            </Atom>
          </scope>
          <guard><!–check if  election time of award  is during  validity time interval of efficacy -->
                  <Operator  type="ruleml:During">
                      <Var>ElectTime</Var> <Var>ValidityInterval</Var>
                  </Operator>
          </guard>
          <Rel>statute</Rel><!– apply scoped literal only on selected scope which fulfills guard -->
          <Interval><Var >MinAmount</Var><Var >MaxAmount</Var> </Interval>
      </Atom>
  </And></if>  ....
```

RRML: ... Satutory Damage Rule

```
    <then>
      <Atom>
            <Rel iri="lkif:payFee">pay Fee</Rel>
            <Var type="lkif:infringer">Infringer</Var>
            <Var type="lkif:copyrightOwner">CopyrightOwner</Var>
            <Interval><Var >MinAmount</Var><Var >MaxAmount</Var> </Interval>
      </Atom>
    </then>
</Rules>
```



Reaction RuleML Translator Framework

Prova 3 representation with @metadata @scopes and guards[].

```
@efficacyDamageStatute([date(1978,1,1), date(1989,3,1)]) statute([250,10000]).
@efficacyDamageStatute([date(1989,3,1), date(1999,12,9)]) statute([500,20000]).
@efficacyDamageStatute([date(1999,12,9), _]) statute([750,30000]).

@label("StatutoryDamageRule") @src("CopyrightLaw")
payFee(Infringer,CopyrightOwner,[MinAmount, MaxAmount]) :-
      infringes(Infringer,copyright),
      electStatutoryDamages(CopyrightOwner,ElectTime),
      @efficacyDamageStatute(ValidityInterval)
      statute([MinAmount, MaxAmount]) [during(ElecTime,ValidityInterval)].
```

(for more information about metadata scopes and guards see the Prova 3 rule engine http://prova.ws)

Reaction RuleML – Key Message from this Copyright Law example

- Support for **Life Cycle Management**
  - descriptive metadata <*meta*>
  - qualifying metadata <*qualification*>
- Support for **Modularization** and highly efficient **Dynamic Views** on the Knowledge Base
  - **scoped reasoning** <*scope*>
    - global knowledge in the KB becomes closed **local knowledge** in a scope onn which reasoning and processing can be done efficiently
    - scopes are at the heart of Reaction RuleML features for modularization, windowing techniques, selection and consumption policies / life cycle mangagement, … it is all about **dynamic knowledge**