

## collecting\_society - Datenbank #665

### New objects

10/14/2018 06:03 AM - Alexander Blum

<b>Status:</b>	Erledigt	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Meik Michalke	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour

#### Description

As a result of a talk with meik, we need the following new objects:

- Tariff Category
  - needed to be compatible with event registry later on
  - see <https://gpad.c3s.cc/p/core/eventregistry%20database>
  - fields
    - name [Char] {Textfield} ("concert", "djset", ...)
    - basic\_value [Float] {Textfield}
    - additional\_variable [Float]
    - (later: version [-1:n-> TARIFF\_VERSION])
    - (later: -n:n-> tariff category to track the relations of different sets of categories)
- Collecting Society
  - no foreign object creation (data administration by admins)
  - Neighbouring Rights Societies are also Collecting Societies
  - needed to hold contract data, etc. later on
  - fields
    - party
    - copyright: Checkbox
    - ancillary copyright: Checkbox
- Producer
  - kind of an artist
  - foreign object creation fields: name
  - might get a piece of the distribution cake
  - needed to hold account data later on
  - fields
    - party
- Publisher
  - foreign object creation fields: name
  - might get a piece of the distribution cake
  - needed to hold account data later on
  - fields
    - party
- Performance
  - performance now is tracked by a creation-artist relation (attribute: contribution type = performance)
  - to consider ancillary copyright in future, we also must know, who represents those rights (a CS with checkbox ancillary copyright)
  - types of performance are at least: instrumental play, production
  - fields
    - performer (artist | producer)
    - type (instrumental | production)
- Copyright Owners
  - no object itself, but a pointer to (e.g. the parties of) n Labels, n Publisher, n Artists

Further as strings or maybe objects, if we want to consider them stakeholders as well:

- Mixer (party, foreign object creation: name)
- Master (party, foreign object creation: name)

Additional relations:

- Collecting Society
  - -n:n-> Tariff Category

- -1:n-> Performance
- Release
  - -n:1-> Label
  - -n:1-> Publisher
  - -n:n-> Producer
  - -n:n-> Party (Copyright Owner)
- Creation
  - -n:n-> Tariff Category

In words:

- Collecting Societies (copyright) are connected to Creations via Tariff Categories
- Collection Societies (ancillary copyright) are connected to Creations via Performances

#### Related issues:

Related to collecting\_society - Konzept #691: Usecases and structure for copy...

**Erledigt**

Related to collecting\_society - Konzept #692: Do we need/want release contrib...

**Erledigt**

#### Associated revisions

##### Revision eb29fd38 - 10/15/2018 02:04 AM - Alexander Blum

ref #665: adds performance type

##### Revision 3d3b8a45 - 10/15/2018 12:26 PM - Alexander Blum

ref #665: adds tariff objects

##### Revision 56663786 - 10/15/2018 01:05 PM - Alexander Blum

ref #665: adds publisher object

##### Revision 75876e87 - 10/16/2018 07:28 PM - Alexander Blum

ref #665: adds content types, collecting society, other improvements

#### History

##### #1 - 10/14/2018 07:01 AM - Alexander Blum

- Status changed from Neu to Feedback

- Assignee set to Meik Michalke

## Refinement

### Make Producer (and Master, Mixer, etc. (?)) just Artists

- The main reason for dedicated objects would be to provide accounting (which artists should implement anyway)
- It would be easy to extend the list of possible contributors, as it is only a relation and not a dedicated object
- It feels right not to judge about the value of the creative contribution on a structural level
- It makes the performance object easier, as it only needs a reference to an artist (and not to different types of objects)
- The types of performance then would be: play (in need of a better word?), production, master, mix, etc
- The distribution algorithm would be easier (just switch types of performances in a for loop)
- There would even be no need to save the category of an Artist within the Artist, as the role is defined by the relation
- The special case of an Artist member to take those roles would also be trivial then

### Integrate Performance into Contribution

- Contribution is a more abstract concept of performance, linking a Creation to an Artist ("an artist contributes to a creation")
  - Contribution types are performance, composition, text
  - Roles might further specify the role of the artist concerning a contribution type (e.g. guitar, vocals, etc)
  - Most field are the same, only Collection Society and Performance Type needs to be added
  - Performance types would be play, master, mixing, etc.
- The ontology would be simpler
- The distribution algorithm would be easier (just loop over all contributions)

## Questions

- Does the role list need to be contribution type dependent (different sets of roles for different types)?
  - What roles would make sense for contribution type: text?
  - What roles would make sense for contribution type: performance and performance type: production, master, mixing, etc?
  - Are the roles for composition and performance (type: play) the same?
  - Shouldn't there be only one composition for one creation?
- Should the user be able to insert new roles?

#2 - 10/15/2018 12:52 PM - Alexander Blum

## Adjustments for Tariff objects

After some thinking, I implemented this (italics are TODOs for later, when we implement the event registry):

- **TariffSystem**
  - code (autosequence, e.g. T00000001)
  - version (free text, e.g. 1.0)
  - valid\_from (date)
  - valid\_through (date)
  - transitional\_through (date)
  - tariffs -1:n-> Tariff
  - *attachment* -1:n-> Attachment
  - *formular* (function name)
  - *variables* (float, e.g. basic value, additional variable)
- **TariffCategory**
  - name (free text, e.g. 'Live')
  - code (free text, e.g. 'L')
  - description (free text)
  - tariffs -1:n-> Tariff
- **Tariff**
  - name (derived: category.name, e.g. 'Live')
  - code (derived, equals category.name + system.version, e.g. 'L1.0')
  - system -n-1-> TariffSystem
  - category -n-1-> TariffCategory
  - *variables* (float, e.g. relevance, contribution)
- **CreationTariffCategory** (would be RightTransfer a better word?)
  - creation -n:1-> Creation
  - category -n:1-> TariffCategory
  - collecting\_society -n:1-> CollectingSociety

In words, it is now possible to

- create versions of the tariff system ("different tariff systems")
- create new tariff categories
- specify concrete tariffs (implementing a tariff category by specifying the concrete value of the variables)
- specify a collecting society for each tariff category of each creation

Feedback would be appreciated.

## Questions

- base value is a property of the tariff system?
- what was the concept of additional variable? Was this just a placeholder for the variables relevance, contribution, etc?

**#3 - 10/15/2018 02:16 PM - Alexander Blum**

We might need to think again about `copyright_owners`.

- What are the usecases for this field?
- Currently it should be a pointer to the involved parties
- It might also include artwork, etc.
- Should it include all copyright owners of the creations (e.g. composition, text)?

Maybe we need also a `ReleaseContribution` object, where different roles might be specified in analogy to `CreationContribution` (e.g. artwork, but also e.g. mixing, which could propagate down to the creations).

Until the usecases for this field and it's implications for releases/creations/distributions are clear, I omit the copyright owner field.

**#4 - 11/05/2018 03:21 PM - Alexander Blum**

- *Related to Konzept #691: Usecases and structure for copyright owners added*

**#5 - 11/05/2018 03:32 PM - Alexander Blum**

- *Related to Konzept #692: Do we need/want release contributions? added*

**#6 - 11/05/2018 03:34 PM - Alexander Blum**

- *Status changed from Feedback to Erledigt*

- *% Done changed from 0 to 100*

The remaining open issues are split into [#691](#) and [#692](#)

**#7 - 10/08/2019 03:52 PM - Alexander Blum**

- *Project changed from repertoire to collecting\_society*