

## Tutorial: CLARIN Component Registry und Browser für CMDI

### Inhalt

1. Einleitung.....	2
2. Zugang.....	2
3. Browseransicht im Überblick .....	4
Kernfunktionalitäten.....	4
Ansicht: Profile vs. Komponenten .....	5
Public Space vs. Work Space .....	5
Verwendungszweck.....	5
Browse-Modus .....	5
Detailansicht: View-Modus vs. XML-Modus.....	5
4. Public Space .....	7
5. Work Space .....	7
6. Profile .....	8
Auswahl.....	8
Ansicht im View-Modus .....	8
Ansicht im XML-Modus .....	10
Download .....	10
Profile im Work Space öffentlich zur Verfügung stellen.....	11
7. Komponenten.....	12
Auswahl.....	12
Datenkategorien.....	13
Name .....	13
Datentyp .....	14
Kontrolliertes Vokabular .....	14
Attribute .....	15
Concept Link.....	17
Display Priority .....	18
Number of Occurrences.....	18
Multilingual (xml:lang) .....	18
Komponenten innerhalb von Komponenten .....	18
Download .....	19
8. Funktionalitäten .....	20
a. Browse: Einsicht und Download von Komponenten/Profilen .....	20
b. Edit: Erstellen, Bearbeiten und Löschen von Komponenten/Profilen .....	20
Edit Item: Bearbeiten vom Komponenten und Profilen.....	21
Edit Item vs. Edit-Tab .....	22
Start Over .....	22
Editieren: Public Space vs. Work Space.....	23
Löschen von Komponenten und Profilen.....	23
Edit-Tab: Erstellen und Bearbeiten von Komponenten und Profilen.....	24
Elemente.....	26



Löschen und Festlegen der Reihenfolge .....	26
Name und Concept Link .....	26
Documentation.....	27
Display Priority.....	27
Min Occurrences und Max Occurrences.....	27
Type .....	28
Primitive Datentypen .....	28
Pattern .....	28
Kontrolliertes Vokabular .....	29
Multilingual (xml:lang).....	31
Attribute.....	32
Hinzufügen weiterer Elemente .....	33
Komponenten .....	33
Erstellen neuer und Hinzufügen bestehender Komponenten (+component-Funktion) .....	33
Einbindung existierender Komponenten (Drag and Drop-Funktion) .....	34
Speichern von Komponenten und Profilen.....	35
Save .....	36
Save as new.....	36
Publish in public space .....	37
c. Import: Importieren lokal erstellter Komponenten/Profile .....	37
9. Referenzen.....	39
10. Kontakt.....	40

## 1. Einleitung

Die CLARIN Component Registry enthält sowohl Profile als auch Komponenten für die Erstellung von Metadateninstanzen im CMDI-Format (CLARIN Component Metadata Infrastructure). Der Nutzer hat die Möglichkeit,

- bereits registrierte Profile/Komponenten einzusehen und herunterzuladen (Komponenten und Profile stehen im XML-Format sowie Schemas der Profile als XML Schema zur Verfügung),
- neue Profile/Komponenten innerhalb der Registry zu erstellen und zu editieren, sowie
- eigene Profile/Komponenten zu importieren.

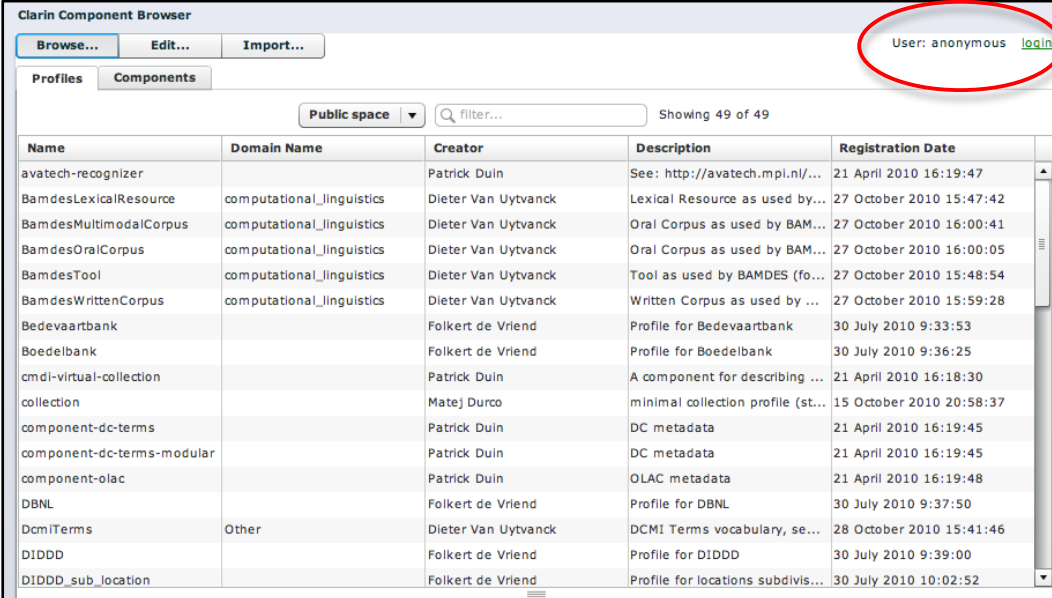
Die Component Registry wird zur Zeit noch stetig weiterentwickelt, sodass weiterhin mit Änderungen und Erweiterungen der Funktionalitäten zu rechnen ist.

## 2. Zugang

Es bestehen zwei Möglichkeiten, um Zugang zur Component Registry zu erhalten:

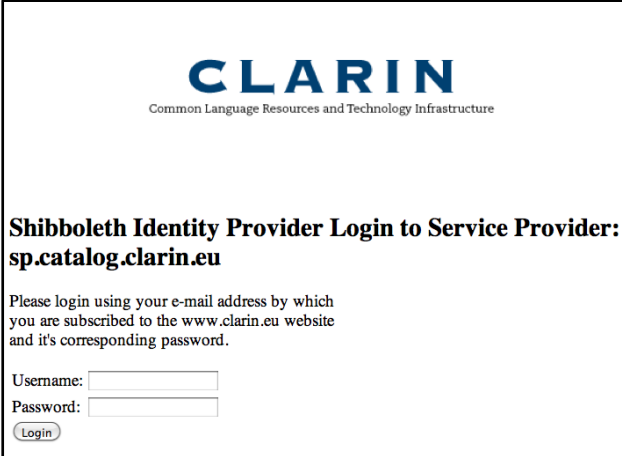
- über die Clarin-Webseite zu CMDI: <http://www.clarin.eu/cmdi> (hier dem Link unter *Component Registry* folgen), oder
- direkt über: <http://catalog.clarin.eu/ds/ComponentRegistry/#> .

Wenn man lediglich bereits registrierte Profile und Komponenten einsehen oder herunterladen möchte, reicht es aus, einen einfachen anonymen Zugang ohne Login zu verwenden (s. Abb. 1). Um jedoch auch Funktionalitäten wie das Erstellen, Editieren oder Importieren von Komponenten und Profilen nutzen zu können, ist ein Login erforderlich (s. Abb. 1). Als Logindaten werden hier die Zugangsdaten der Clarin-Webseite (<http://www.clarin.eu>) verwendet (s. Abb. 2). Sollte ein Nutzer nicht über diese Zugangsdaten verfügen, ist eine Registrierung auf der Clarin-Webseite erforderlich (<http://www.clarin.eu/user/register>).



Name	Domain Name	Creator	Description	Registration Date
avatech-recognizer		Patrick Duin	See: <a href="http://avatech.mpi.nl/...">http://avatech.mpi.nl/...</a>	21 April 2010 16:19:47
BamdesLexicalResource	computational_linguistics	Dieter Van Uytvanck	Lexical Resource as used by...	27 October 2010 15:47:42
BamdesMultimodalCorpus	computational_linguistics	Dieter Van Uytvanck	Oral Corpus as used by BAM...	27 October 2010 16:00:41
BamdesOralCorpus	computational_linguistics	Dieter Van Uytvanck	Oral Corpus as used by BAM...	27 October 2010 16:00:05
BamdesTool	computational_linguistics	Dieter Van Uytvanck	Tool as used by BAMDES (fo...	27 October 2010 15:48:54
BamdesWrittenCorpus	computational_linguistics	Dieter Van Uytvanck	Written Corpus as used by ...	27 October 2010 15:59:28
Bedevaartbank		Folkert de Vriend	Profile for Bedevaartbank	30 July 2010 9:33:53
Boedelbank		Folkert de Vriend	Profile for Boedelbank	30 July 2010 9:36:25
cmdi-virtual-collection		Patrick Duin	A component for describing ...	21 April 2010 16:18:30
collection		Matej Durco	minimal collection profile (st...	15 October 2010 20:58:37
component-dc-terms		Patrick Duin	DC metadata	21 April 2010 16:19:45
component-dc-terms-modular		Patrick Duin	DC metadata	21 April 2010 16:19:45
component-olac		Patrick Duin	OLAC metadata	21 April 2010 16:19:48
DBNL		Folkert de Vriend	Profile for DBNL	30 July 2010 9:37:50
DcmITerms	Other	Dieter Van Uytvanck	DCMI Terms vocabulary, se...	28 October 2010 15:41:46
DIDDD		Folkert de Vriend	Profile for DIDDD	30 July 2010 9:39:00
DIDDD_sub_location		Folkert de Vriend	Profile for locations subdivis...	30 July 2010 10:02:52

Abb. 1: Anonymer Zugang und Link zum Login



**CLARIN**  
Common Language Resources and Technology Infrastructure

**Shibboleth Identity Provider Login to Service Provider:  
sp.catalog.clarin.eu**

Please login using your e-mail address by which you are subscribed to the [www.clarin.eu](http://www.clarin.eu) website and it's corresponding password.

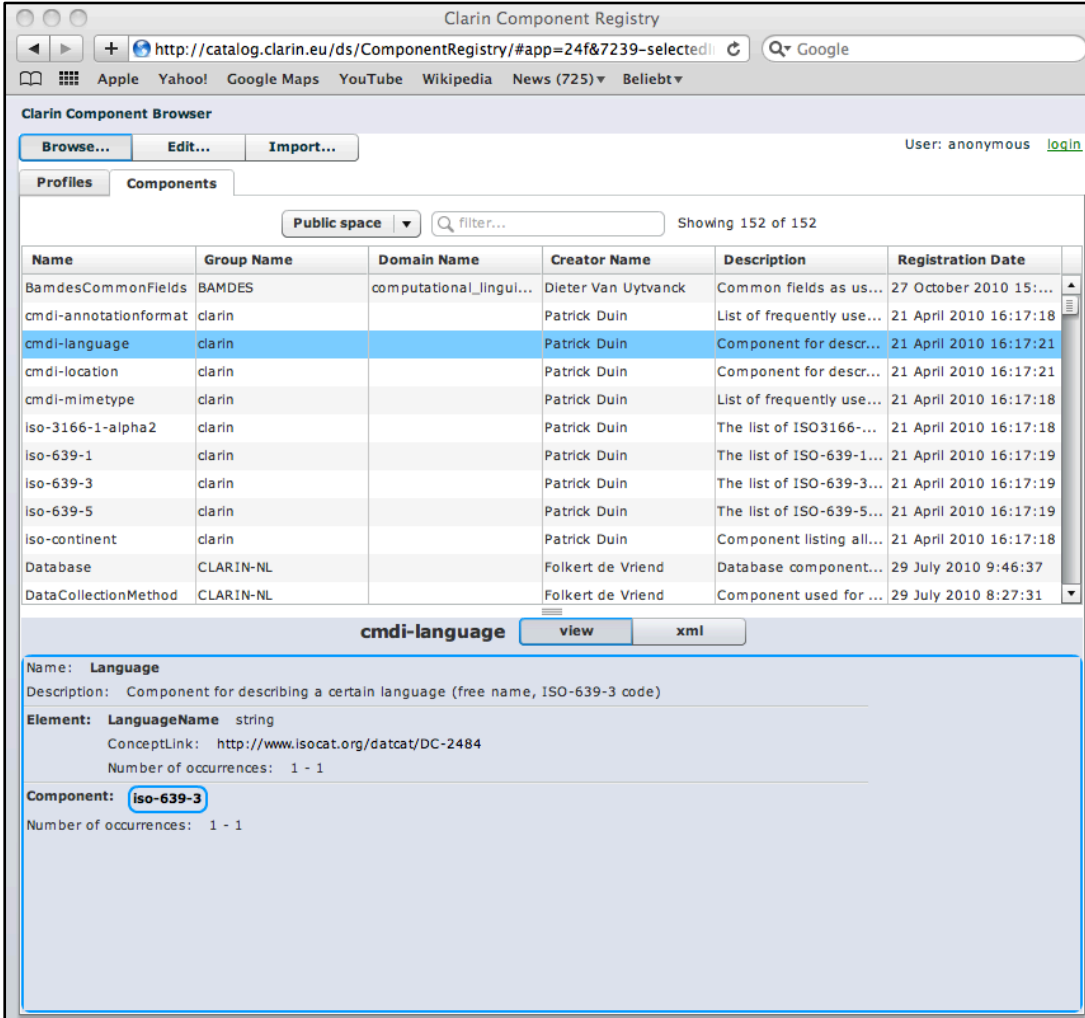
Username:

Password:

Abb. 2: Login Component Registry

### 3. Browseransicht im Überblick

In der Browseransicht der CLARIN Component Registry stehen dem Nutzer unterschiedliche Darstellungsoptionen und Funktionalitäten zur Verfügung. Im Folgenden wird eine allgemeine Gesamtübersicht über diese Aspekte gegeben, wobei sie im Einzelnen in den nachfolgenden Inhaltspunkten detaillierter erläutert werden.



Name	Group Name	Domain Name	Creator Name	Description	Registration Date
BamdesCommonFields	BAMDES	computational_lingul...	Dieter Van Uytvanck	Common fields as us...	27 October 2010 15:...
cmdi-annotationformat	clarin		Patrick Duin	List of frequently use...	21 April 2010 16:17:18
cmdi-language	clarin		Patrick Duin	Component for descr...	21 April 2010 16:17:21
cmdi-location	clarin		Patrick Duin	Component for descr...	21 April 2010 16:17:21
cmdi-mimetype	clarin		Patrick Duin	List of frequently use...	21 April 2010 16:17:18
iso-3166-1-alpha2	clarin		Patrick Duin	The list of ISO3166-...	21 April 2010 16:17:18
iso-639-1	clarin		Patrick Duin	The list of ISO-639-1...	21 April 2010 16:17:19
iso-639-3	clarin		Patrick Duin	The list of ISO-639-3...	21 April 2010 16:17:19
iso-639-5	clarin		Patrick Duin	The list of ISO-639-5...	21 April 2010 16:17:19
iso-continent	clarin		Patrick Duin	Component listing all...	21 April 2010 16:17:18
Database	CLARIN-NL		Folkert de Vriend	Database component...	29 July 2010 9:46:37
DataCollectionMethod	CLARIN-NL		Folkert de Vriend	Component used for ...	29 July 2010 8:27:31

**cmdi-language** view xml

Name: **Language**  
Description: Component for describing a certain language (free name, ISO-639-3 code)

Element: **LanguageName** string  
ConceptLink: <http://www.isocat.org/datcat/DC-2484>  
Number of occurrences: 1 - 1

Component: **iso-639-3**  
Number of occurrences: 1 - 1

Abb. 3: Browseransicht im Gesamtüberblick

#### Kernfunktionalitäten

In der in Abb. 3 dargestellten Ansicht sind zunächst die drei Kernfunktionalitäten des Browsers zu erkennen: *Browse*, *Edit* und *Import*. Die Browse-Funktion gestattet dem Nutzer auch ohne Login die Einsicht und den Download der bereits registrierten Komponenten und Profile (s. Inhaltspunkt 8a). Der Edit-Modus ermöglicht im registrierten Zugang das Erstellen und Editieren von Komponenten und Profilen (s. Inhaltspunkt 8b). Die dritte Funktionalität, dargestellt durch den Import-Tab, erlaubt



ebenfalls nach erfolgreichem Login das Importieren von Komponenten und Profilen in die Component Registry (s. Inhaltspunkt 8c). Durch die Anordnung der Funktionalitäten in drei unterschiedliche Tabs kann der Nutzer durch Anklicken des entsprechenden Tabs zwischen den Optionen wechseln. Als Defaultansicht fungiert dabei die Browse-Funktionalität.

### Ansicht: Profile vs. Komponenten

Unterhalb dieser drei Kernfunktionalitäten sind im Browser zwei weitere Tabs als Auswahlmöglichkeiten dargestellt, die mit *Profiles* und *Components* betitelt sind. Sie bieten dem Nutzer die Möglichkeit, zwischen der Überblicksansicht der Profile (s. Inhaltspunkt 6) und der Überblicksansicht der Komponenten (s. Inhaltspunkt 7) zu wechseln.

### Public Space vs. Work Space

Oberhalb der tabellarischen Übersicht über die Profile oder Komponenten finden sich drei weitere Optionen. Erstens wird dem Nutzer ein sogenanntes Dropdown-Menü angezeigt, das eine Ansicht der Profile/Komponenten im *Public Space* (s. Inhaltspunkt 4) oder im *Work Space* (s. Inhaltspunkt 5) zur Auswahl stellt. Zweitens ist ein Suchfenster vorhanden, das ein Filtern der Profile/Komponenten ermöglicht. Drittens wird angezeigt, wie viele Profile bzw. Komponenten aktuell im jeweiligen gewählten Bereich (d.h. Public Space oder Work Space) in der Component Registry registriert sind (s. Abb. 3: *Showing 152 of 152*).

### Verwendungszweck

Der eigentliche Zweck des Browsers ist neben den erläuterten Kernfunktionalitäten ebenso das nachhaltige Speichern von Profilen/Komponenten mittels einer Vergabe von persistenten Identifikatoren (d.h. *Persistent Identifiers*, PIDs) wie die Einsicht der registrierten Komponenten und Profile zur Erstellung von Metadateninstanzen in CMDI. Solche Instanzen können dabei entweder vom Nutzer per Hand im XML-Format selbst erstellt werden oder mit Hilfe des Metadateneditors *Arbil* (Zugang über <http://www.clarin.eu/cmdi>: *Start Link* anklicken unter *Arbil metadata editor with CMDI support*), der die im Public Space der Component Registry registrierten Profile und Komponenten automatisch einbindet.

### Browse-Modus

Für die Einsicht der Komponenten/Profile dient dem Nutzer die tabellarische Übersicht im Browse-Modus mit der bereits erwähnten Suchfunktion (s. Abb. 3). Ein Profil oder eine Komponente kann per Anklicken ausgewählt werden (s. Abb. 3: blau unterlegte Komponente *cmdi-language*) und erscheint dann in einer Detailansicht unterhalb der Tabelle (s. Abb. 4).

### Detailansicht: View-Modus vs. XML-Modus

In dieser Detailansicht werden im als Default gesetzten interaktiven View-Modus zunächst allgemeine Informationen zur Komponente angegeben (z.B. Name und Beschreibung) sowie die enthaltenen Elemente und weiteren Komponenten dokumentiert (Details s. Inhaltspunkte 6 und 7). Enthaltene Komponenten sind

außerdem im Public Space an einer blauen Umrandung des Namens zu erkennen (s. Komponente *iso-639-3* in Abb. 3), den man anklicken kann, sodass sich an dieser Stelle ein weiteres Fenster mit den Inhalten dieser eingebundenen Komponente öffnet (s. Abb. 4).

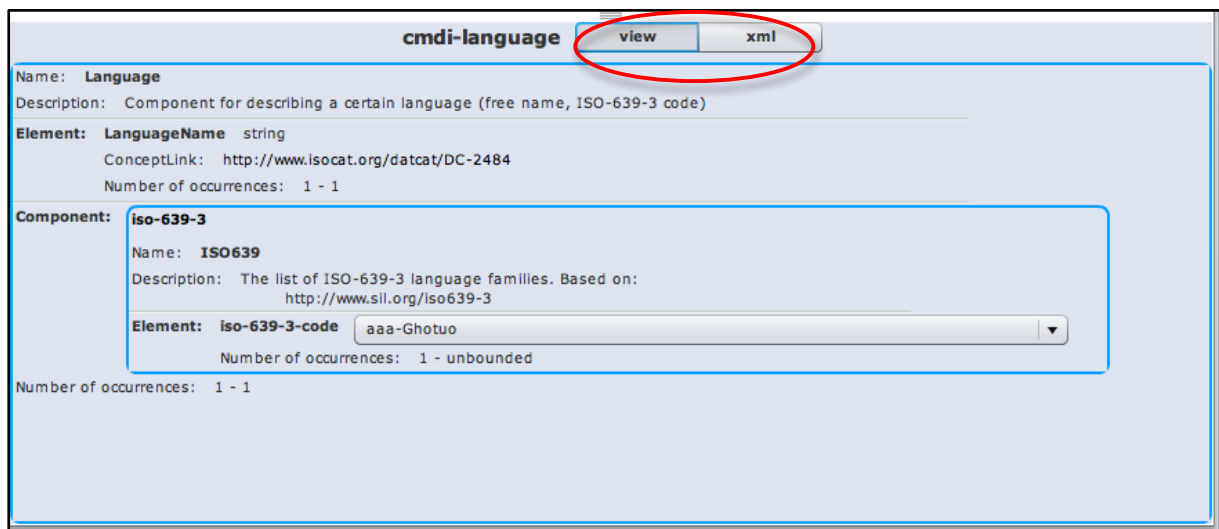


Abb. 4: Detailansicht einer Komponente im View-Modus

Analog zu der in Abb. 4 angezeigten Ansicht einer Komponente würde in Hinblick auf den Aufbau auch eine Detailansicht für Profile aussehen. Neben dem View-Modus ist ebenfalls eine Ansicht der Komponente im XML-Format durch Auswahl des Tabs *xml* (s. Abb. 4) möglich. Diese Ansichtform ist besonders hilfreich für Nutzer, die ihre Metadateninstanzen selbst im XML-Format erstellen und neue Profile und Komponenten in CMDI entwickeln, da beispielsweise nur im XML-Modus die PID einer Komponente oder eines Profils sichtbar ist (s. Abb. 5, Element *<ID>*).



Abb. 5: Detailansicht einer Komponente im XML-Modus

Die einzelnen bereits genannten Funktionalitäten und Darstellungsoptionen der Component Registry werden im Folgenden genauer erläutert.

## 4. Public Space

Der *Public Space* ist der Bereich in der Component Registry, in dem Profile und Komponenten öffentlich zugänglich sind. Dies bedeutet, dass sie registriert sind (d.h. sie besitzen eine PID) und durch den öffentlichen Status von anderen Nutzern durch die Möglichkeit zur Einsicht und zum Download wiederverwendet werden können. Nach dem momentanen Entwicklungsstand erfordert eine solche Wiederverwendung jedoch, dass Profile und Komponenten nach einer einmal erfolgten Veröffentlichung nicht mehr abgeändert werden können, da andernfalls auf ihnen basierende Metadateninstanzen möglicherweise nicht mehr valide wären. Für einen Zugriff auf den Public Space reicht es aus, den anonymisierten Zugang zur Component Registry zu verwenden. Der Public Space wird in der Component Registry in einem Dropdown-Menü angezeigt (s. Abb. 6), in dem man neben dem öffentlichen Bereich außerdem den privaten *Work Space* auswählen kann, der dem Nutzer jedoch nur nach einem erfolgten Login zugänglich ist.

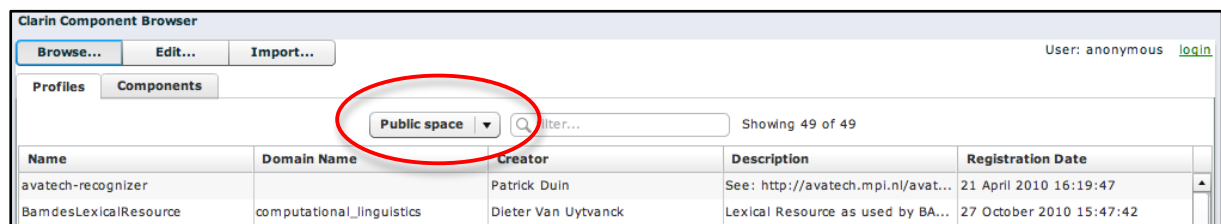


Abb. 6: Public Space

## 5. Work Space

Der *Work Space* (s. Abb. 7) ist der Bereich in der Component Registry, in dem Profile und Komponenten noch nicht der Öffentlichkeit zur Verfügung stehen, sondern nur privat für den erstellenden registrierten Nutzer nach einem erfolgten Login zugänglich sind (s. Abb. 7: Angabe des Users ist nicht mehr wie im Public Space *anonymous*, sondern *nalida@...*). Dieser private Bereich dient vor allem dazu, dass sich im Entwicklungsprozess befindliche Profile und Komponenten noch abgeändert werden können, weil dies nach einer einmal erfolgten Veröffentlichung nicht mehr möglich ist. Dennoch werden Profile/Komponenten ebenso wie im öffentlichen Bereich der Component Registry auch im Work Space registriert und erhalten eine PID, bleiben jedoch zunächst nur privat zugänglich. Für den registrierten Nutzer ist ebenfalls die Einsicht und das Herunterladen seiner eigenen Profile/Komponenten im Work Space möglich. Dennoch besteht außerdem die Möglichkeit, Schemas von Profilen im privaten Bereich über eine URL auch öffentlich zur Verfügung zu stellen. Dies ist besonders zu Präsentationszwecken im Entwicklungsprozess hilfreich. Details zu dieser Funktionalität finden sich im Kontext von Profilen unter Inhaltspunkt 6.



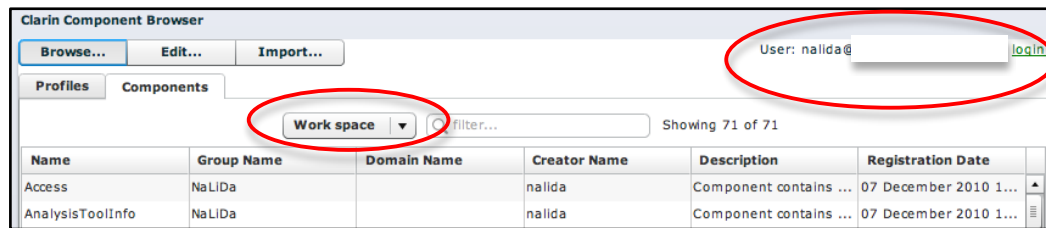


Abb. 7: Work Space

## 6. Profile

### Auswahl

Die Übersicht über Profile in der Component Registry kann, wie bereits erläutert, über den Tab *Profiles* ausgewählt werden. Sobald ein Profil durch Anklicken ausgewählt wurde, wie beispielsweise das *TextCorpusProfile* (s. Abb. 8), erscheint unterhalb der tabellarischen Übersicht eine Detailansicht des Profils.

### Ansicht im View-Modus

Die in diesem Profil enthaltenen Komponenten sind im Public Space in der Detailansicht im View-Modus blau umrandet (s. Abb. 8).

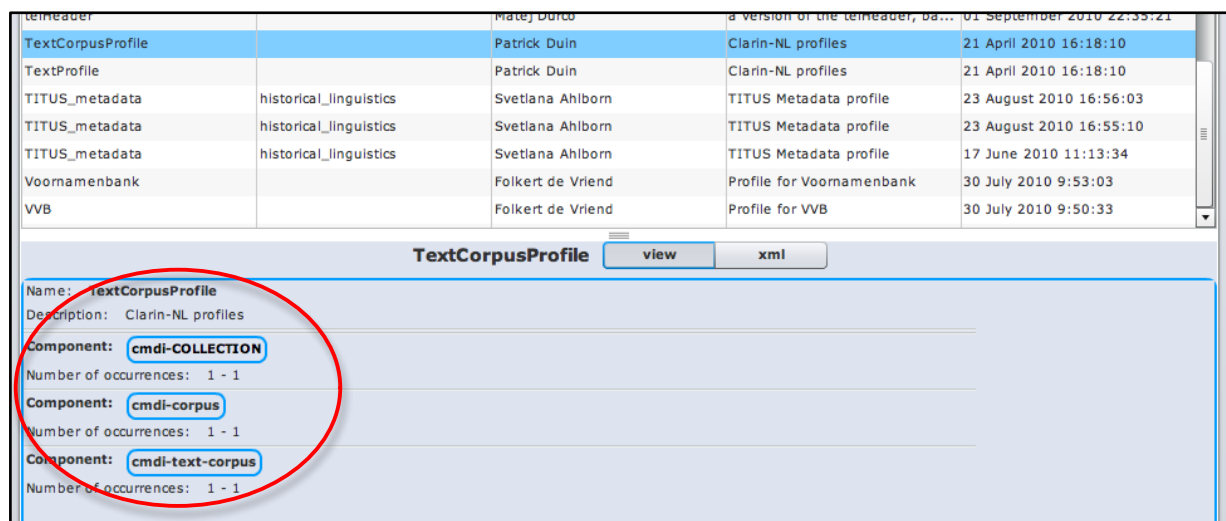


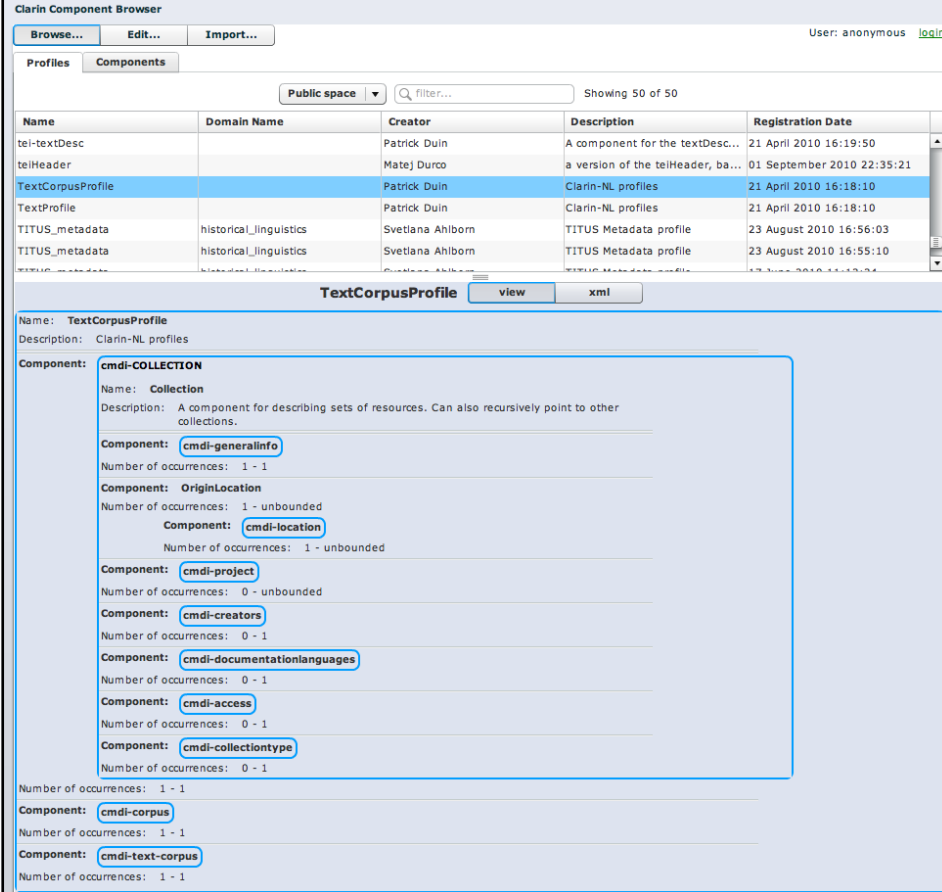
Abb. 8: Detailansicht eines Profils im View-Modus

Wenn man nun den umrandeten Namen einer Komponente wie z.B. *cmdi-COLLECTION* (s. Abb. 8) anklickt, wird diese Komponente expandiert und man kann ihre enthaltenen Inhalte einsehen (s. Abb. 9: Komponente *cmdi-COLLECTION*).

Wenn sich ein Profil oder eine Komponente noch im privaten Work Space befindet und eine weitere Komponente einbindet, die ebenfalls noch nicht öffentlich zugänglich ist, findet man statt einer blauen Umrandung der Komponente eine rote Umrandung vor. Das zugrunde liegende Prinzip ist hier identisch zu dem zuvor erläuterten, d.h. der Inhalt einer Komponente kann durch Anklicken des Namens expandiert werden. Die andere Farbe der Umrandung soll lediglich den noch nicht veröffentlichten Status der



Komponente signalisieren. Hervorzuheben ist, dass diese Vorgehensweise sowohl für Komponenten innerhalb von Profilen als auch innerhalb anderer Komponenten gültig ist.



The screenshot shows the 'Clarin Component Browser' interface. At the top, there are buttons for 'Browse...', 'Edit...', and 'Import...'. The user is identified as 'anonymous' with a 'login' link. The main area is divided into 'Profiles' and 'Components' tabs. A dropdown menu shows 'Public space' and a search filter. Below this is a table of components:

Name	Domain Name	Creator	Description	Registration Date
tei-textDesc		Patrick Duin	A component for the textDesc...	21 April 2010 16:19:50
teiHeader		Matej Durco	a version of the teiHeader, ba...	01 September 2010 22:35:21
TextCorpusProfile		Patrick Duin	Clarin-NL profiles	21 April 2010 16:18:10
TextProfile		Patrick Duin	Clarin-NL profiles	21 April 2010 16:18:10
TITUS_metadata	historical_linguistics	Svetlana Ahlborn	TITUS Metadata profile	23 August 2010 16:56:03
TITUS_metadata	historical_linguistics	Svetlana Ahlborn	TITUS Metadata profile	23 August 2010 16:55:10

Below the table, the 'TextCorpusProfile' component is expanded in 'view' mode. The details are as follows:

- Name:** TextCorpusProfile
- Description:** Clarin-NL profiles
- Component:** **cmdi-COLLECTION**
  - Name:** Collection
  - Description:** A component for describing sets of resources. Can also recursively point to other collections.
  - Component:** cmdi-generalinfo (Number of occurrences: 1 - 1)
  - Component:** OriginLocation (Number of occurrences: 1 - unbounded)
    - Component:** cmdi-location (Number of occurrences: 1 - unbounded)
  - Component:** cmdi-project (Number of occurrences: 0 - unbounded)
  - Component:** cmdi-creators (Number of occurrences: 0 - 1)
  - Component:** cmdi-documentationlanguages (Number of occurrences: 0 - 1)
  - Component:** cmdi-access (Number of occurrences: 0 - 1)
  - Component:** cmdi-collectiontype (Number of occurrences: 0 - 1)
- Number of occurrences: 1 - 1
- Component:** cmdi-corpus (Number of occurrences: 1 - 1)
- Component:** cmdi-text-corpus (Number of occurrences: 1 - 1)

Abb. 9: Expandierte Detailansicht eines Profils im View-Modus

Ein solches Beispiel für eine Komponente im privaten Bereich der Component Registry ist in Abb. 10 für die Komponente *DocumentationLanguage* zu erkennen. Bei der rot umrandeten Komponente *Descriptions* (s. Abb. 10) handelt es sich um eine weitere noch nicht veröffentlichte Komponente. Ebenso ist es aber möglich, in einer sich im privaten Bereich befindlichen Komponente (d.h. *DocumentationLanguage*) eine Komponente (d.h. *cmdi-language*) einzubinden, die bereits im Public Space zugänglich ist. Die Zugehörigkeit dieser Komponente zum öffentlichen Bereich der Component Registry wird durch die blaue Umrandung (s. Abb. 10) gekennzeichnet.

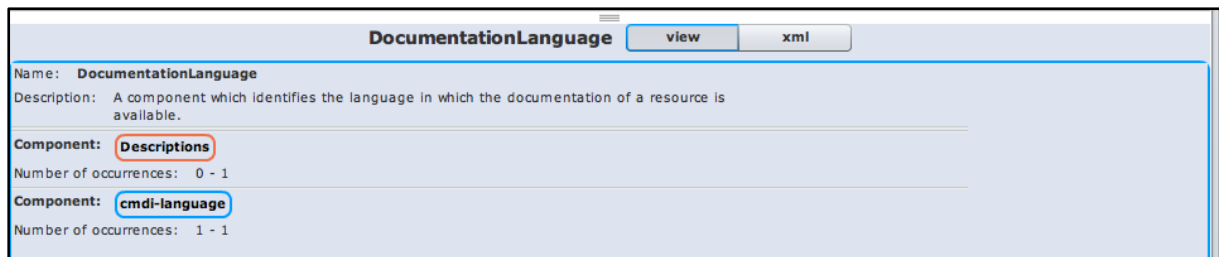


Abb. 10: Detailansicht einer Komponente im Work Space

Die weiteren Einzelheiten zu den dargestellten Informationen innerhalb der Komponenten werden im folgenden Inhaltspunkt 7 genauer erläutert.

### Ansicht im XML-Modus

Neben der Ansicht im View-Modus existiert für Profile ebenfalls eine Ansicht in der XML-Struktur, die durch den Tab *xml* ausgewählt werden kann (s. Inhaltspunkt 3). Grundsätzlich ist die XML-Ansicht für Profile und Komponenten gleich aufgebaut, da es sich bei Profilen bezüglich ihrer XML-Struktur allgemein ausgedrückt um Komponenten handelt, die als Containerelemente für weitere Komponenten fungieren. Sowohl bei Komponenten als auch bei Profilen wird beispielsweise im Header im Element *ID* die PID der Komponente (s. Abb. 5) bzw. des Profils (s. Abb. 11) angegeben. Es gibt jedoch einen wesentlichen Unterschied bezüglich der Spezifikation: jedes Profil und jede Komponente in CMDI besitzt das Wurzelement *CMD\_ComponentSpec* mit dem Attribut *isProfile* (s. Abb. 11). Wenn es sich bei der Komponente um ein Profil handelt, erhält das Attribut *isProfile* den Wert "true" (s. Abb. 11). Liegt hingegen eine Komponente vor, wird der Wert des Attributs mit "false" gleichgesetzt (s. Abb. 5).

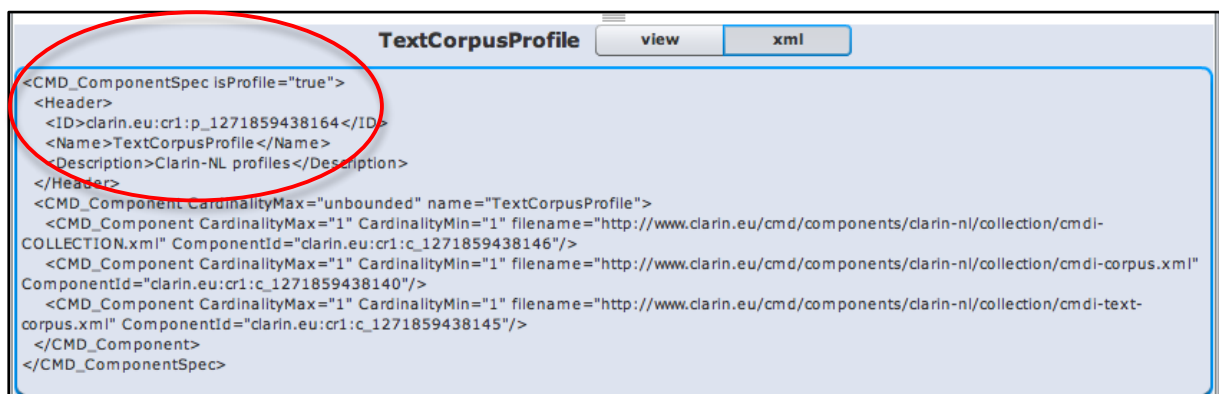


Abb. 11: Detailansicht eines Profils im XML-Modus

### Download

Eine weitere Funktionalität der Component Registry betrifft die Möglichkeit, Profile als XML- oder Schema-Dateien herunterzuladen. Dies ist sowohl für die lokale Erstellung von Profilen als auch für die Validierung erstellter Metadateninstanzen in CMDI gegen das jeweilige Schema eines Profils notwendig. Um zu diesen Downloadoptionen zu gelangen, kann der Nutzer das entsprechende Profil in der Übersichtstabelle auswählen

und bekommt per Rechtsklick auf den Namen des gewählten Profils ein Fenster mit weiteren Bearbeitungsoptionen angezeigt (s. Abb. 12a). Die Option *Download as XML...* ermöglicht das Speichern eines Profils als XML-Datei, die Option *Download as XSD...* hingegen das Sichern der Schema-Datei eines Profils. Außerdem werden zusätzliche Optionen zum Editieren (*Edit Item...*) oder Löschen (*Delete Item...*) eines Profils dargeboten (Details s. Inhaltspunkt 8).

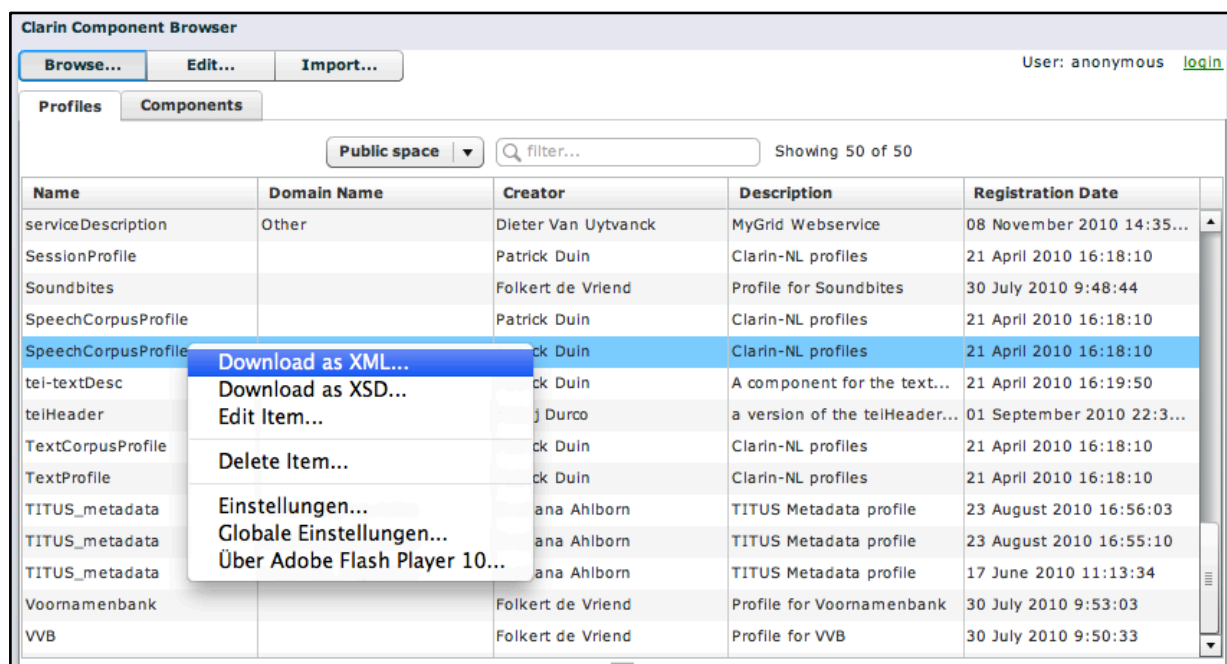


Abb. 12a: Bearbeitungsoptionen eines Profils I

### Profile im Work Space öffentlich zur Verfügung stellen

Obwohl Profile/Komponenten im privaten Work Space nur vom registrierten Nutzer heruntergeladen werden können, besteht in der Component Registry dennoch eine Möglichkeit, um Profile – genauer gesagt die Schema-Dateien von Profilen – öffentlich zugänglich machen zu können. Dazu wählt der eingeloggte Benutzer im Work Space den Tab *Profiles* aus und markiert ein gewünschtes Profil durch Anklicken mit der linken Maustaste (s. Abb. 12b). Durch einen Rechtsklick auf dieses ausgewählte Profil werden die Bearbeitungsoptionen sichtbar, aus denen man die Option *Show info* auswählt (s. Abb. 12b). Daraufhin erscheint ein weiteres Fenster, das neben der Bezeichnung *Link to xsd* den Link zum Schema des ausgewählten Profils enthält (s. Abb. 12c). Hier kann man entweder direkt durch Anklicken des Links die Schema-Datei herunterladen oder durch Anklicken des Buttons *copy to clipboard* (s. Abb. 12c) die URL in der Zwischenablage speichern, um sie auch an Personen weitergeben zu können, die keinen Zugang zum privaten Work Space haben. Neben dem Link zur Schema-Datei kann man außerdem durch Anklicken des Links neben der Bezeichnung *Bookmark link* (s. Abb. 12c) diese

URL direkt als Lesezeichen verwenden und ebenso in der Zwischenablage speichern. Diese Option steht im Work Space jedoch nur eingeloggten Nutzern zur Verfügung.

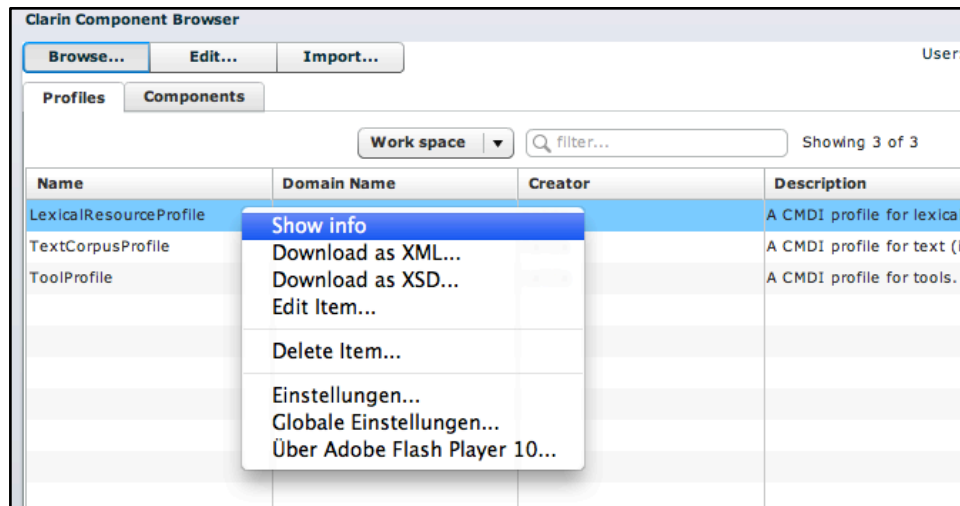


Abb. 12b: Bearbeitungsoptionen eines Profils II

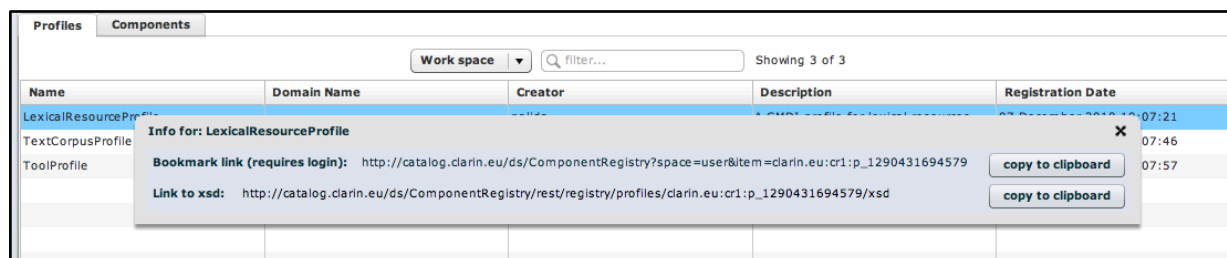


Abb. 12c: Bearbeitungsoptionen eines Profils III

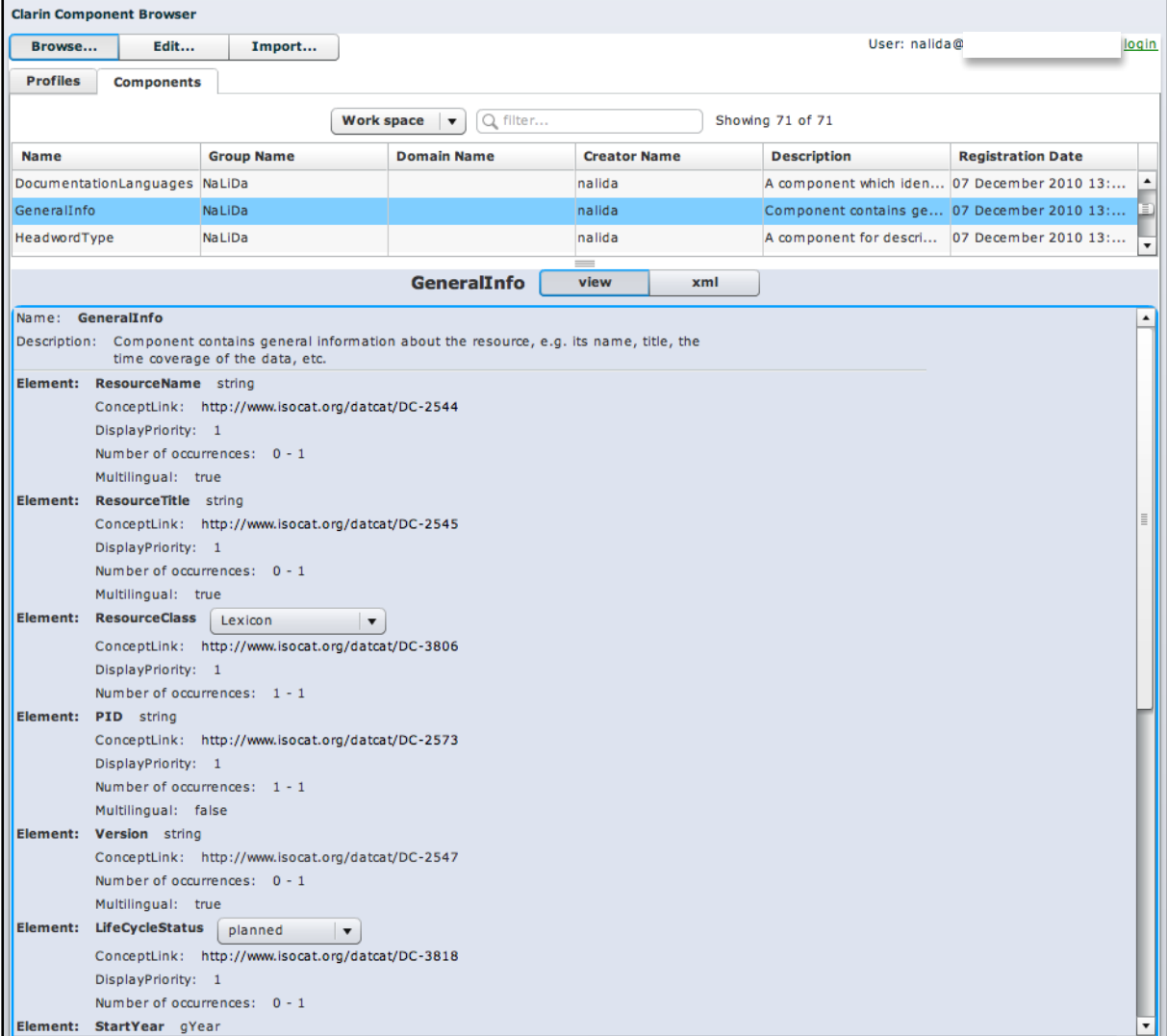
## 7. Komponenten

### Auswahl

Analog zu der Darstellung von Profilen kann die Übersicht über Komponenten in der Component Registry im Browse-Modus über den Tab *Components* ausgewählt werden. Sobald eine Komponente durch Anklicken ausgesucht wurde, wie beispielsweise die Komponente *GeneralInfo* (s. Abb. 13), erscheint unterhalb der tabellarischen Übersicht eine Detailansicht dieser Komponente (s. Abb. 13).

Der Ausschnitt der Detailansicht für die Komponente *GeneralInfo* in Abb. 13 zeigt nur die enthaltenen Datenkategorien (d.h. Elemente) dieser Komponente, die jedoch auch noch weitere, in der Abbildung nicht sichtbare Komponenten einbindet. Die Darstellung der Komponenten erfolgt nach dem bereits in Inhaltspunkt 6 erläuterten Prinzip (d.h. die Komponentennamen sind je nach Veröffentlichungsstatus farbig umrandet und können durch Anklicken expandiert werden) und wird daher an dieser Stelle nicht erneut aufgegriffen. Ebenso beziehen sich die Erklärungen auf die Ansicht im View-

Modus. Die bereits eingeführte XML-Ansicht (s. Abb. 5), die sich vor allem durch die zusätzliche Angabe der PID auszeichnet, wird im Falle von abweichenden Benennungen oder Darstellungsoptionen im jeweiligen Zusammenhang erörtert.



The screenshot shows the 'Clarin Component Browser' interface. At the top, there are buttons for 'Browse...', 'Edit...', and 'Import...'. The user is logged in as 'nalida@'. Below this, there are tabs for 'Profiles' and 'Components'. A 'Work space' dropdown and a search filter are present. A table lists components with columns for Name, Group Name, Domain Name, Creator Name, Description, and Registration Date. The 'GeneralInfo' component is selected and highlighted. Below the table, there are tabs for 'GeneralInfo', 'view', and 'xml'. The 'view' tab is active, showing detailed information for the 'GeneralInfo' component, including its description and several elements with their respective data types, concept links, display priorities, and occurrence counts.

Name	Group Name	Domain Name	Creator Name	Description	Registration Date
DocumentationLanguages	NaLiDa		nalida	A component which iden...	07 December 2010 13:...
<b>GeneralInfo</b>	NaLiDa		nalida	Component contains ge...	07 December 2010 13:...
HeadwordType	NaLiDa		nalida	A component for descri...	07 December 2010 13:...

**GeneralInfo** view xml

Name: **GeneralInfo**  
 Description: Component contains general information about the resource, e.g. its name, title, the time coverage of the data, etc.

**Element: ResourceName** string  
 ConceptLink: <http://www.isocat.org/datcat/DC-2544>  
 DisplayPriority: 1  
 Number of occurrences: 0 - 1  
 Multilingual: true

**Element: ResourceTitle** string  
 ConceptLink: <http://www.isocat.org/datcat/DC-2545>  
 DisplayPriority: 1  
 Number of occurrences: 0 - 1  
 Multilingual: true

**Element: ResourceClass**   
 ConceptLink: <http://www.isocat.org/datcat/DC-3806>  
 DisplayPriority: 1  
 Number of occurrences: 1 - 1

**Element: PID** string  
 ConceptLink: <http://www.isocat.org/datcat/DC-2573>  
 DisplayPriority: 1  
 Number of occurrences: 1 - 1  
 Multilingual: false

**Element: Version** string  
 ConceptLink: <http://www.isocat.org/datcat/DC-2547>  
 Number of occurrences: 0 - 1  
 Multilingual: true

**Element: LifecycleStatus**   
 ConceptLink: <http://www.isocat.org/datcat/DC-3818>  
 DisplayPriority: 1  
 Number of occurrences: 0 - 1

**Element: StartYear** gYear

Abb. 13: Ausschnitt einer Detailansicht im View-Modus für eine Komponente im Work Space

## Datenkategorien

Die Detailansicht einer Komponente im View-Modus stellt dem Nutzer verschiedene Informationen zum Inhalt einer Komponente dar, zu denen neben weiteren eingebundenen Komponenten die Datenkategorien zählen.

### Name

Für Datenkategorien wird zuerst der Name der jeweiligen Datenkategorie angegeben (z.B. *ResourceName*, s. Abb. 13). Die Benennung einer Datenkategorie spielt in CMDI nur



eine untergeordnete Rolle. Entscheidend ist die Angabe des Links zu ISOcat (s. im folgenden Unterpunkt Concept Link), welcher die PID (Persistent Identifier) der Datenkategorie enthält. So können Elemente trotz unterschiedlicher Benennungen in CMDI auf dasselbe Konzept verweisen. Dennoch sei es zur Einhaltung der Konsistenz empfohlen, bei der Verwendung bereits existierender Komponenten in der Component Registry oder definierter Datenkategorien in ISOcat die bestehenden Benennungen der Datenkategorien auch für eigene Verwendungszwecke zu übernehmen.

### Datentyp

Zusammen mit dem Namen wird der zugehörige Datentyp aufgeführt (d.h. im Fall von *ResourceName* ist der Inhaltstyp des Elements ein beliebiger String). Neben Strings können auch andere Datentypen für Elemente, wie z.B. *boolean*, *date*, *gYear* oder *anyURI*, gewählt werden (für weitere Datentypen und Erläuterungen s. <http://www.w3.org/TR/xmlschema-2/>, Punkt 3.2: *Primitive Datatypes*).

### Kontrolliertes Vokabular

Wenn bei einer Datenkategorie explizit kein zugehöriger Datentyp angegeben ist, liegt eine Einschränkung vor: es darf nur kontrolliertes Vokabular für den Inhalt der Datenkategorie verwendet werden. Dies bedeutet, dass dem Nutzer eine Auswahl an gültigen Begriffen präsentiert wird, aus denen der für die jeweilige Datenkategorie einer Metadateninstanz zutreffende Wert ausgewählt werden kann.

Dazu wird im View-Modus der Component Registry eine interaktive Darstellung in Form einer sogenannten Picklist dargeboten. Beispiele für kontrolliertes Vokabular sind in Abb. 13 bei den Datenkategorien *ResourceClass* und *LifeCycleStatus* zu sehen. Durch Anklicken des nach unten gerichteten Pfeils der Picklist (s. Abb. 14, unten) wird diese expandiert und der Nutzer kann durch Scrollen die zulässigen Werte einsehen und auswählen (s. Abb. 14, oben). So können z.B. für die Datenkategorie *ResourceClass* u.a. die Werte *Lexicon*, *Corpus*, *Tool*, *Experiment* oder *Grammar* ausgesucht werden. Beim Erstellen einer Metadateninstanz in CMDI ist dabei zu beachten, dass für Datenkategorien, die über Listen von kontrolliertem Vokabular verfügen, nur die darin definierten Begriffe zulässig sind. Wenn ein nicht in der Liste enthaltener Begriff in der Metadateninstanz verwendet wird, ist diese Instanz nicht mehr gegen das zugehörige Schema valide.

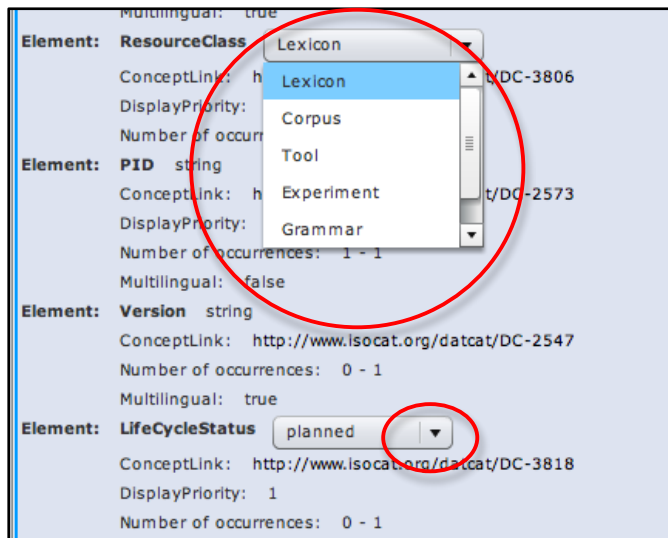


Abb. 14: Picklist für kontrolliertes Vokabular als Elementinhalte

In der XML-Struktur (s. Abb. 15) für kontrolliertes Vokabular wird die bereits im View-Modus illustrierte Ansicht (s. Abb. 14) folgendermaßen dargestellt:

```
<CMD_Element DisplayPriority="1" CardinalityMax="1" CardinalityMin="1" ConceptLink="http://www.isocat.org/datcat/DC-3806" name="ResourceClass">
  <ValueScheme>
    <enumeration>
      <item AppInfo="" ConceptLink="">Lexicon</item>
      <item AppInfo="" ConceptLink="">Corpus</item>
      <item AppInfo="" ConceptLink="">Tool</item>
      <item AppInfo="" ConceptLink="">Experiment</item>
      <item AppInfo="" ConceptLink="">Grammar</item>
      <item AppInfo="" ConceptLink="">Fieldwork Material</item>
    </enumeration>
  </ValueScheme>
</CMD_Element>
```

Abb. 15: Kontrolliertes Vokabular in der XML-Ansicht

Das in Abb. 15 aufgelistete kontrollierte Vokabular für den Inhalt des Elementes *ResourceClass* enthält das Containerelement `<ValueScheme>...</ValueScheme>`, das in den mit *item* betitelten Elementen die möglichen Elementinhalte von *ResourceClass* auflistet. Positioniert wird dieses Containerelement innerhalb der entsprechenden Datenkategorie, die in der XML-Struktur mit `<CMD_Element ...>...</CMD_Element>` beginnt.

### Attribute

Anzumerken sei an dieser Stelle außerdem, dass im Component Browser nicht nur für kontrolliertes Vokabular von Elementen Gebrauch von Picklisten gemacht wird, sondern auch für Attributwerte von Komponenten oder Elementen. Ein Beispiel hierfür zeigt die in Abb. 16 dargestellte Komponente *Prerequisite*, die über das Attribut *type* verfügt.



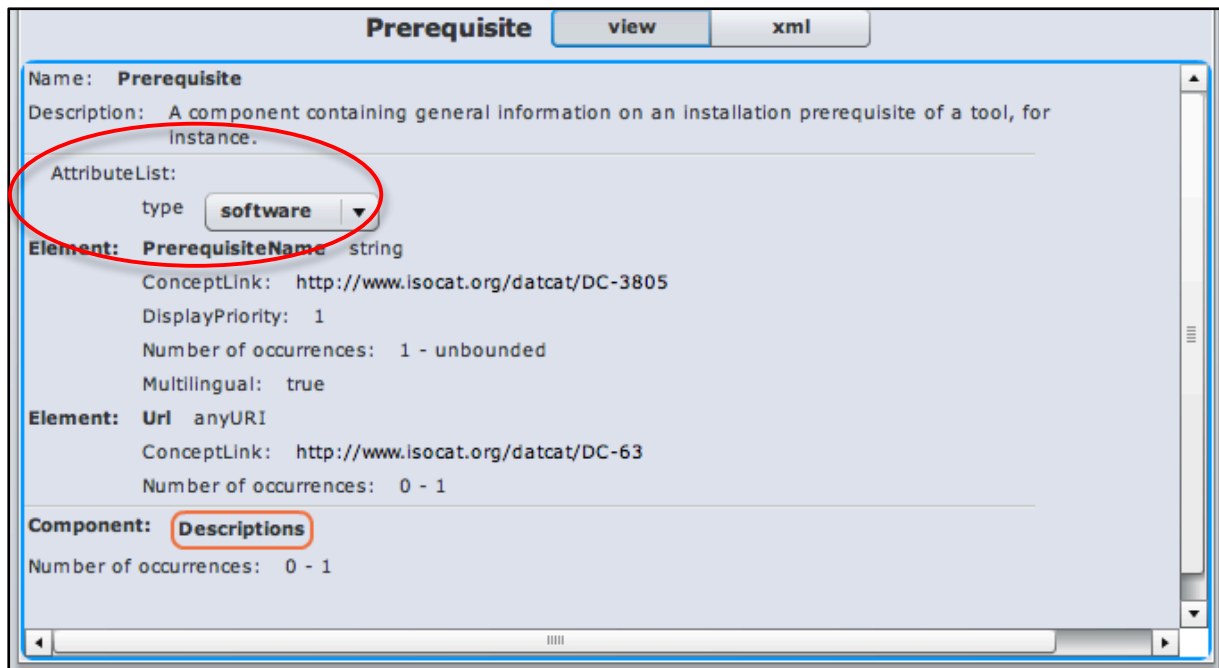


Abb. 16: Picklist für Attributwerte

Nach dem Namen und der Beschreibung der Komponente ist in der Ansicht in Abb. 16 die *Attribute List* angegeben, die den Namen des Attributes (d.h. *type*) und eine Picklist für die Attributwerte umfasst (z.B. kann als Wert *software* ausgewählt werden). In der XML-Ansicht wird eine solche Liste von vorgegebenen Attributwerten folgendermaßen dargestellt:

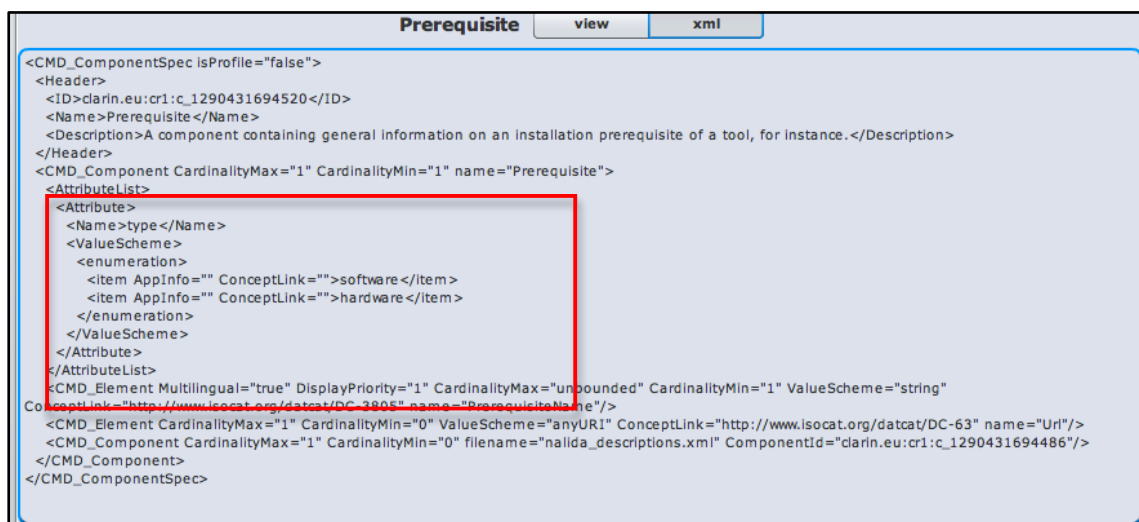


Abb. 17: Vordefinierte Attributwerte in der XML-Ansicht

Die in Abbildung 17 illustrierte XML-Struktur für Attributwertlisten beginnt mit dem Element `<AttributeList>...</AttributeList>`, welches als Container fungiert und beliebig viele Attribut-Elemente (`<Attribute></Attribute>`) beinhalten kann. Letztere dienen

ebenfalls als Containerelement für die XML-Elemente `<Name>...</Name>`, welches den Namen des Attributes festlegt (s. *type* in Abb. 16), und `<ValueScheme>...</ValueScheme>`, das eine Aufzählung der zugelassenen Attributwerte enthält (s. *software* und *hardware* in Abb. 17).

### Concept Link

Zurückkehrend zur Ausgangsbeschreibung der Datenkategorien innerhalb einer Komponente ist es notwendig, neben den bereits thematisierten Aspekten wie dem Elementnamen und Datentyp sowie dem kontrolliertem Vokabular für Elementinhalte und Attributwerte weitere Angaben im Component Browser zu erläutern. Dazu zählt beispielsweise die Angabe des *ConceptLink* für Elemente (s. Abb. 18).

Diese im View-Modus gewählten Angaben (s. Abb. 18, z.B. Elementname, Datentyp, ConceptLink, etc.) werden in der XML-Struktur einer Komponente als Attributwertpaare dargestellt. Beispielsweise handelt es sich bei *ConceptLink* um den Namen des Attributes und beim angegebenen Link um den zugehörigen Wert. Dieser Link verweist auf die Dokumentation der entsprechenden Datenkategorie, die durch die interaktive Beschaffenheit des View-Modus durch ein Anklicken des dokumentierten Links in einem neuen Fenster geöffnet werden kann. Die Dokumentation ist extern in ISOcat gespeichert (Data Category Registry for ISO TC 37, s. <http://www.isocat.org>), dem Verzeichnis für Datenkategorien.

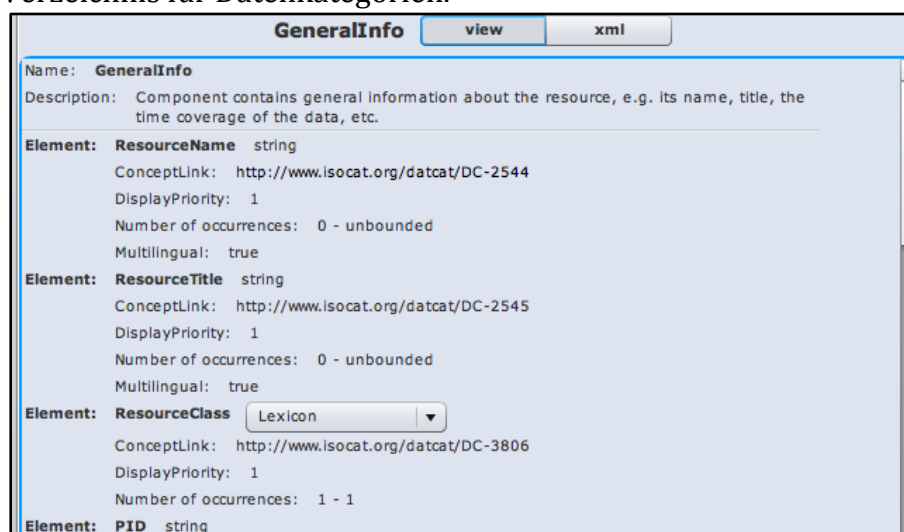


Abb. 18: Ausschnitt der Angaben zu Datenkategorien im View-Modus

In diesem Verzeichnis besitzt jede Datenkategorie einen Persistent Identifier, der auch in der Component Registry in den angegebenen Links der Datenkategorien auf ISOcat verzeichnet ist. Der Vorteil dieses Prinzips ist, dass die Benennung der Datenkategorien in CMDI nur eine nebensächliche Rolle spielt. Relevant ist lediglich die Angabe des Links zu ISOcat, der die PID der Datenkategorie enthält. Somit können auch unterschiedliche Benennungen von Elementen in verschiedenen CMDI-Metadateninstanzen auf dasselbe Konzept in ISOcat verweisen.



### Display Priority

Für Datenkategorien kann in der Component Registry außerdem ihre Anzeigepriorität (s. *Display Priority*, Abb. 18) bei der Verarbeitung mit externen Tools angegeben werden. Wenn ein Element zuerst angezeigt werden soll, setzt man den Wert von Display Priority gleich 1, wodurch das Element in der Anzeige höchste Priorität erhält. Soll eine Datenkategorie hingegen in der Anzeige nicht automatisch erscheinen, setzt man den Wert gleich 0. Gute Kandidaten für eine hohe Anzeigepriorität sind beispielsweise der Name, Titel oder die Klasse einer Ressource (s. Abb. 18).

### Number of Occurrences

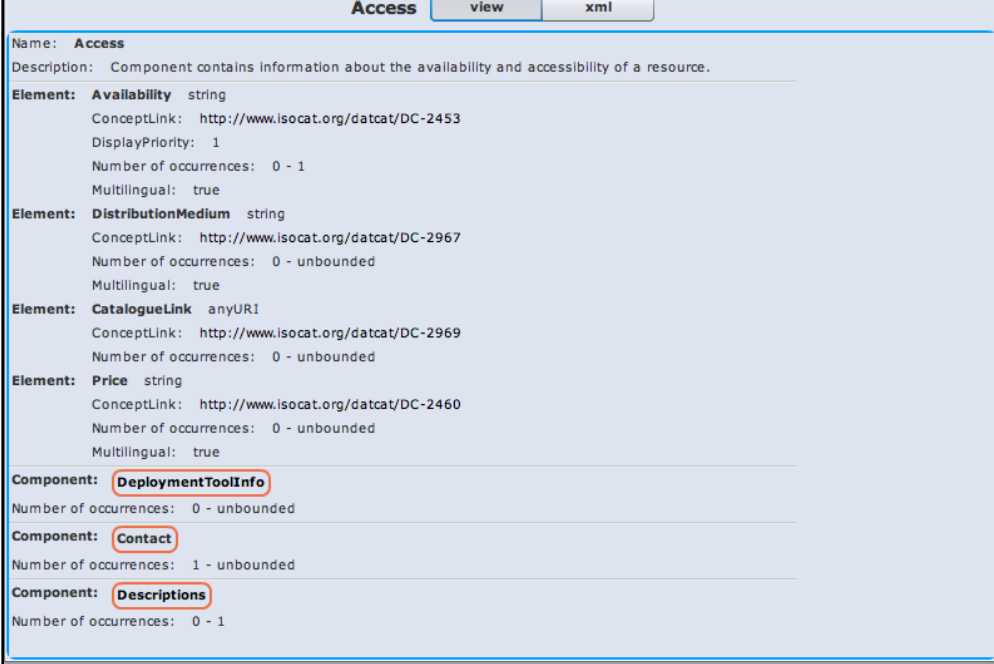
Des Weiteren wird für jede Datenkategorie mit *Number of Occurrences* die zugelassene Anzahl ihrer Verwendung in einer Metadateninstanz definiert. Die erste Zahl gibt dabei immer das minimale und die letzte Zahl das maximale Auftreten einer Datenkategorie an. Ein Element ist optional, wenn das minimale Auftreten mit Null gleichgesetzt ist (s. z.B. Element *ResourceName*, Abb. 18). Wenn die Kardinalität 1-1 gewählt wurde, darf ein Element nur genau einmal gebraucht werden. Um eine unendliche Verwendung erzielen zu können, muss die maximale Kardinalität mit *unbounded* gleichgesetzt werden. In der XML-Ansicht einer Komponente oder eines Profils werden für diese Definitionen der Kardinalität die Attribute *CardinalityMax* und *CardinalityMin* verwendet (s. Abb. 17).

### Multilingual (xml:lang)

Die letzte Anzeigeeinformation für Datenkategorien im Component Browser betrifft die Angabe zur Multilingualität. Ausgenommen von Elementen, deren Inhalte über kontrolliertes Vokabular verfügen, wird dieses Attribut für jedes Element zur Identifikation der Mehrsprachigkeit angegeben. Diese Information dient der Sprachidentifikation in den Metadateninstanzen, in denen durch die Verwendung des Attributs *xml:lang* die Sprache eines Elementinhaltes spezifiziert werden kann. Wenn *Multilingual* den Wert *true* besitzt (s. Abb. 18), kann das Attribut *xml:lang* in der Metadateninstanz verwendet werden. Somit kann beispielsweise der Titel einer Ressource in verschiedenen Sprachen dokumentiert werden, da das Element durch die zugelassene Mehrsprachigkeit beliebig oft verwendet werden und bei jeder Verwendung des Elements das Attribut *xml:lang* einen anderen Wert für die jeweilige vorliegende Sprache erhalten kann. In der Component Registry wird nach momentanem Entwicklungsstand beim Vorliegen einer erlaubten Multilingualität die maximale Kardinalität eines Elementes automatisch als *unbounded* definiert, sofern nicht zuvor ein maximal einmaliges Auftreten festgelegt war. Jedoch ist die Verwendung des Attributes *xml:lang* in einer Metadateninstanz nicht zulässig, wenn *Multilingual* mit dem Wert *false* gleichgesetzt ist.

### Komponenten innerhalb von Komponenten

Die bisher erläuterten Details zur Darstellung der Datenkategorien treffen zum Teil ebenso auf die Illustration von Komponenten zu, die selbst weitere Komponenten einbinden können (s. Abb. 19).



**Name:** Access

**Description:** Component contains information about the availability and accessibility of a resource.

**Element: Availability** string  
 ConceptLink: <http://www.isocat.org/datcat/DC-2453>  
 DisplayPriority: 1  
 Number of occurrences: 0 - 1  
 Multilingual: true

**Element: DistributionMedium** string  
 ConceptLink: <http://www.isocat.org/datcat/DC-2967>  
 Number of occurrences: 0 - unbounded  
 Multilingual: true

**Element: CatalogueLink** anyURI  
 ConceptLink: <http://www.isocat.org/datcat/DC-2969>  
 Number of occurrences: 0 - unbounded

**Element: Price** string  
 ConceptLink: <http://www.isocat.org/datcat/DC-2460>  
 Number of occurrences: 0 - unbounded  
 Multilingual: true

**Component: DeploymentToolInfo**  
 Number of occurrences: 0 - unbounded

**Component: Contact**  
 Number of occurrences: 1 - unbounded

**Component: Descriptions**  
 Number of occurrences: 0 - 1

Abb. 19: Darstellung von eingebundenen Komponenten im View-Modus

Für die innerhalb der Komponente *Access* (s. Abb. 19) eingebundenen Komponenten *DeploymentToolInfo*, *Contact* und *Descriptions* wird jeweils der genannte Name sowie die Kardinalität der Komponente angegeben. Des Weiteren besteht die Möglichkeit, Komponenten durch Anklicken des Namens zu expandieren (s. Inhaltspunkt 6), um weitere Informationen über ihre Inhalte zu erhalten.

## Download

Eine weitere Funktionalität der Component Registry betrifft die Möglichkeit, Komponenten als XML-Dateien herunterzuladen. Dies ist beispielsweise für die lokale Erstellung von Komponenten von Nutzen. Um zu diesen Downloadoptionen zu gelangen, kann der Nutzer die entsprechende Komponente in der Übersichtstabelle auswählen und bekommt per Rechtsklick auf den Namen der gewählten Komponente ein Fenster mit weiteren Bearbeitungsoptionen angezeigt (s. Abb. 20). Durch die Auswahl der dargebotenen Option *Download als XML...* kann die Komponente im XML-Format gespeichert werden. Die weiteren Bearbeitungsfunktionalitäten werden im folgenden Inhaltspunkt genauer erläutert.

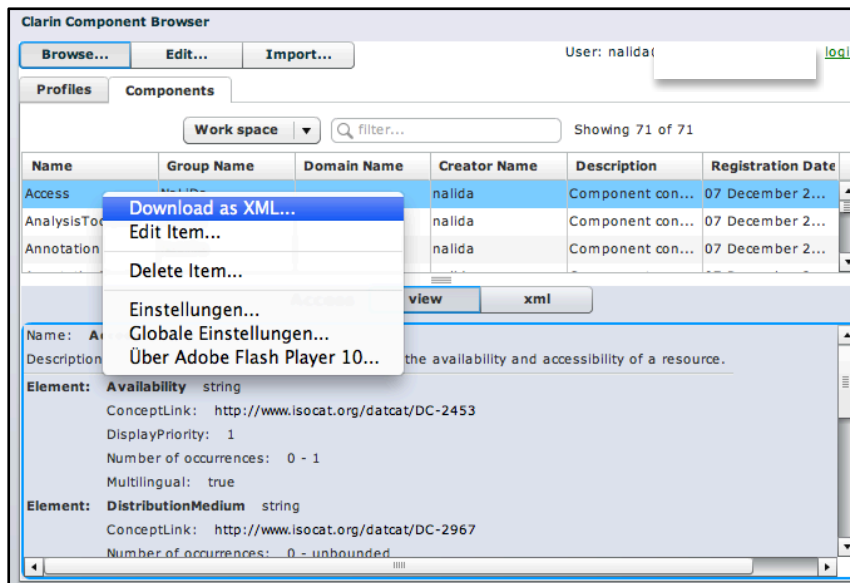


Abb. 20: Bearbeitungsoptionen einer Komponente

## 8. Funktionalitäten

Die Bearbeitungsoptionen innerhalb der Component Registry umfassen drei Kernfunktionalitäten, die in den folgenden Unterpunkten detailliert beschrieben werden: *Browse*, *Edit* und *Import*. In der Browseransicht sind diese Funktionalitäten in drei unterschiedlichen Tabs angeordnet, sodass der Nutzer durch Anklicken des entsprechenden Tabs zwischen den Optionen wechseln kann. Als Defaultansicht fungiert dabei die Browse-Funktionalität.

### a. Browse: Einsicht und Download von Komponenten/Profilen

Die Browse-Funktion gestattet dem Nutzer die Einsicht und den Download der bereits registrierten Komponenten und Profile. Für diese Einsicht ist im öffentlich zugänglichen Bereich kein Login erforderlich. Lediglich für eigene Komponenten und Profile, die sich noch im privaten Work Space befinden, kann ein Nutzer diese auch im Browse-Modus nur nach einem erfolgten Login einblicken. Die dem Nutzer zur Verfügung stehenden Optionen umfassen in der Browse-Funktionalität u.a. die Übersicht über Profile und Komponenten mit jeweiliger Detailansicht im View- oder XML-Modus (s. Inhaltspunkt 3), die Wahl zwischen Public Space (s. Inhaltspunkt 4) und Work Space (s. Inhaltspunkt 5) und eine Filteroption für die Suchfunktionalität (s. Inhaltspunkt 3).

### b. Edit: Erstellen, Bearbeiten und Löschen von Komponenten/Profilen

Der Edit-Modus ermöglicht im registrierten Zugang das Erstellen neuer und das Editieren bestehender Komponenten und Profile in der Component Registry. Dazu existieren zwei unterschiedliche Funktionalitäten: eine Funktion verwendet den Edit-Tab in der Browseransicht und die andere Funktion die Option *Edit Item*, wobei beide den Nutzer zu einem Webformular zur weiteren Bearbeitung führen. Die Verwendung dieser beiden Funktionalitäten wird im Folgenden am Beispiel von Komponenten näher

erläutert, ist aber gleichermaßen auf die Bearbeitung von Profilen übertragbar. Anzumerken sei außerdem, dass neben diesen Editierungsmodi auch die Möglichkeit besteht, als XML-Dateien gespeicherte lokale Komponenten und Profile mittels einer Importfunktion in die Component Registry hochzuladen (s. Inhaltspunkt 8c).

### Edit Item: Bearbeiten vom Komponenten und Profilen

Die Editierfunktion *Edit Item* wird mithilfe eines Rechtsklicks mit der Maus auf die jeweilige zu bearbeitende Komponente (s. Abb. 21, *Komponente GeneralInfo*) bzw. das jeweilige zu bearbeitende Profil ausgewählt.

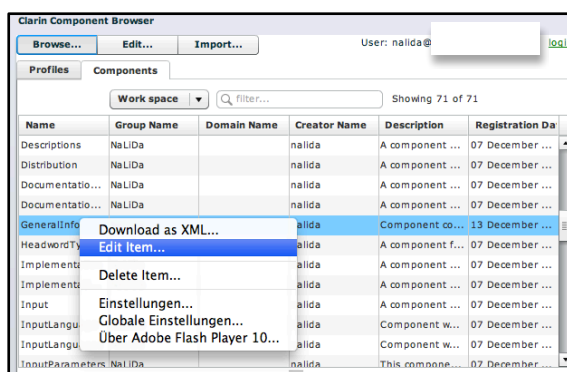


Abb. 21: Editierfunktion einer Komponente

Diese Funktion bietet die einzige Möglichkeit, bereits bestehende Komponenten und Profile in der Component Registry zu bearbeiten. Eine einfache Auswahl einer Komponente/eines Profils durch Markieren mit der linken Maustaste und einem anschließenden Anklicken des Edit-Tabs in der Browseransicht würde jedoch nicht zum gleichen Resultat führen (s. folgender Unterpunkt Edit Item vs. Edit Tab).

Nachdem man also die Option *Edit Item* mittels Anklicken für die markierte Komponente *GeneralInfo* aktiviert hat (s. Abb. 21), gelangt man zur Ansicht im Edit-Modus der ausgewählten Komponente (s. Abb. 22). Hier besteht die Möglichkeit, mittels eines Webformulars Änderungen an den bereits vorhandenen Inhalten der Komponente vorzunehmen oder weitere Elemente oder Komponenten hinzuzufügen. Die für diese Zwecke zur Verfügung stehenden Bearbeitungsoptionen werden in der Ansicht des Edit-Tabs im Component Browser dargestellt (s. Abb. 22) und daher im Folgenden im Zusammenhang mit dieser Funktion gemeinsam thematisiert.

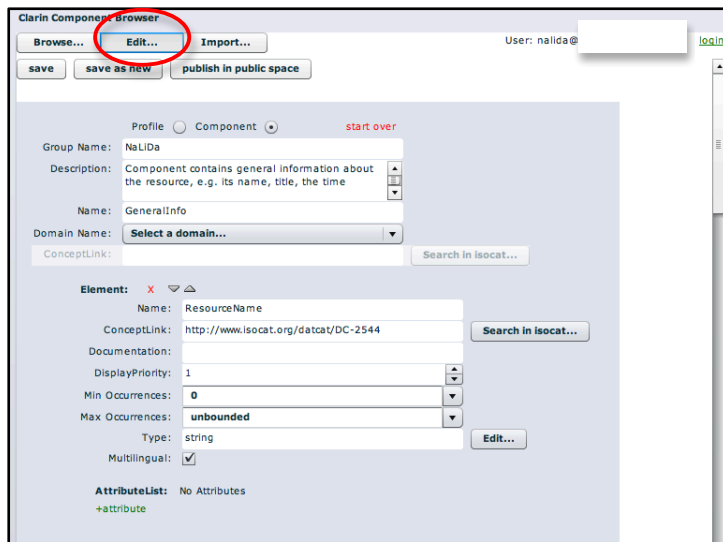


Abb. 22: Ausschnitt einer Komponente im Edit-Modus

### Edit Item vs. Edit-Tab

Daher ist eine Unterscheidung zwischen der Funktion *Edit Item* und dem Edit-Tab vorzunehmen. Der Edit-Tab wird primär zur Erstellung neuer Komponenten verwendet sowie zum Editieren der zuletzt bearbeiteten Komponente. Wenn man beispielsweise zuerst die Komponente *Descriptions* über die Funktion *Edit Item* bearbeitet hat und danach die Komponente *Input* auswählt und dieses Mal aber den Edit-Tab zur weiteren Bearbeitung anklickt, wird nicht die zweite ausgewählte Komponente *Input* im Edit-Modus angezeigt. Stattdessen wird die Komponente im Editiermodus geöffnet, die man zuletzt bearbeitet hat (d.h. im Beispiel die Komponente *Descriptions*). Grund für dieses Feature ist, dass man so bei der Entwicklung von Komponenten/Profilen vom Edit-Modus in den Browse-Modus wechseln kann ohne den Bearbeitungsprozess der aktuellen Komponente abubrechen (d.h. obwohl man die aktuell bearbeitete Komponente noch nicht abgespeichert hat, gehen die Informationen durch einen Wechsel in den Browse-Modus nicht verloren). Daher sollte man zum Bearbeiten einer existierenden Komponente stets die Funktion *Edit Item* auswählen und zum Erstellen neuer Komponenten den Edit-Tab.

### Start Over

Da im Bearbeitungsmodus im Edit-Tab die zuletzt bearbeiteten Profile/Komponenten zwischengespeichert werden, kann es notwendig sein, zur Erstellung einer neuen Komponente von einer weiteren Funktionalität Gebrauch zu machen: der Funktion *Start Over* (s. Abb. 22, rot hervorgehobene Schrift). Diese erzeugt ein leeres Webformular zur Erstellung einer neuen Komponente. Die Komponente wird dabei als neue Komponente abgespeichert und überschreibt keine andere, die sich vorher möglicherweise im Zwischenspeicher des Edit-Modus befunden hat.





### Editieren: Public Space vs. Work Space

Ebenso ist an dieser Stelle außerdem eine grundlegende Unterscheidung zwischen den Editiermöglichkeiten im Public Space und im Work Space zu treffen. Grundsätzlich ist ein Abändern einer Komponente oder eines Profils nur für selbst erstellte, noch nicht veröffentlichte Komponenten/Profile im Work Space möglich. Im Public Space wird eine Verwendung der Funktion *Edit Item* ebenfalls unterstützt, hat allerdings ein anderes Resultat zur Folge: da weder einmal veröffentlichte Profile/Komponenten abgeändert werden können noch eine Abänderung dieser Profile/Komponenten durch andere Nutzer möglich ist, bei denen es sich nicht um die Ersteller selbst handelt, wird nicht die ausgewählte Komponente bzw. das gewählte Profil editiert. Stattdessen wird eine Kopie dieser Komponente/dieses Profils erzeugt, die zwar editiert werden kann, aber im Work Space unter Berücksichtigung der erfolgten Änderungen und unter Vergabe einer neuen PID gespeichert wird. Somit bleibt die ursprüngliche Komponente/das ursprüngliche Profil im Public Space unverändert (d.h. es wird nichts überschrieben) und dient nur als Grundlage einer modifizierten neuen Komponente/Profils im Work Space.

### Löschen von Komponenten und Profilen

Des Weiteren bedarf es einer Erörterung einer weiteren grundlegenden Bearbeitungsfunktion im Entwicklungsprozess von Komponenten und Profilen in CMDI: dem Löschen existierender Komponenten oder Profile. Diese Funktion kann nur ausgeführt werden, wenn es sich bei dem Nutzer selbst um den Ersteller der zu löschenden Komponente oder des zu löschenden Profils handelt und wenn sich die Komponente/das Profil im privaten Work Space befindet. Wenn beide Voraussetzungen erfüllt sind, kann der Nutzer wie auch bei der Option *Edit Item* durch einen Rechtsklick auf eine markierte zu löschende Komponente/ein zu löschendes Profil die Funktion *Delete Item* auswählen (s. Abb. 21). Diese Funktion ist wie alle Bearbeitungsfunktionen für bestehende Komponenten und Profile nur über die mittels eines Rechtsklicks zugänglichen Optionen aktivierbar. Bevor die Operation durchgeführt wird, wird dem Nutzer ein Meldungsfenster präsentiert, bei dem er entweder erneut das Löschen der Komponente/des Profils bestätigen oder den Prozess abbrechen kann (s. Abb. 23). Ein einmal erfolgtes Löschen einer Komponente/eines Profils kann allerdings nicht mehr rückgängig gemacht werden.

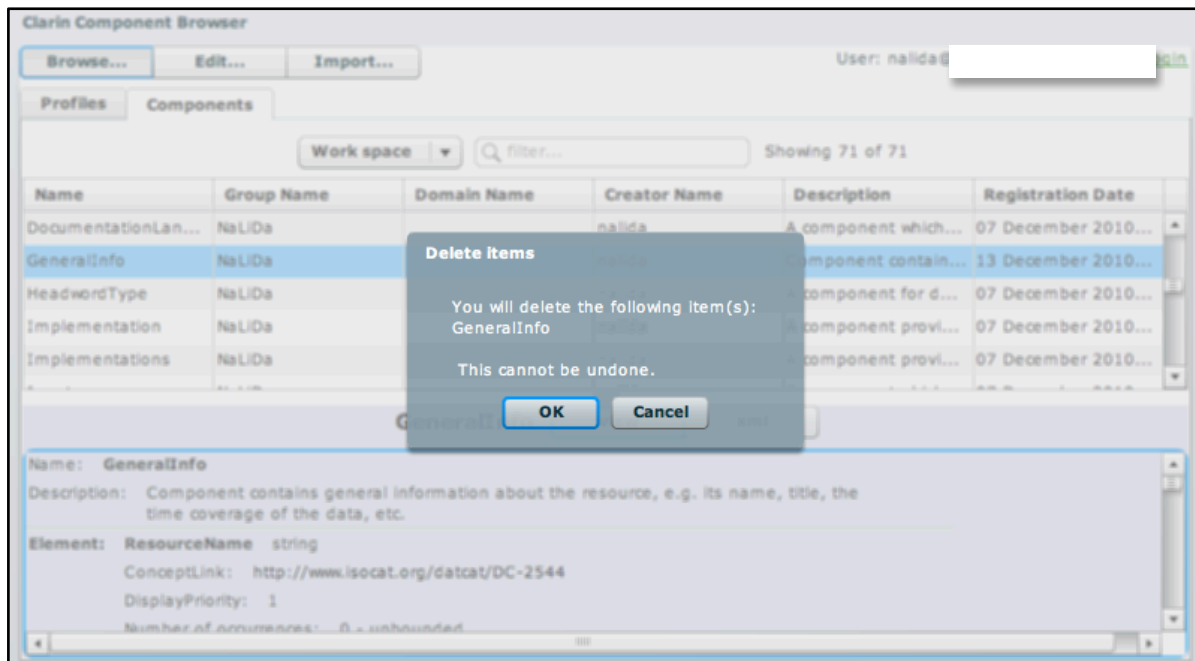


Abb. 23: Löschen einer Komponente

### Edit-Tab: Erstellen und Bearbeiten von Komponenten und Profilen

Die zweite Funktion, die sowohl zum Erstellen als auch zum Bearbeiten von Komponenten und Profilen dient, stellt das über den Edit-Tab im Component Browser zugängliche Webformular dar (s. Abb. 24). Das in Abb. 24 gezeigte leere Webformular dient der Erstellung neuer Komponenten und Profile. Bei einem Großteil der Funktionalitäten im Component Browser ist es möglich, durch ein Mouseover eine Definition der Funktion zu erhalten (d.h. der Nutzer kann mit der Maus über das Symbol oder den Text fahren und bekommt dabei ein Textfeld mit einer Funktionserläuterung präsentiert). Mittels eines Radio Buttons kann ausgewählt werden, ob ein neues Profil (s. Abb. 24, *Profile*) oder eine neue Komponente (s. Abb. 24, *Component*) erstellt werden soll. In den Textfeldern unterhalb werden allgemeine Informationen zum Profil/zur Komponente eingetragen, die in der tabellarischen Übersicht im Browse-Modus angezeigt werden. Zu diesen Informationen zählen beispielsweise der Name des Profils/der Komponente und eine dazugehörige Beschreibung der intendierten Verwendung.

**Clarín Component Browser**

Browse... **Edit...** Import... User: nalida@  [login](#)

save save as new publish in public space

Profile  Component  [start over](#)

Group Name:

Description:

Name:

Domain Name: **Select a domain...**

ConceptLink:  [Search in isocat...](#)

**Element:** X ▾ ▲

Name:

ConceptLink:  [Search in isocat...](#)

Documentation:

DisplayPriority: 0

Min Occurrences: 1

Max Occurrences: 1

Type: string [Edit...](#)

Multilingual:

**AttributeList:** No Attributes  
+attribute

+element  
+component

save save as new publish in public space

Drag existing components onto edit canvas

**Work space** ▾  Showing 71 of 71

Name	Group Name	Domain Name	Creator Name	Description	Registration Date
Access	NaLiDa		nalida	Component cont...	07 December 20...
AnalysisToolInfo	NaLiDa		nalida	Component cont...	07 December 20...
Annotation	NaLiDa		nalida	Component cont...	07 December 20...
AnnotationToolI...	NaLiDa		nalida	Component cont...	07 December 20...
AnnotationType	NaLiDa		nalida	A component for...	07 December 20...
AnnotationTypes	NaLiDa		nalida	A component for...	07 December 20...

Abb. 24: Browseransicht im Editiermodus

## Elemente

Nach den allgemeinen Angaben können dem Profil bzw. der Komponente Elemente hinzugefügt werden (s. Abb. 25).

Abb. 25: Ausschnitt des Editiermodus für Elemente

## Löschen und Festlegen der Reihenfolge

Neben der Bezeichnung *Element* sind in Abb. 25 drei Symbole zu erkennen: ein rotes Kreuz zum Löschen eines Elementes sowie ein nach oben und ein nach unten zeigender Pfeil zur Änderung der Reihenfolge der Elemente, sofern in einer Komponente bzw. einem Profil mehrere Elemente vorhanden sind.

## Name und Concept Link

Danach folgt ein Textfeld zur Eingabe des Namens der Datenkategorie sowie ein weiteres für die Angabe des sogenannten *Concept Link*. Dieser Link verweist auf die Definition der Datenkategorie in ISOcat (s. Inhaltspunkt 7). Damit ein Element in einer Komponente/einem Profil hinzugefügt werden kann, ist es also notwendig, dass es entweder bereits in ISOcat verzeichnet ist oder dort neu angelegt wird. Der Component Browser bietet an dieser Stelle die Möglichkeit, nach einem Anklicken des Buttons *Search in isocat...* ein neues Fenster innerhalb der Browseransicht als Pop-up zu öffnen, in dem der Nutzer im ISOcat-Verzeichnis nach bestehenden Datenkategorien suchen kann (s. Abb. 26).

Name	Definition	Identifier	Owner	Persistent Identifier	Type	Version
resource title	The title is th...	resourceTitle	Wittenburg, P...	http://www.isocat.or...	complex	1:0
project title	The full title o...	projectTitle	Wittenburg, P...	http://www.isocat.or...	complex	1:0
resource class	Defines the d...	resourceClass	Hoppermann,...	http://www.isocat.or...	complex	1:0
resource name	A short name ...	resourceName	Wittenburg, P...	http://www.isocat.or...	complex	1:0
input resource	Classifies the ...	inputResource	Hoppermann,...	http://www.isocat.or...	complex	1:0
creation tool	Indicates the ...	creationTool	Wittenburg, P...	http://www.isocat.or...	complex	1:0

Abb. 26: Suchfenster für ISOcat

Im erscheinenden Suchfenster kann man nach einem Konzept in ISOcat suchen, wie z.B. nach *resource title* für den Titel einer Ressource (s. Abb. 26). Unterhalb der Suchfunktion erhält man dann in einer tabellarischen Übersicht die Suchresultate für zutreffende Datenkategorien, aus denen man eine durch Markieren auswählen und durch ein Bestätigen mit dem Button *OK* den zugehörigen Concept Link in das Textfeld des Formulars einfügen kann (s. Abb. 27).

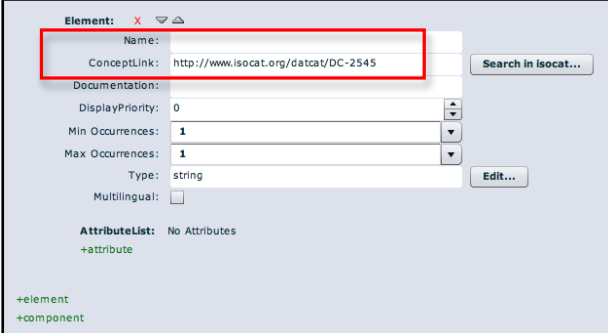


Abb. 27: Concept Link aus ISOcat einfügen

### Documentation

Des Weiteren ist es optional möglich, eine Dokumentation der Datenkategorie unter dem Textfeld *Documentation* anzugeben (s. Abb. 27). Eine solche Dokumentation kann beispielsweise aus der Definition der Datenkategorie in ISOcat entnommen werden.

### Display Priority

Für Datenkategorien kann in der Component Registry außerdem ihre Anzeigepriorität (s. *Display Priority*, Abb. 27) bei der Verarbeitung mit externen Tools angegeben werden (s. auch Inhaltspunkt 7). Wenn ein Element zuerst angezeigt werden soll, setzt man den Wert von *Display Priority* mithilfe der nach oben zeigenden Pfeiltaste oder durch Eingabe der Zahl über die Tastatur gleich 1. Dadurch erhält das Element in der Anzeige höchste Priorität. Soll eine Datenkategorie hingegen in der Anzeige nicht automatisch erscheinen, setzt man den Wert mittels der nach unten zeigenden Pfeiltaste oder einer manuellen Eingabe gleich 0. Wenn eine Komponente jedoch nur eine Datenkategorie enthält, muss die *Display Priority* auf 1 gesetzt werden. Gute Kandidaten für eine hohe Anzeigepriorität sind beispielsweise der Name, Titel oder die Klasse einer Ressource.

### Min Occurrences und Max Occurrences

Neben der Anzeigepriorität wird beim Erstellen einer Komponente bzw. eines Profils für jede Datenkategorie die zugelassene Anzahl ihrer Verwendung in einer Metadateninstanz definiert (s. auch Inhaltspunkt 7). Das Feld *Min Occurrences* im Editiermodus gibt dabei immer das minimale und *Max Occurrences* das maximale Auftreten einer Datenkategorie an. Beiden Feldern können entweder durch eine Eingabe über die Tastatur oder über eine sogenannte Drop-Down-Liste (s. Abb. 27, nach unten zeigender Pfeil neben dem jeweiligen Textfeld für *Min Occurrences* oder *Max Occurrences*) Zahlenwerte zugewiesen werden. Ein Element ist optional, wenn das

minimale Auftreten 0 entspricht. Wenn die Kardinalitäten bei *Min Occurrences* und *Max Occurrences* je gleich 1 gewählt werden, darf ein Element nur genau einmal gebraucht werden (s. Formulareinstellung in Abb. 27). Um eine unendliche Verwendung erzielen zu können, muss die maximale Kardinalität mit *unbounded* gleichgesetzt werden.

## Type

In einem nächsten Schritt wird der zulässige Datentyp eines Elementes unter der Bezeichnung *Type* (s. Abb. 27) festgelegt. Bei Datentypen handelt es sich um den Inhaltstyp eines Elementes, wie beispielsweise um Strings oder Typen wie *boolean*, *date*, *gYear* oder *anyURI* (s. Inhaltspunkt 7, s. <http://www.w3.org/TR/xmlschema-2/>, Punkt 3.2: *Primitive Datatypes*). Die Definition eines zulässigen Datentyps wird im Component Browser durch Anklicken des *Edit*-Buttons neben dem Textfeld für *Type* festgelegt (s. Abb. 28).

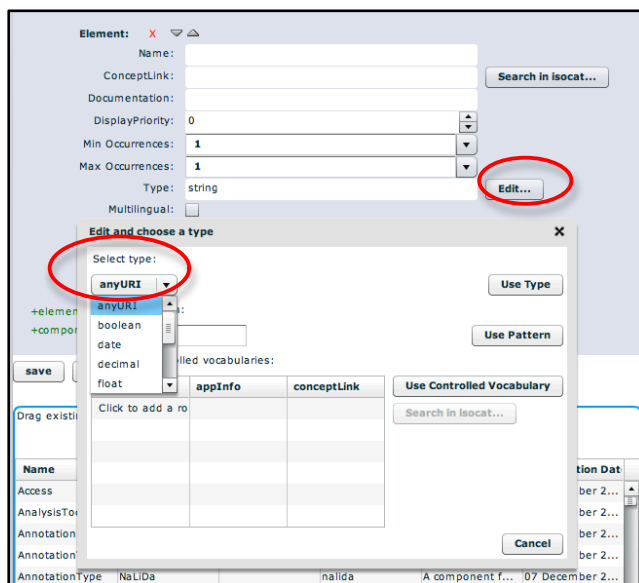


Abb. 28: Auswahl des Datentyps I

Nach Anklicken des *Edit*-Buttons öffnet sich ein Fenster als Pop-up (s. Abb. 28) mit drei unterschiedlichen Optionen zur Festlegung des Datentyps: die Definition von primitiven Datentypen, Pattern und kontrolliertem Vokabular.

### Primitive Datentypen

Die Auswahl primitiver Datentypen erfolgt über eine Drop-Down-Liste unterhalb der Option *Select type* (s. Abb. 28). Der Nutzer kann hier einen Typ durch Markieren definieren und die Wahl über den Button *Use Type* bestätigen. Dadurch wird der festgelegte Typ als Inhalt des Feldes *Type* im Editierformular im Component Browser eingefügt.

### Pattern

Zweitens ist es möglich, sogenannte Pattern (d.h. Muster wie z.B. reguläre Ausdrücke) zu spezifizieren, um nur bestimmte Werte zuzulassen (s. Abb. 29).

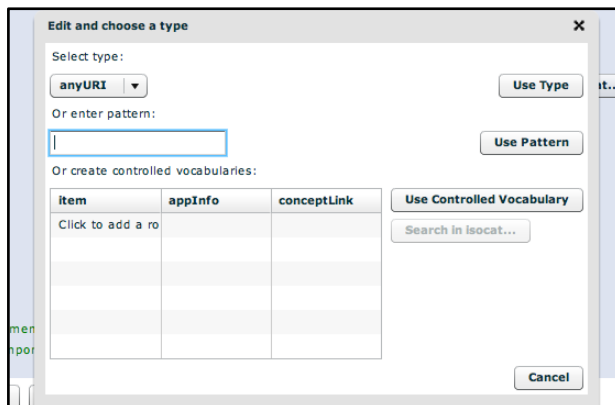


Abb. 29: Auswahl des Datentyps II

Der Nutzer kann in diesem Fall das gewünschte Muster im Textfeld unterhalb der Beschreibung *Or enter pattern* manuell eingeben und mit dem Button *Use Pattern* bestätigen.

#### Kontrolliertes Vokabular

Drittens kann als Elementinhalt auch nur ein geschlossenes Vokabular zugelassen werden. In diesem Fall wird bei der späteren Erstellung einer Metadateninstanz dem Nutzer eine Auswahl an gültigen Begriffen präsentiert, aus denen der für die jeweilige Datenkategorie zutreffende Wert ausgesucht werden kann (s. Abb. 14). Die Verwendung anderer Werte, die nicht innerhalb des kontrollierten Vokabulars aufgeführt sind, ist nicht möglich (d.h. die Instanz wäre sonst nicht mehr valide). Definiert wird dieses kontrollierte Vokabular in der tabellarischen Übersicht unter der Option *Or create controlled vocabularies* (s. Abb. 30). Ein Abspeichern ist über den Button *Use Controlled Vocabulary* möglich (s. Abb. 30).

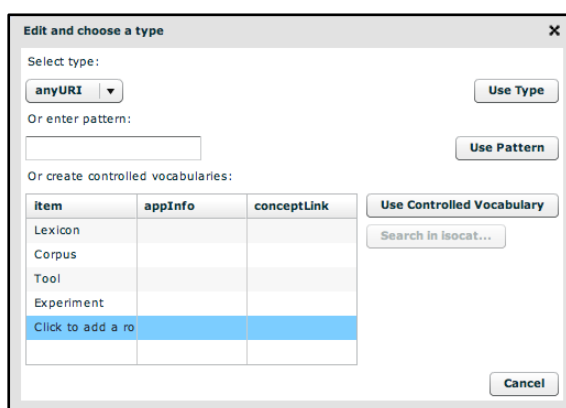


Abb. 30: Definition von kontrolliertem Vokabular

In der Spalte *Item* (s. Abb. 30) werden die gültigen Begriffe innerhalb des geschlossenen Vokabulars festgelegt. Dazu klickt man innerhalb der Spalte auf eine Zeile und gibt über die Tastatur den entsprechenden Begriff ein (z.B. *Lexicon*, *Corpus*, *Tool*, *Experiment* in Abb. 30).



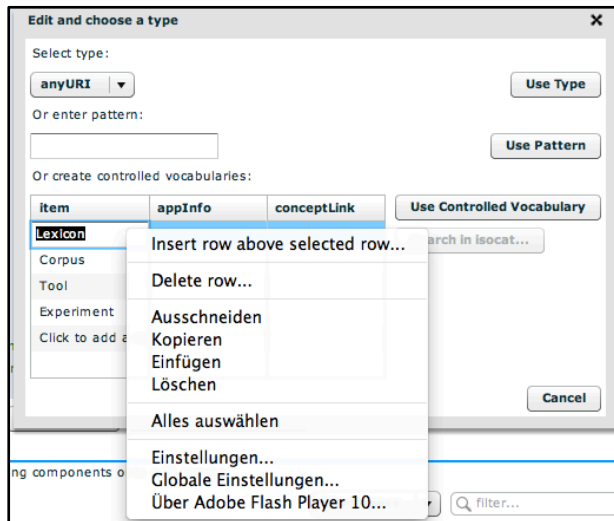


Abb. 31: Bearbeitungsmöglichkeiten von kontrolliertem Vokabular

Um kontrolliertes Vokabular bearbeiten zu können, kann man mittels eines Rechtsklicks mit der Maus auf den zu ändernden Begriff weitere Bearbeitungsoptionen aufrufen (s. Abb. 31). Die zur Verfügung stehenden Optionen beziehen sich entweder auf das Bearbeiten einer gesamten Zeile (z.B. das Einfügen einer neuen Zeile oberhalb der ausgewählten Zeile oder das Löschen einer Zeile, s. Abb. 31) oder das Editieren eines einzelnen Begriffs (z.B. das Ausschneiden, Kopieren, Einfügen oder Löschen eines Begriffs, s. Abb. 31).

Die festgelegten Werte des kontrollierten Vokabulars können in der Component Registry zwecks der Verständlichkeit für andere Nutzer weiter erläutert werden. Dazu ist neben der Spalte *Item* die Spalte *appInfo* zu finden (s. Abb. 30). Diese gibt Informationen über die jeweiligen zu verwendenden Items innerhalb des kontrollierten Vokabulars an. Beispielsweise kann im Falle der ISO-639-3-Codes für Sprachen mit Hilfe des Attributs *appInfo* als Erläuterung der volle Name des Codes angegeben werden (s. Abb. 32).

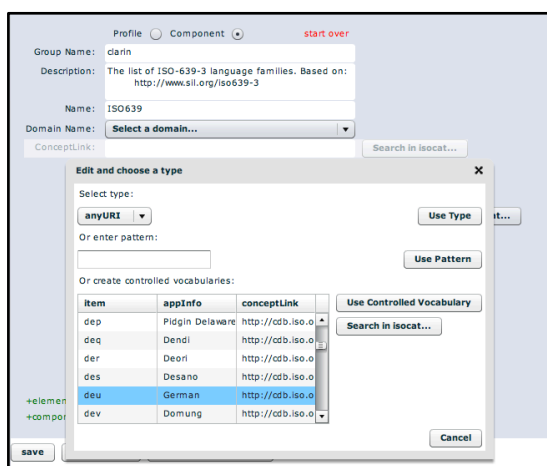


Abb. 32: Beschreibungen von kontrolliertem Vokabular

So wird z.B. für den ISO-639-3-Code *deu* die Bezeichnung *German* angegeben (s. Abb. 32), da es sich um den Code zur Kennzeichnung der deutschen Sprache handelt. Zusätzliche Informationen zu einem Wert innerhalb des kontrollierten Vokabulars kann man erhalten, wenn man dem jeweiligen Link in der Spalte *conceptLink* folgt. Es sei jedoch darauf hingewiesen, dass die Angabe eines *conceptLink* für kontrolliertes Vokabular nicht in jedem Fall zu empfehlen ist. Für die erläuterten ISO-Codes ist die Angabe sinnvoll, weil sie innerhalb der ISO definiert sind und die Links auf die ISO-Webseiten verweisen. Jedoch finden sich innerhalb der Component Registry für kontrolliertes Vokabular auch an einigen Stellen Verweise auf Konzepte in ISOcat, die mittels des Buttons *Search in isocat...* eingefügt werden können. Dieser Button ist zunächst grau unterlegt und wird erst anklickbar, wenn ein Begriff in der Spalte *Item* angegeben wurde und man in der entsprechenden Zeile das noch freie Feld in der Spalte *conceptLink* anklickt. In diesem Fall öffnet sich durch Anklicken des Buttons *Search in isocat...* ein neues Fenster, das eine Suchoption innerhalb von ISOcat einbindet (s. Abb. 33).

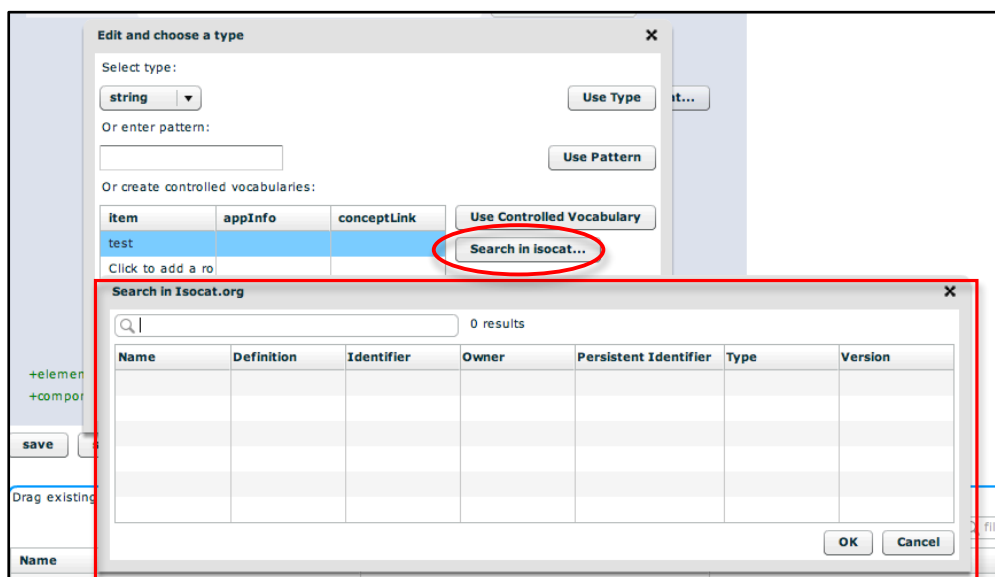


Abb. 33: Suchoption in ISOcat

Zur Erhaltung der Konsistenz sollte diese Vorgehensweise jedoch vermieden werden, da ISOcat zum jetzigen Zeitpunkt nur Datenkategorien enthält, jedoch nicht die zugelassene Werte (d.h. kontrolliertes Vokabular) von Datenkategorien. Somit wäre die Angabe eines Links auf ISOcat für den Wert einer Datenkategorie genau genommen nicht zulässig, da es sich um einen Verweis auf die Datenkategorie selbst handelt.

#### Multilingual (xml:lang)

Zurückkehrend zur Definition von Elementen bietet die Component Registry eine weitere wichtige Funktionalität im Editiermodus an: die Festlegung der Multilingualität eines Elementinhaltes (s. Abb. 34).

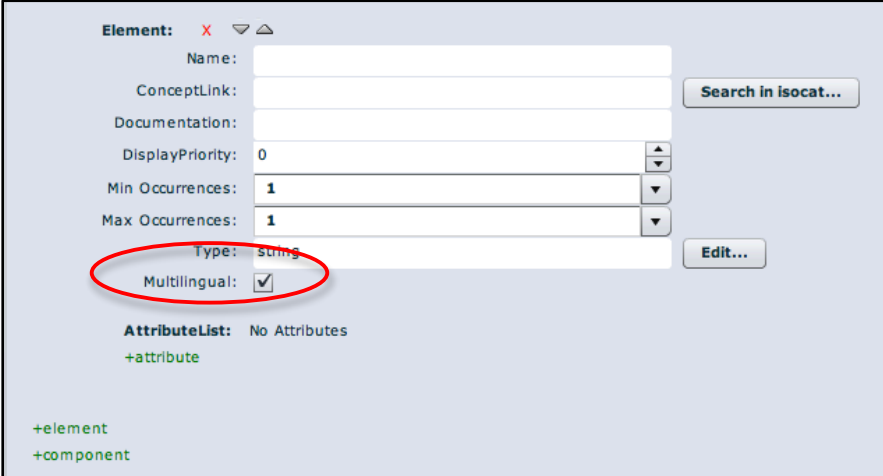


Abb. 34: Multilingual-Checkbox

Multilingualität bedeutet hier, dass ein Element einmal oder mehrmals verwendet werden kann, um den Elementinhalt in verschiedenen Sprachen anzugeben. Dabei wird die Sprache durch Verwendung des Attributs *xml:lang* in den CMDI-Metadateninstanzen im XML-Format identifiziert. Beispielsweise kann auf diese Weise ein Element *resourceTitle* in der Form `<resourceTitle xml:lang="de">...</resourceTitle>` durch das Attribut *xml:lang* mit dem Wert *de* angeben, dass der Elementinhalt auf Deutsch verfasst ist. Für diese Verwendung ist es zuvor jedoch notwendig, dass im Editiermodus in der Component Registry für das entsprechende Element in der Multilingual-Checkbox durch Anklicken der Box ein Häkchen gesetzt wurde (s. Abb. 34). Dies hat zur Folge, dass die Verwendung von *xml:lang* in den Metadateninstanzen optional ist. Setzt man das Häkchen jedoch nicht, ist der Gebrauch von *xml:lang* gegen das Schema nicht valide. Daher ist es zu empfehlen, im Zweifelsfall die Multilingual-Checkbox zu aktivieren, um eine optionale Verwendung des Attributs zu ermöglichen. Für Elementinhalte, die als Typ *string* definiert haben, ist die Aktivierung der Multilingual-Checkbox besonders zu empfehlen. Wenn als Typ hingegen kontrolliertes Vokabular vorliegt, wird im Editiermodus der Component Registry die Multilingual-Checkbox nach momentanen Entwicklungsstand automatisch nicht mehr sichtbar sein, da die zulässigen Werte, und damit auch die Sprache, eines Elementinhalts vorab festgelegt werden.

#### Attribute

Neben der Verwendung des Attributs *xml:lang* können Elemente in CMDI stets auch weitere Attribute besitzen. Als Default ist in der Component Registry zwar das Vorhandensein von Attributen nicht eingestellt (s. Abb. 34, *AttributeList: No Attributes*), aber ein Hinzufügen ist durch Anklicken der grün hervorgehobenen Option *+attribute* möglich (s. Abb. 34). Hat man diese Option angeklickt, öffnen sich weitere Bearbeitungsoptionen unter *AttributeList* (s. Abb. 35).



Abb. 35: Hinzufügen von Attributen

Hier kann man Attribute durch Festlegung des Namens und des Typs hinzufügen. Als Default ist der Typ *string* angegeben, der jedoch wie auch bei der Festlegung des Typs von Elementinhalten über den Edit-Button (s. Abb. 35) bearbeitet werden kann. Durch erneutes Anklicken der Option *+attribute* können beliebig viele Attribute für ein Element hinzugefügt werden.

#### Hinzufügen weiterer Elemente

Soll eine Komponente mehr als ein Element enthalten, besteht im Editiermodus der Component Registry die Möglichkeit, durch Anklicken der grün hervorgehobenen Option *+element* (s. Abb. 34) beliebig viele weitere Elemente hinzuzufügen. Für jedes neue Element öffnet sich ein neuer Bearbeitungsbereich, der die vorherigen bereits erläuterten Bearbeitungsoptionen für Elemente enthält (s. Abb. 24).

#### Komponenten

Für das Erstellen neuer bzw. das Hinzufügen existierender Komponenten innerhalb von Komponenten/Profilen stehen im Editiermodus der Component Registry zwei unterschiedliche Funktionen zur Verfügung.

#### Erstellen neuer und Hinzufügen bestehender Komponenten (+component-Funktion)

Erstens kann eine Komponente wie auch im Fall von Elementen oder Attributen durch Anklicken der grün hervorgehobenen Option *+component* (s. Abb. 34) entweder neu erstellt oder eine bereits vorhandene hinzugefügt werden. Durch Aktivieren dieser Option öffnet sich ein Bearbeitungsbereich für die zu definierende Komponente (s. Abb. 36).




Abb. 36: Bearbeitungsoptionen einer Komponente

Die Bearbeitungsoptionen umfassen dabei u.a. die Namensgebung der Komponente sowie das Festlegen des minimalen (s. Abb. 36, *Min Occurrences*) bzw. maximalen Auftretens (s. Abb. 36, *Max Occurrences*) der Komponente. Ebenso können weitere Elemente oder auch Komponenten innerhalb dieser Komponente durch die Optionen

+*element* bzw. +*component* hinzugefügt werden (s. Abb. 36). Nach aktuellem Entwicklungsstand der Component Registry existiert außerdem das Feld *ConceptLink* mit dem Button *Search in isocat...* (s. Abb. 36). Diese Benennungen sind irreführend, da ein *ConceptLink* mit Referenz auf ISOcat nur im Falle von Datenkategorien verwendet wird. An diese Stelle wird in der nächsten Version der Component Registry stattdessen ein Textfeld zur Angabe der *ComponentId* treten, welche als Wert die PID einer Komponente enthält. Der momentan sichtbare Button *Search in isocat...* mit der Browse-Funktion in ISOcat wird dann durch eine Suchmöglichkeit für Komponenten innerhalb der Component Registry ersetzt.

#### Einbindung existierender Komponenten (Drag and Drop-Funktion)

Die zweite Möglichkeit zur Einbindung bereits existierender Komponenten innerhalb einer Komponente oder eines Profils wird durch eine sogenannte „Drag and Drop“-Funktion ermöglicht.

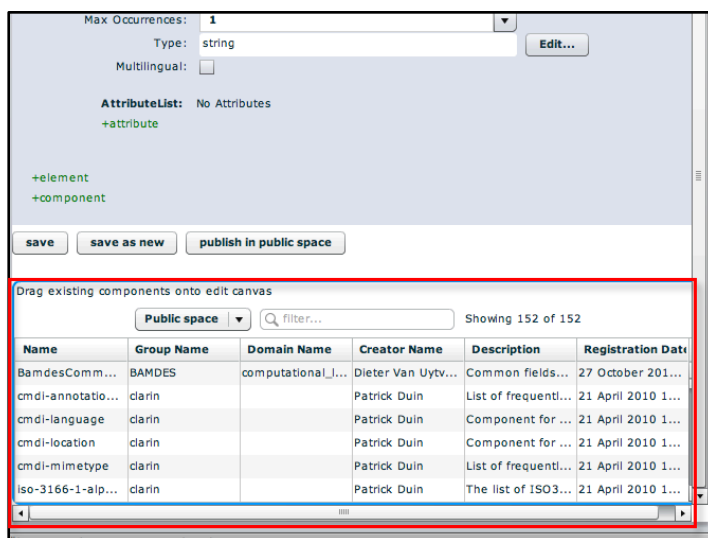


Abb. 37: Übersicht über bereits existierende Komponenten im Editiermodus

Innerhalb des Editiermodus in der Component Registry ist unterhalb der Editieransicht im Browse-Modus eine tabellarische Übersicht über Komponenten zu sehen (s. Abb. 37), die sowohl im öffentlich zugänglichen *Public Space* als auch im privaten *Work Space* vorhanden sind. Eine gewünschte existierende Komponente, die zu der neuen Komponente im Editiermodus hinzugefügt werden soll, wird zunächst durch Anklicken mit der linken Maustaste ausgewählt (s. Abb. 38, Komponente *Access*). Durch das Drag-and-Drop-Verfahren kann sie dann hinzugefügt werden, indem man die linke Maustaste nach Anklicken der ausgewählten Komponente gedrückt hält und die Komponente in den Editierbereich hineinzieht (s. Abb. 38, oben).

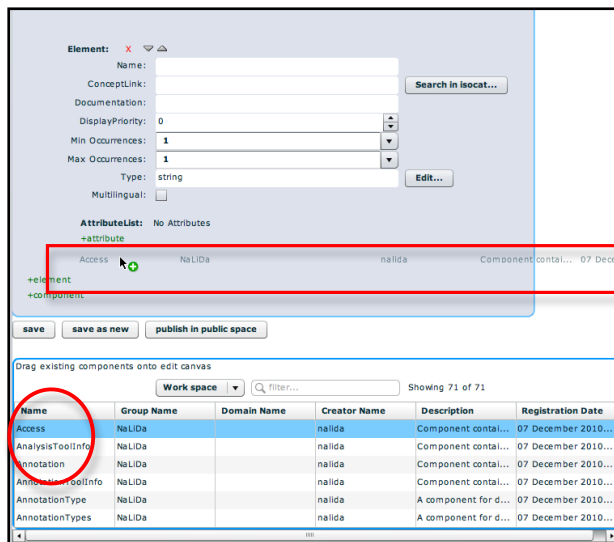


Abb. 38: Hinzufügen einer existierenden Komponente mittels Drag and Drop I

Löst man nach dem Platzieren der Komponente an die gewünschte Stelle die Maustaste, so wird die Komponente an die entsprechende Stelle positioniert (s. Abb. 39).

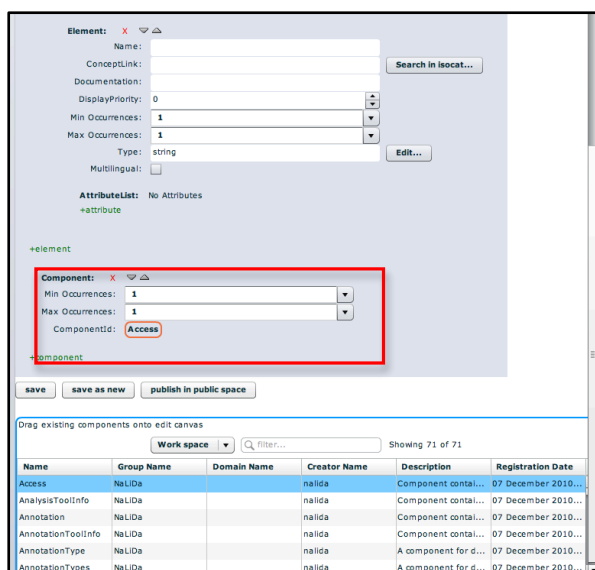


Abb. 39: Hinzufügen einer existierenden Komponente mittels Drag and Drop II

### Speichern von Komponenten und Profilen

Zum Speichern einer Komponente oder eines Profils existieren in der Component Registry drei unterschiedliche Optionen, die jeweils durch Anklicken eines Buttons realisiert werden: *save*, *save as new* und *publish in public space* (s. Abb. 40).

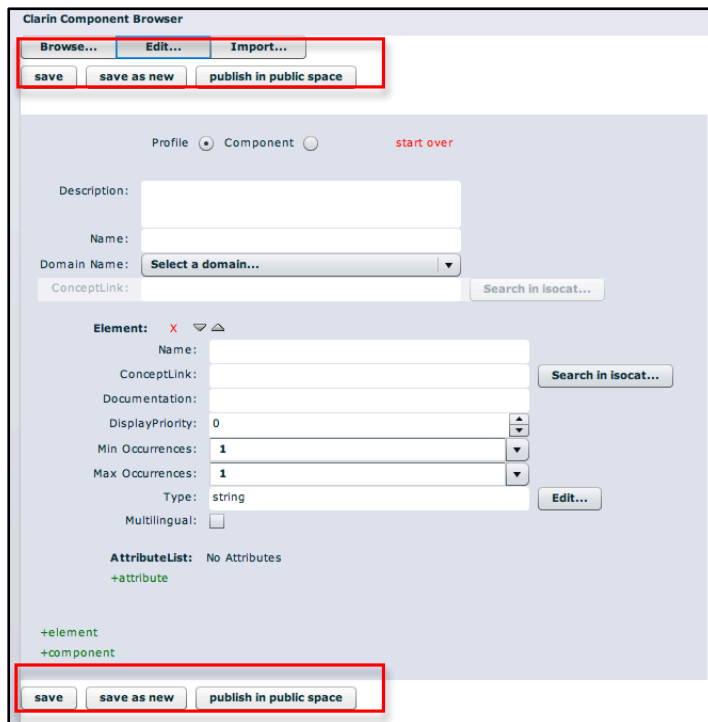


Abb. 40: Speicheroptionen

Die drei Buttons sind sowohl in der Kopfleiste als auch innerhalb der Fußleiste des Component Browsers platziert (s. Abb. 40), um dem Nutzer ein vermehrtes Scrollen zu ersparen.

### Save

Die Funktion *save* (s. Abb. 40) ermöglicht sowohl das Speichern einer neu angelegten als auch einer bereits existierenden Komponente, an der Änderungen vorgenommen werden sollen. Beide Optionen unterstützen das Speichern von Neuanlegungen oder Überschreibungen nur im privat zugänglichen *Work Space* und treffen ebenso auf die Bearbeitung von Profilen zu. Wird eine Komponente/ein Profil im *Public Space* editiert und dieser Vorgang mit *save* gespeichert, so wird lediglich eine Kopie dieser modifizierten Komponente/dieses modifizierten Profils im *Work Space* erzeugt. Die ursprüngliche Version im *Public Space* bleibt jedoch unverändert.

### Save as new

Die zweite Funktion *save as new* (s. Abb. 40) unterstützt das Anlegen neuer Komponenten/Profile ohne mögliche vorherige Versionen zu ersetzen. Diese Funktion eignet sich besonders, wenn man eine existierende Komponente/ein existierendes Profil aus dem eigenen *Work Space* als Grundlage nehmen und für andere Zwecke modifizieren möchte. Änderungen können so durchgeführt werden und werden als neue Komponente/neues Profil mit einer neuen PID gespeichert. Somit bleibt die ursprüngliche Version erhalten und die neue modifizierte Version wird ebenso im *Work Space* gespeichert.

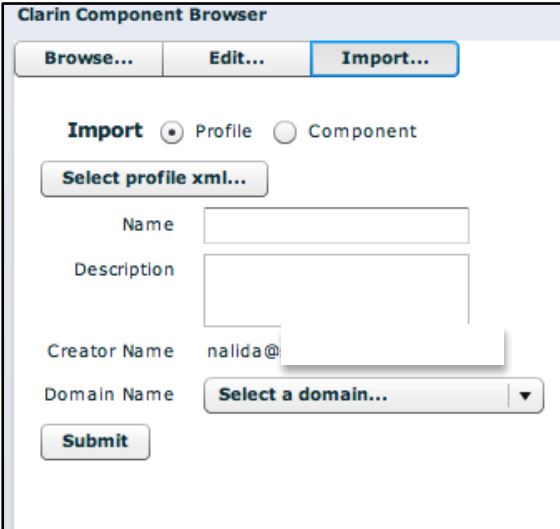


### Publish in public space

Die bisher erläuterten Speicheroptionen beziehen sich alle auf das Sichern von Komponenten/Profilen im privaten *Work Space*. Möchte ein Nutzer seine Komponenten/Profile der Öffentlichkeit im *Public Space* zur Verfügung stellen, kann die dritte Speicherfunktion ausgewählt werden: *publish in public space* (s. Abb. 40). Diese Option ermöglicht zwei unterschiedliche Vorgehensweisen zur Veröffentlichung von Komponenten/Profilen in der Component Registry: erstens umfasst sie das direkte Veröffentlichen neu erstellter Komponenten/Profile im *Public Space* (d.h. die Komponenten/Profile müssen sich vorher nicht im *Work Space* befunden haben, sondern können direkt im *Public Space* gespeichert werden). Von dieser Methode sei jedoch abzuraten, da einmal veröffentlichte Komponenten/Profile nicht mehr editiert werden können. Besonders im Entwicklungsprozess von Komponenten/Profilen ist dies ein gravierender Nachteil. Daher ist die zweite Vorgehensweise zu bevorzugen. Diese ermöglicht es, dass Komponenten/Profile, die zunächst nur im *Work Space* abgespeichert werden, ebenso veröffentlicht werden können. Die bestehende Komponente/das bestehende Profil wird dazu im Editiermodus geöffnet (d.h. indem durch einen Rechtsklick auf die Komponente/das Profil die Option *Edit Item* ausgewählt wird) und dann im Editiermodus durch Anklicken des Buttons *publish in public space* (s. Abb. 40) im öffentlichen Bereich publiziert. Es ist zu beachten, dass durch diesen Vorgang ein Verschieben der Komponente/des Profils vom privaten *Work Space* in den *Public Space* stattfindet (d.h. die Komponente/das Profil ist nur noch über den *Public Space* zugänglich und erscheint nicht mehr im *Work Space*).

### c. Import: Importieren lokal erstellter Komponenten/Profile

Die dritte Funktionalität, dargestellt durch den Import-Tab, erlaubt das Importieren von lokal erstellten Komponenten und Profilen in die Component Registry. Voraussetzungen für einen erfolgreichen Import sind, dass man als Nutzer eingeloggt sein muss und die zu importierenden Dateien im CMDI-Format als XML-Dateien vorliegen müssen.



The screenshot shows the 'Clarin Component Browser' window with the 'Import...' tab selected. The 'Import' section has two radio buttons: 'Profile' (selected) and 'Component'. Below this is a 'Select profile xml...' button. The form includes fields for 'Name', 'Description', 'Creator Name' (with the value 'nalida@'), and 'Domain Name' (with a dropdown menu showing 'Select a domain...'). A 'Submit' button is at the bottom.

Abb. 41: Importfunktion für Profile

Die Component Registry stellt nach Auswahl des Import-Tab ein Webformular für den Import zur Verfügung (s. Abb. 41). Zunächst wird mittels eines Radio Buttons festgelegt, ob ein Profil oder eine Komponente hochgeladen werden soll. Danach wird über den Button *Select profile xml...* im Fall von Profilen (s. Abb. 41) bzw. *Select component xml...* (s. Abb. 42) im Fall von Komponenten die entsprechende Datei ausgewählt. In den folgenden Textfeldern wird der Name des Profils/der Komponente sowie eine Beschreibung zur intendierten Verwendung angegeben (s. Abb. 41). Der Name des Erstellers (s. Abb. 41, *Creator Name*) wird automatisch mit dem Benutzernamen gleichgesetzt, der für den Login in die Component Registry verwendet wird. Optional besteht zusätzlich die Möglichkeit, die Komponente bzw. das Profil einer Domäne zuzuordnen (z.B. *Applied Linguistics, Computational Linguistics, Lexicography, Psycholinguistics, Syntax, etc.*). Diese Domäne kann über ein Dropdown-Menü (s. Abb. 41, *Select a domain...*) ausgewählt werden. Nach aktuellem Entwicklungsstand kann für Komponenten außerdem ein sogenannter *Group Name* angegeben werden (s. Abb. 42), der z.B. die Arbeitsgruppe kennzeichnen kann, in der eine Komponente/ein Profil erstellt wurde. Diese Option soll in einer neuen Version der Component Registry auch für Profile angeboten werden.

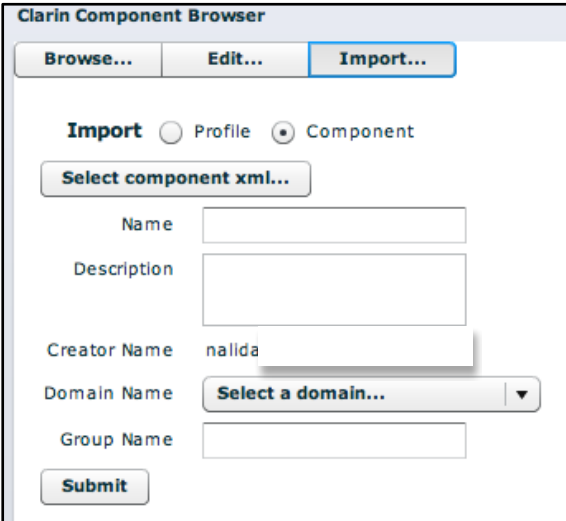


Abb. 42: Importfunktion für Komponenten

Den letzten Arbeitsschritt beim Import einer Komponente/eines Profils bildet das Anklicken des Submit-Buttons (s. Abb. 42). Dieser aktiviert den eigentlich Importprozess in die Component Registry und bewirkt das Speichern der importierten Datei in den privaten *Work Space*. Dabei wird gleichzeitig eine PID für die Komponente/das Profil vergeben.

Bei der Import-Funktion ist außerdem darauf zu achten, dass nur die Inhalte übertragen werden, die auch in der ursprünglichen Datei vorhanden waren. Daher sollte man vorab prüfen, ob in den zu importierenden XML-Dateien z.B. alle notwendigen Attribute für Elemente angegeben sind (z.B. das Multilingual-Attribut). Nach einem bereits erfolgten Import können diese Änderungen alternativ noch im Editiermodus der Component Registry vorgenommen werden.



Nach aktuellem Entwicklungsstand der Component Registry ist abschließend ein weiterer Aspekt hervorzuheben: wenn man beim Importieren einer Datei für die Komponente/das Profil einen Namen im Webformular (s. Abb. 41 und 42) vergeben hat, der nicht dem Wurzelement der Komponente/des Profils entspricht, wird dieser Name zunächst wie eingegeben abgespeichert. So kann beispielsweise für eine Komponente *Descriptions* als (Datei-)Name *nalida\_descriptions* gewählt worden sein, um eine abgeleitete Komponente (z.B. die von CLARIN entwickelte Komponente *cmdi-description*) innerhalb eines Projekts (z.B. dem NaLiDa-Projekt) kennzeichnen zu können. Öffnet man die importierte Datei danach jedoch im Editiermodus der Component Registry, führt Änderungen durch und speichert die Datei erneut ab, wird der zuvor eingegebene Name automatisch durch das Wurzelement ersetzt. Somit würden im Beispiel sowohl der vergebene Name als auch das Wurzelement der Komponente *Descriptions* lauten. Versucht man diesen Vorgang rückgängig zu machen, indem man die Komponente nochmals editiert und manuell den Namen erneut eingibt (d.h. *nalida\_descriptions*), wird das Wurzelement der Komponente mit diesem Namen gleichgesetzt (d.h. *nalida\_descriptions*). Daher ist von dieser Vorgehensweise aufgrund der aktuellen Funktionalitäten der Component Registry abzuraten. Stattdessen empfiehlt es sich, für die Zugehörigkeit zu einem Projekt oder einer Arbeitsgruppe das Feld *Group Name* auszufüllen (s. Abb. 42). Weitere Optionen zur Kennzeichnung von abgeleiteten Komponenten/Profilen und ihrem Entwicklungsstatus befinden sich noch in der Entstehung.

## 9. Referenzen

Informationen zu CMDI:

- <http://www.clarin.eu/cmdi> (Dokumentation, Beispiele, FAQ, Forum, XML Toolkit)
- <http://www.sfs.uni-tuebingen.de/nalida/blog.html> (Blogartikel, u.a. zu CMDI)
- <http://www.sfs.uni-tuebingen.de/nalida/doku/glossar.html> (Glossar, u.a. zu CMDI)
- <http://www.sfs.uni-tuebingen.de/nalida/doku/workshops.html> (Workshop zu CMDI)

Informationen zur Component Registry:

- <http://www.clarin.eu/cmdi> (Dokumentation für Nutzer und Entwickler)
- <http://www.sfs.uni-tuebingen.de/nalida/blog.html> (Blogartikel)

Zugang zur Component Registry:

- <http://catalog.clarin.eu/ds/ComponentRegistry/#> (direkter Zugang)
- <http://www.clarin.eu/cmdi> (Link *Component Registry* anklicken)

Zugang zu ISOcat:

- <http://www.isocat.org/interface/index.html>

Zugang zu Arbil:

- <http://www.clarin.eu/cmdi> (*Start Link* unter *Arbil metadata editor* anklicken)



## 10. Kontakt

Für Rückfragen und Anregungen steht Ihnen das NaLiDa-Projekt gerne zur Verfügung:

DFG-Projekt NaLiDa (Nachhaltigkeit Linguistischer Daten)  
Eberhard Karls Universität Tübingen  
Seminar für Sprachwissenschaft  
Allgemeine Sprachwissenschaft und Computerlinguistik  
Wilhelmstraße 19  
72074 Tübingen

E-Mail: [nalida@sfs.uni-tuebingen.de](mailto:nalida@sfs.uni-tuebingen.de)

Webseite: <http://www.sfs.uni-tuebingen.de/nalida/>

Ansprechpartner Tutorial:  
Christina Hoppermann

Projektansprechpartner:  
Reinhild Barkey  
Christina Hoppermann  
Dr. Thorsten Trippel

Projektleitung:  
Prof. Dr. Erhard Hinrichs