– Diversity Workbench –
The Resource management module
DiversityResources
[preliminary & fragmentary version!]

by G. Hagedorn
Version 0.2, 16. Oct. 2001

Introduction

DiversityResources provides an online access management system for document oriented information. In the context of an information system, it allows the integration of documents like formatted text, images, or audio/video streams. All resources can be integrated using structured lists of pointers (hyperlinks), but certain resources (esp. text and images) may also be directly embedded in documents generated by the information system.

The term resource attempts a generalization for digital information that is online available. It is generally used in the internet in the term for addresses of web content, the URL (= Universal Resource Locator) or URI (= Unified Resource Identifier). The older term 'URL' focuses maps a physical address of protocol, server, path, and file name directly on the internet name. The newer concept of URI states that although any URL is also a URI, the physical mapping does not need to hold. The URI is only a unique identifying name for a resource, which may in fact be present in a different physical location. In most cases the web server translates URI addresses transparently, so that the difference is not noticed by the client. The difference between URL and URI is, however, noticeable in services using content negotiation, i.e. the client and server negotiate which of several available formats will be transferred. For example, content negotiation URIs are used to provide the 'png' image file format to those clients that can read it, or to provide html, xhtml, or xml versions of a document, depending on the client browser.

The Resource management module is based on URI addresses. The independence of address and physical location has the practical implication, that it is meaningful to manage certain digital offline resource, e.g. images that reside on a CD, as long as the URI remains unique. Some functionality like online-link checking will naturally fail, but the information can be integrated. Later, when the resource is made available online, an address translation on the web server can translate this URI to a physical address.

Differences between the resource and reference management modules

The management of information sources is primarily the task of the references module DiversityReferences. A reference can be not only be a printed book or article, it can also be a collection of published photographic slides, an unpublished laboratory notebook, or a digital web page. By connecting information with its source, references allow to acknowledge the work of others, to find more detailed information, and to recognize errors and misconceptions that may be derived from certain information sources.

DiversityReferences deals mainly with abstract information items, not with actual instances of the information source. It is generally assumed that all instances of the reference (e.g. a book in a certain edition) are identical. The module dealing with instances (entity 'ReferenceAvailability') has been added for practical purposes.

In contrast, the primary task on the resource module DiversityResources is to provide management of online resources. Resources are document hyperlinks, images, audio or video streams, or processing resources like query interfaces, utilities for interactive identification. All resources are assumed to be accessible by means of a URI (Universal resource identifier) in the internet. In practice, the storage location may change through time, requiring special mechanisms to manage these changes. Due to the nature of the internet, the same abstract resources (e.g. a drawing) may exist in several quality levels, representing different compromises between speed of access (e.g. thumbnail image) and provided information (e.g. high resolution image).

The following table summarizes some differences between the Reference and the Resource module:

Table X: Comparison of Reference and Resource module

	Reference module	**Resource module**
Focus	Abstract item of which an unknown number of instances may exist	Actual instance of resource
Intellectual property	All information is supposed to be public, citing the information does not require permission from intellectual property holder	Linking to a digital online resource does not require permission, but may require some legal considerations (e.g. regarding trademarks or illegal information) Embedding a resource (e.g. an image) requires permission from the copyright holder.
Acknowledgement	is implicit in the	special care must be taken to

	standardized way of citing a reference	appropriately acknowledge information
Access, Availability	added value, no directly linked to purpose of module	Main focus of module
Immediacy of Access	The referenced information resources may not be accessible at all (e.g. a thesis from a foreign university)	Only resources of known location are relevant, online access should be immediate. Special care (link checking mechanism) must be taken to guarantee the "quality of service"

Note: *It must be noted that some unsolved problems of overlap in the definitions of modules exist. In practice, however, these problems are relatively rarely of any consequence. This may change in the future, when online information resources will not just be an occasional source for the references module, but will become the mainstay. An integration of both modules may then become desirable. At the moment it was considered more pragmatic to design separate modules, to adequately explore the necessities of each application of information management. An integrated system would be considerably more complex and should be attempted after the current analyses and implementations have reached a satisfactory level.*

The information model

The main entities of the model (and terms used)

The model distinguishes between abstract resources items (**ResourceAbstractItem**) and resource instances (**ResourceInstances**) that have an actual storage location or access method. The abstract resource describes those properties that are identical for all copies or quality versions of a resource. For an image these are creator, or a description of the content. Instances can be the original web page and a copy on a mirror site, or different compression levels or resolutions of images.

Typically instances of an abstract resource item can be created by automatic means, e.g. changes of image resolution, colour depth, compression level, or media format type. Other manipulations (for images e.g. despeckling or cropping of an image to remove unused border area) may not be fully automated, but are also assumed to fall into this category. These changes do not require changes in the content description or the intellectual property right documentation. In contrast, any changes that substantially change a resource and contain a creative element must be registered as new, independent abstract items. (The relationship of such related items may optionally be recorded in the attribute **ParentResourceID**.)

To simplify the management, resources must always be contained in collections (**ResourceCollection**). The collection defines properties that are identical for all resource items in a collection. Ideally a collection should be defined by the origin and content of the

resources, and resources in a collection should have similar URIs to identify them. An example would be a collection of microscopic drawings of a group of fungi from a single author or a working group. However, it is also possible to use "personal" default collections like "My web links" or "My images".

Certain properties of instances rather than items in a collection depend on the quality level (e.g. for images resolution, compression, or media format) of the instance. For each collection, these properties can be defined once in **ResourceQualityClass** and are applied to all instances of each abstract resource item. The most important property is the option to define URI components for different quality classes (e.g. different folders for image resolutions).

The four parts of these model can be visualized as a 2-dimensional table, with the resource instances forming the body, quality classes forming the column headers, abstract items forming the row headers and collection defining information that applies to the entire table (see Fig. ResEnt1). The presence of multiple resource collections can in turn be visualized using a 3-dimensional table (see Fig. ResEnt2).

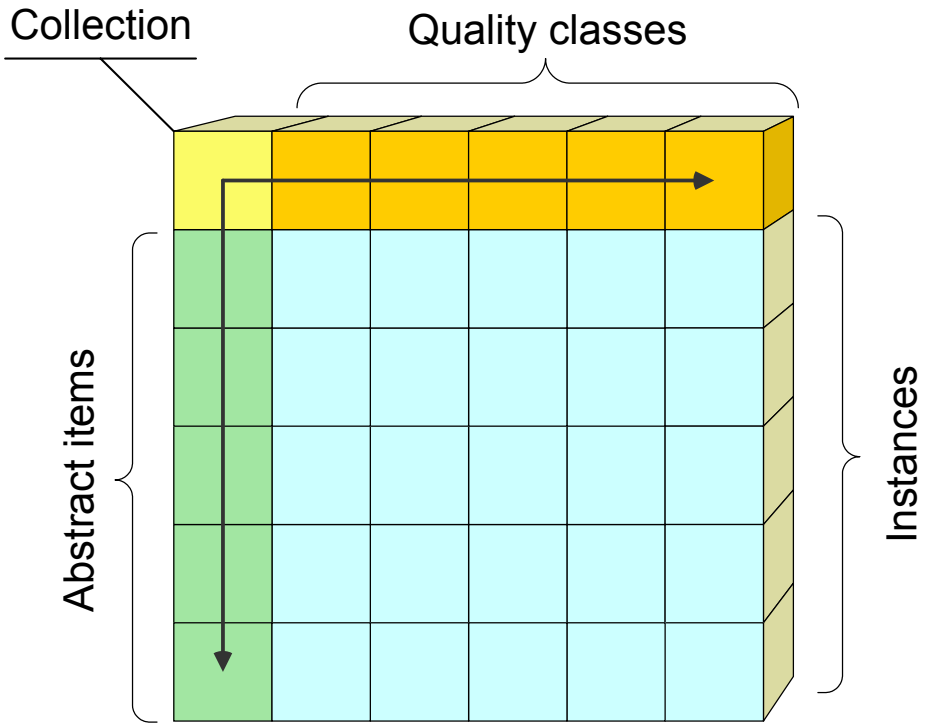

Fig. ResEnt1: Visualization of 4 entities as elements of a 2-dimensional table.

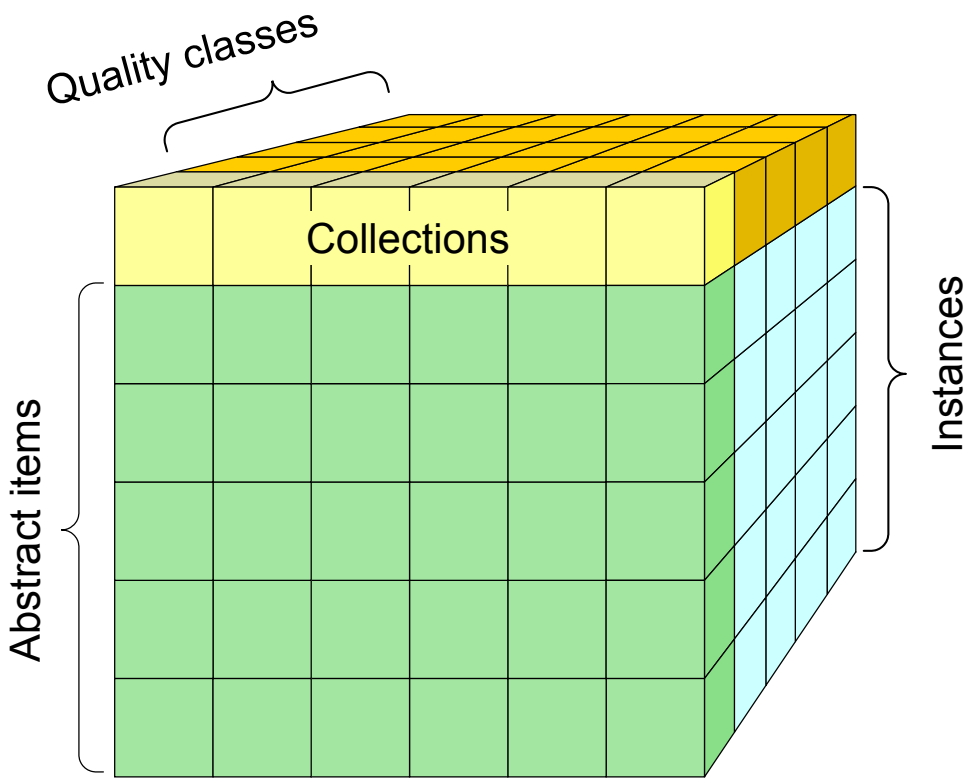

Fig. ResEnt2: The central resource model with multiple collections can be visualized in the 3rd dimension.

The model is not designed to store resource collections with mixed quality levels. Although it is principally possible to have missing resource instances like in Fig. ResEnt3, but the instances have to be manually inserted or deleted in that case. Normally the resource manager assumes an orthologous model with an instance for each quality class and item.

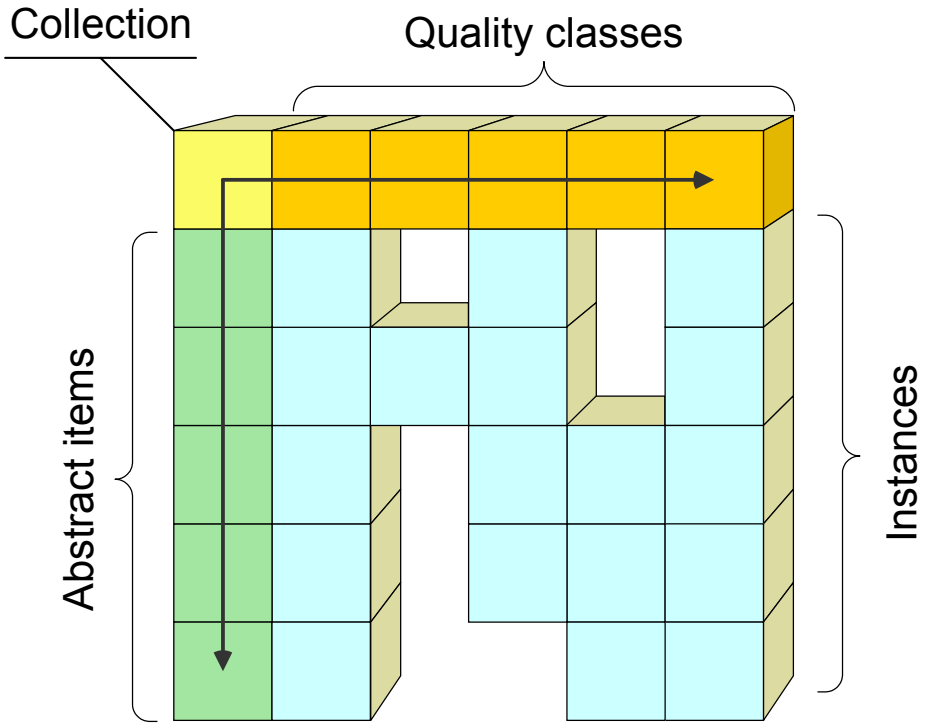

Fig. ResEnt3: Abstract resource items with instance of multiple, but variable quality levels. Such resources should normally be managed in multiple resource collections.

The resource instances, i.e. links to web pages or images, are the central object that is requested from the resource manager. In object terminology one can say that the resource instances show multiple inheritance from both ResourceQualityClass and ResourceItem, and that ResourceItem inherits information from ResourceCollection.

> ***Note:*** *The term 'instance' in ResourceInstance is intended to clarify the multiple inheritance of properties from Item, Collection, and Quality. It should not be confused with object oriented information modelling terminology. All entity types (ResourceInstance as well as ResourceQualityClass) are classes insofar as each actual object (e.g. a file containing an image) is an instance of this type. Thus, several files can be instances of the entity class "ResourceInstance".*

The atomic structure of resources and resource parts

Parts are important e.g. in identification interface. An interactive identification program may use a clickable hyperlink map on top of an image to allow the user to make a selection among several possible states of a character.

Display entities and multilingual definitions

@@@@@@@

Trigger definitions

[Later layout in a table, with insert/update/delete symbols and association with the entities]

To do: Everything that has Cache at the end needs a trigger, follow dependencies!

When EditorialControl is set to false, EditorialReleaseBy and EditorialReleaseWhen are updated.

When EditorialControl is false, EditorialReleaseBy and EditorialReleaseWhen are write-protected.

When inserting new resource quality class to a resource collection, attempt to add instances for all abstract items of that collection

When inserting new abstract items, attempt to add instances for all quality classes of the resource collection.

Details of the information model

The discussion of the information model is structured as follows:

- The chapter "The resource collection" discusses information concerning the entire resource collection (e.g., "*Ascochyta* line drawings by V.A. Mel'nik, 1977") and its quality classes (e.g. thumbnails, medium resolution images, high resolution images).
- The chapter "The resource item and description" discusses information concerning an abstract resource item (e.g., a photograph, regardless of storage location or resolution in which it is available), its parts (e.g. multiple drawings in a single figure), and its description (caption of an image, keywords to retrieve a resource, etc.).
- The chapter "The resource instance" discusses the digital representations of the abstract item (e.g., the thumbnail resolution of a photograph).

Two chapters discuss functionality that requires information from multiple entities and is therefore discussed only briefly in the first chapters:

- The definition of the URI for a resource instance based on collection, quality class, abstract item, and instance specific parts is discussed in the chapter "Introduction to the URI definition".
- The definition of availability for the collection, the quality class, and the abstract item is discussed in the chapter "Availability management".

The resource collection

The resource collection allows to define information about a large set of resources in a single place. The information can be grouped into four main areas:

- Attributes describing the collection itself (the creation of a collection is an effort that gives rise to intellectual property rights independent of the rights to the resources)
- Attributes describing origin and intellectual property rights of the resources contained in the collection
- Attributes describing storage location and availability of the resources contained in the collection
- Attributes depending on quality classes (e.g. different resolutions of an image)

Some attributes are relevant predominantly, if resources are managed locally, not only information about (and links to) external resources.

Point for discussion: should each collection be either own or external resources? Should an extra attribute be introduced for this purpose? see preliminary **xx_ManagedContent** attribute!

Attributes describing the collection itself

Each resource is identified by a **ResourceCollectionID** which is automatically created by the database. The collection itself (i.e. not the collected resources) is described by a short **Title**, a longer **Description,** and the **Authors** or **Editors** that created the collection. The **Title** may be up to 80 characters long and must uniquely describe the collection within the entire resource management system (the DiversityResources database subsystem). The **Description** should go into more detail, and inform the user adequately about the scope and content of a description. It should be approximately one paragraph long. You can use basic xhtml formatting tags (, <br />, etc.; paragraph level tags like <p>, <li>, <hr />, etc. should be avoided). Note that the **Authors** or **Editors** of the collection may be different from the creators of the resources the collection provides access to. The **Authors** or **Editors** are the "collectors" of the resources, those persons which have invested the time to collect the resources, index them with keywords and descriptions, and who manage the collection to maintain access to the resources even when the URI of a resource may change.

Together with a publication date, which is derived from the creation date of the resource collection itself, the 4 elements **Title**, **Description, Authors**, and **Editors** are used to generate a title page of the collection. The title page is displayed only if a client (e.g. internet user) requests the entire collection. It informs about the collection, the number of items which are available in it, and provides access to related pages (index to items generated from item description keywords, a table of contents generated from item subheadings, and a starting page to directly browse through the collection from the first to the last page. The title page is not displayed if only resources from are found (e.g. through the keywords attached to resource items) and displayed.

Finally, the attribute **InternalNotes** provides are place to store any internal notes regarding the collection as a whole. Examples are a list of things that need to be done, or notes about problems. In the case of images it may also be of interest to record the digitization method (digital camera, flat bed scanner, slide scanner, Internet, Photo-CD, etc.) has been used. Internal notes are only visible to managers of the resource collection and managers of the web site.

Task: Write cfm code to generate a title page for a collection! Automatically generate number of items, and index, table of contents, and direct browsing! Plus (see below): if RefID is set, if should also be used on the title page!

Attributes describing origin and intellectual property rights of resources

The following attributes document the origin and intellectual property rights of the collected items. The most general attribute is the **SourceDescription**, which may contain any information about the origin of the resources being indexed in the collection. In many cases the material contained in a resource collection has already been published elsewhere. It is desirable to point to a reference already managed in the reference management module

(DiversityReferences). The attributes **SourceReferenceID** and **SourceRefDescriptionCache** (which is automatically set whenever **SourceReferenceID** changes) provide the means to do so. Please verify that either the copyright has expired or a copyright agreement has been reached and the material can be legally republished digitally!

The diversity workbench modules use a separate module, DiversityAcknowledgements, to provide both a documentation of external efforts, and to define the format of an acknowledgement. Such an acknowledgment is optional. If present, only the ID number (**AcknowledgementID**) is saved in the resource collection. **CopyrightDocumentation** is especially important if no separate acknowledgment is defined. It is extremely important for the future of a complex system with many collaborators, that the copyright or usage agreements are carefully documented. The documentation should include dates and the names of the responsible persons or institutions with which agreements were reached. If letters or emails are involved, these should be cited (include the senders date and possibly a filing location) or even quoted.

SourceDescription, **CopyrightDocumentation**, and **InternalNotes** may all three be considered being private to the managers of the resource collection and managers of the web site and will not be presented to the public. The reference defined in **SourceReferenceID**, and the formatted acknowledgment to which **AcknowledgementID** refers, may, however, be output.

Task: Does this require to generate acknowledgement definitions over the web? Perhaps not, but analyze!

Attributes describing storage location and availability of resources

The following attributes apply to the storage and availability of resources. Storage locations are available in three attributes: **URI_Part1**, **LAN_Part1**, and **BackupLocation**. The most fundamental is **URI_Part1,** defining the first part of the resource instance URI (see chapter "Introduction to the URI definition"). In many cases, local resources will also be available over a local area network (LAN). As long as all other parts of the URI except for **URI_Part1** are identical in the LAN and internet path, the **LAN_Part1** can be used to define an alternative access path. **LAN_Part1** should be left empty, if this access alternative is not available, either because the resource are not available in the LAN, or because the URI and the LAN path are not congruent.

BackupLocation is a free text field that allows to describe a location where an additional backup of the resources in the collection is stored. This may be a storage location and label of a CD-R, writeable DVD, backup tape, etc. It may also point to a path on a backup file server. **BackupLocation** is, however, not intended to provide a machine-readable location that is automatically used in the case of failure of the primary location defined in the URI.

Availability restrictions are defined in **ReleaseDate**, **ReleaseNotes**, **EditorialControl**, **EditorialReleaseBy**, and **EditorialReleaseWhen**. The first two are defined by managers of the collection and allow them to postpone public availability for up to two years (e.g. until another publication has appeared). The latter three are updated by managers of the web site and are read-only for resource collection managers. The editorial review mechanism implements the necessary editorial supervision of the website, preventing illegal or undesirable information from being published. This mechanism is especially important to prevent abuse if a mechanism to upload images by external users is provided. See the chapter "Availability management" for further information on the availability management.

Attributes depending on quality classes

As already discussed in the introduction (see chapter "The main entities of the model (and terms used)") some information that can be defined for an entire resource collection depends on the quality class. For example, the thumbnail, medium resolution, and high resolution instances of an image have properties that are identical for all items in the collection. The information is very similar to the generic resource collection attributes discussed above, except that they can be defined several times, for each quality class that is available. At least one quality class must be defined for each resource collection.

A class is defined by its **QualityLevel**. Resource items in a resource collection may have instances of several quality levels. A typical resource collection has between 1 and 4 quality classes. For each quality level the URI components **URI_Part2** and **URI_Part4** may be defined (see chapter "Introduction to the URI definition" for more information). The **RecommendedPresentation** defines a presentation mode for internet presentations. Currently defined modes are 'Link' (generate a text or icon based hyperlink), 'Embed' (esp. images can be directly integrated into the web page generated from the resource information) and 'Download' (provide a link that, when clicked upon initiates a download on the client computer rather than opening the target in the web browser). The application presenting information from the resource module may either follow these recommendations or implement its own rules.

Some quality levels, e.g. thumbnails of images, can be generated automatically from a higher quality level. If the application supports such a mechanism (the first version will probably not!) the attribute **AutoGenerateFrom** can define the quality level from which the resource can be generated.

Point for discussion: This would require further parameters: e.g. desired size of thumbnail, color depth, compression, etc. The parameters are media type dependend as well as conversion robot dependant → not yet introduced!

For each quality class the availability scope (Internet, Intranet, login) and access immediacy (online, on request) can be defined (**QualityAvailability**, **QualityAvailableOnline**). This enables managers to publish thumbnails and medium quality versions on the internet, but

reserve high quality versions for in-house use. The information for the quality classes is combined with availability restrictions that may be placed upon resource items (see Abstract resource item) and on the entire collection (see Attributes describing storage location and availability above). A full explanation of all components of the availability management can be found in the chapter "Availability management".

The **InternalNotes** are provided to store notes the resource managers may want to make about a specific quality class, for example the history of a quality class, especially from which other quality level it was generated, modifications performed, compression rate used, etc.

List of quality classes and their definition

Standard quality levels:

ID	*Description*	*PresentationDefa*
0	Undefined, especially in unstructured collection of resources of single or mixed quality	L
1	Lowest quality, abridged information preview, e.g. thumbnail size images	E
2	Lower quality version of resource, optimized for file size/internet use (e.g. jpg, mp3 with high compression)	E
3	Medium quality version of resource, optimized for screen use (e.g. gif/png with reduced resolution)	E
4	Highest quality of resource available (e.g. high resolution png/tiff images, wave-audio files)	L
5	Unprocessed version (e.g. files retained for backup purposes)	D

Additional quality levels are defined to support mirroring of a resource on one or two mirror locations. These quality levels are not normally available in the user interface pick lists:

ID	*Description*	*PresentationDefa*
10	[MIRROR COPY 1:] Undefined, especially in unstructured collection of resources of single or mixed quality	L
11	[MIRROR COPY 1:] Lowest quality, abridged information preview (e.g. thumbnail size images)	E
12	[MIRROR COPY 1:] Lower quality version of resource, optimized for file size/internet use (e.g. jpg, mp3 with high compression)	E
13	[MIRROR COPY 1:] Medium quality version of resource, optimized for screen use (e.g. gif/png with reduced resolution)	E
14	[MIRROR COPY 1:] Highest quality of resource available (e.g. high resolution png/tiff images, wave-audio files)	L
15	[MIRROR COPY 1:] Unprocessed version (e.g. files retained for backup purposes)	D
20	[MIRROR COPY 2:] Undefined, especially in unstructured collection of resources of single or mixed quality	L
21	[MIRROR COPY 2:] Lowest quality, abridged information preview (e.g. thumbnail size images)	E
22	[MIRROR COPY 2:] Lower quality version of resource, optimized for file size/internet use (e.g. jpg, mp3 with high compression)	E
23	[MIRROR COPY 2:] Medium quality version of resource, optimized for screen	E

	use (e.g. gif/png with reduced resolution)	
24	[MIRROR COPY 2:] Highest quality of resource available (e.g. high resolution png/tiff images, wave-audio files)	L
25	[MIRROR COPY 2:] Unprocessed version (e.g. files retained for backup purposes)	D

The resource item and description

An abstract resource item is @@@

(e.g., a photograph, regardless of storage location or resolution in which it is available),

its parts (e.g. multiple drawings in a single figure),

and its description (caption of an image, keywords to retrieve a resource, etc.)

The information on resource items can be grouped into three main areas:.

- Abstract resource item
- Item parts
- Item or item part description and keywords

@@@ TO DO @@@

Abstract resource item

Each resource has a unique number (**ResourceID**) that is automatically generated by the database for new records. The number uniquely identifies a resource item, regardless of the resource collection in which it has been placed. However, each item must be part of exactly one resource collection (**ResourceCollectionID**). It is not possible to manage resources without a resource collection.

The item specific part of the URI string is defined in **URI_Part3**. It is combined with **URI_Part1** defined in the resource collection, and **URI_Part2** and **URI_Part4** defined in the quality class, to form the full URI. If only a single quality class is present, **URI_Part2** and **URI_Part4** are usually empty, and **URI_Part3** contains everything that has not already been defined in the resource collection. In many cases **URI_Part3** will then contain a complete URI, perhaps except for the "http://" lead. However, if multiple quality classes are present in the resource collection, the **URI_Part3** contains only the common part of the different quality instance URIs. A frequent example would be the filename without an extension. See the chapter "Introduction to the URI definition" for a full explanation of URI definition.

Title

Occasionally it is unsatisfactorily to directly use the title to define the order in which hyperlinks are presented in a collection.

If filled, used to sort the resource in a presentation. Example: Title 'The Mycological Society of Washington DC' shall be sorted under 'Washington DC'. Normally the attribute remains empty and resource items are sorted by title.

DisplayOrderText

SourceCaption contains the full, unchanged text of the original caption of a resource (if an original caption exists). It is relevant especially in the case of images scanned from a publication, and should then contain the caption directly copied from the book. If the resource item has several parts (e.g. several figures in a single image file), **OriginalCaption** should contain the caption of all parts one after another.

For each abstract item the availability scope (**ItemAvailability**: Internet, Intranet, login) and access immediacy (**ItemAvailableOnline**: 'online' or 'on request') can be defined. This enables managers to prevent access to specific items in a resource collection, while publishing the other items. The availability definition for an item is combined with the availability restrictions that may be placed upon quality classes (see Attributes depending on quality classes) and on the entire collection (see Attributes describing storage location and availability above). A full explanation of all components of the availability management can be found in the chapter "Availability management".

The optional attribute **LinkAssociatedIcon** allows to define a resource specific icon for a resource item. This can, for example, be used to add a journals logo or icon to a list of online journals, or to add the icon of a computer program to a list of software applications. It should not be used to define standard icons that depend on the media type (html, pdf, image, audio, etc.) or on the quality class, and it should not be used to define thumbnail versions for a collection of images. See the chapter "Presentation of resources using icons" for a full discussion of this topic.

To alert users to changes in a resource collection, esp. new items or updated material, the attribute **UpdateStatus** can be used. The web presentation procedure can use this information, for example, to add appropriate icons to a link, or to provide a page that contains only new or updated material. The attribute has the following states:

ID	*Code*	*Description*
0	Normal	The resource has no special update status
1	New	The resource is newly introduced
2	Updated	The resource has been modified or updated

An optional mechanism is provided to record how one item has been derived from another item. It is desirable, if several versions of a resource item are maintained, e.g. when an image or text file is repeatedly modified, but the previous versions are still of interest. The attribute **ParentResourceID** can point to a resource item from which the current modified version has

been derived. Note that **ParentResourceID** is not restricted to resource IDs in the current resource collection.

It is assumed that the need for this mechanism is currently low, and that the information gained from this is rarely worth the additional effort to document the derivation of items from each other. The first versions of the user interface will therefore not implement this attribute. However, the attribute may be important if standard documents are to be managed, each version of which should remain permanently visible and quotable. Also, if software would directly interface the management system to record derivations, this would be very helpful.

Item parts

@@@ TO DO @@@

Item or item part description and keywords

@@@ TO DO @@@

Resource Indexing Categories

The resource instance

A resource instance is a digital representations of an abstract item (e.g., the thumbnail resolution of a photograph). The information on resource instances can be grouped into four main areas:

- URI and availability of resource instances
- Technical description of resource instance
- Automated link checking (tests the validity of the instance URI to verify the continued presence of a resource)

Note that in most cases the author or editor of a resource collection provides no information for a resource instance directly. The attributes are either automatically filled (URI- and availability cache attributes, technical description and link checking attributes) or are used only in exceptional cases (**URI_ManualOverride**).

URI and availability of resource instances

The attribute **URI_ManualOverride** is used in the few cases where the URI can not be managed by the URI model preferred by the DiversityResource model. This situation can be avoided for all resource collections locally managed, but it may occur when external

resources have to be integrated. The chapter "Introduction to the URI definition" describes the preferred URI model as well as situations where one must resort to **URI_ManualOverride**.

The remaining attributes **URI_Cache**, **AvailableInternetCache**, **AvailableIntranetCache**, and **AvailableRequestCache** are all calculated fields that may either be dynamically implemented using functions and stored procedures, or which may be calculated using triggers in the database engine to improve the response speed of the presentations that rely on this information. The calculations can be time consuming: Five attributes from four entities must be retrieved to calculate **URI_Cache**, and another five attributes from four entities to calculate each **Available...Cache**.

Technical description of resource instance

The following attributes record technical information about an instance of the resource. Most of this information is automatically obtained by software procedures and is automatically entered into the resource instance records.

MediaTypeID defines the media type (e.g. text/html or image/gif). A list of media types is defined in the entity SysMediaType. The definitions follow the types defined in the MIME standard (rfc 2046), but only a selection of all possible MIME media types is present in SysMediaType. SysMediaType further defines a list of file extensions that is used to automatically recognize a media type based on its URI extension. If no extension is present, e.g. because the web server uses content negotiation, the type can either be set based on content negotiation information, or manually.

ResourceSize = Size of resource in bytes (file size, or length of download stream); this may be used to add size indications after links to help users in making appropriate choices. The value is 0 if the size is unknown.

FileDate = The file date of those resource of which the file system date can be read. Used to identify updated instances and update those all other attributes that can be automatically obtained (e.g. size, pixel, etc.) are updated as well.

Specifically applicable to images are **ImgPixelHeight**, **ImgPixelWidth**, and **ImgColorDepth**. Note that web pages with embedded images display better if size information is embedded in the image tag. The color depth is stored for presentations where a comparison of image quality is desired.

Automated link checking

The validity of URIs to resource instances are automatically being checked. The application that does this should be designed as a scheduled, independent background task. The check tests on two criteria:

- Continuity of availability: if a link is broken for more than a defined time period, it will be automatically disabled
- Reliability of availability: The failure rate of a link is recorded.

The continuity is recorded in the resource instance attributes **CheckAvailability_LastResult** (Success or Failure), **CheckAvailability_LastTestWhen** and **CheckAvailability_LastSuccessWhen** (both date and time). The reliability is recorded in the counter attributes **CheckAvailability_Attempts** and **CheckAvailability_Failures** (both long numbers), with **CheckAvailability_ResetWhen** defining the date both counters have been reset to zero.

When the link fails to satisfy the predefined conditions it is automatically deactivated by setting **CheckAvailability_Deactivated** to true and **CheckAvailability_DeactivatedWhen** to the current date. The link will remain in the database until an editor decides that is shall be ultimately deleted.

The frequency of link checking depends on the number of links in the resource manager, and the additional load the checking produces on the internet connection. As a starting point, each link could be retested every 3 days. In a situation where it is acceptable to have occasionally non-functional links, it may be desirable to deactivate a link only after it failed for at least 7 days, perhaps even only after 14 days. This reduces the management overhead and avoids links being deactivated when a site is down only temporarily. A downtime of several days is not unusual in a scientific and amateur environment.

Question: A desirable functionality would be to detect automatic forwards. Which database attributes would be necessary to automatically update links by parsing out the forwarding code? This should be implemented in a future version, not in the first version!

Introduction to the URI definition

DiversityResources has two methods to store the complete path to a resource:

1) It is possible to enter the path for an image for each resolution of an image manually, using the attribute **URI_ManualOverride** present in resource instances. Since this method is very labour intensive, it is not recommend. **Note:** **URI_ManualOverride** is not supported in the first version of the web interface!

2) The path can be split into up to 4 parts: A first part that is common to all images in a collection, a second part that optionally defines a folder for a given QualityLevel (e.g. thumbnail, compressed jpg for the web, high quality version), a third part

Examples for the application of URI parts

	Stem of URI	Quality Prefix	Specific part [1]	Quality Suffix

Attribute: *defined in:*	*URI_Part1* *Collection*	*URI_Part2* *QualityClass*	*URI_Part3* *Item*	*URI_Part4* *QualityClass*
1. Collection of links from various web servers (normally only a single quality level will be defined for s collections)				
		–	www.xy.net/target1.html	–
	http://	–	www.abc.net/qry/search.ht ml	–
2. Collection of images, quality versions are structured using file extensions and suffixes				
		–	ma_001	_tn.gif
	http://www.xy.net/img/as	–	ma_001	.jpg
	c/	–	ma_001	.png
		–	ma_002	_tn.gif
3. Collection of images, quality versions are structured using folders				
		thumbnails/	ma_001	.gif
	http://www.xy.net/img/as	web/	ma_001	.jpg
	c/	hires/	ma_001	.png
		thumbnails/	ma_002	.gif
4. Collection of images, quality versions are structured using root folders				
		thumbnails/asc0	ma_001	.gif
	http://www.xy.net/	0/		
		img/asc00/	ma_001	.jpg
5. Large collection of images which is dispersed through subfolders, quality versions use file extensions				
		–	001_100/ma_001	_tn.gif
	http://www.xy.net/img/as	–	001_100/ma_001	.jpg
	c/	–	101_200/ma_150	_tn.gif
		–	101_200/ma_150	.jpg

[1] (resource item specific, but in the case of images applicable to all resolutions)

Splitting the URI into 4 parts, of which only the item part is not optional, allows to structure one own resources according to a simple but still flexible model. It is designed to be normative for the storage strategies employed for ones own resource collections. All resources in a collection should be placed in a folder, optionally containing subfolders and the quality version should either be distinguished using folders or file extensions plus optionally suffixes.

The model can further efficiently manage a resource site that has a mirror sites. This situation will be relatively rare in the biodiversity research, so that no explicit design elements where introduced for that purpose. However, it is possible to use two quality classes "primary location" and "mirror location". The collection attribute URI_Part1 is then left empty (except perhaps for "http://"), and the two URI stems are defined in the QualityClass attribute URI_Part2.

Note that certain storage strategies are *not* supported by the model described above. Firstly, and probably most noticeable, it is not possible to apply the model when images within a quality version have various file extensions, but not all different quality versions have always the same extension. Example:

http://www.xy.net/img/asc/thumbnail/ma_001.gif
http://www.xy.net/img/asc/web/ma_001.**jpg**
http://www.xy.net/img/asc/thumbnail/ma_150.gif
http://www.xy.net/img/asc/web/ma_150.**png**

Further, it is not possible to structure collections into folders *and* to use another folder level for quality versions. For example, the URIs

http://www.xy.net/img/asc/**001_100/thumbnail/**ma_001.gif
http://www.xy.net/img/asc/**001_100/web/**ma_001.jpg
http://www.xy.net/img/asc/**101_200/thumbnail/**ma_150.gif
http://www.xy.net/img/asc/**101_200/web/**ma_150.jpg

can not be managed with the model. In contrast, a storage model where the quality designation folder is placed first:

http://www.xy.net/img/asc/**thumbnail/001_100/**ma_001.gif
http://www.xy.net/img/asc/**web/001_100/**ma_001.jpg
http://www.xy.net/img/asc/**thumbnail/101_200/**ma_150.gif
http://www.xy.net/img/asc/**web/101_200/**web/ma_150.jpg

can be managed successfully.

It was considered whether the model should be further developed, introducing for example an additional URI_Part5 in each item which could handle the variable extensions. Currently it was decided that this would make the model too difficult to explain to the average user. Instead, an override mechanism has been added in the Instance entity. If the attribute **URI_ManualOverride** is filled, the system expects a complete URI in this attribute and ignores all other URI parts for that instance. This should allow to manage any external resources that do not fit the preferred storage or URI model. However, in these cases, the management of the URI (e.g. in the case of URI changes) is less efficient.

It may further be noted that the model of quality classes allows to use a resource collection with only a single quality class to collect mixed resources of any type, but that it does not allow to have mixed resources in the same collection together with resources for which several quality levels exist. It will therefore often be necessary, to have one collection of images, and a related collection of web links.

Point for discussion: Should a mix of images with an orthogonal quality structure and mixed other links without quality versions be made possible? Perhaps by defining an attribute orthogonal in some classes (or use the SingleClass attribute), and define an abstract item attribute "IsOrthogonal" versus "MixedContent" or something like this? How important is a mixture of images, audio files, and text links, or should such a mixture rather be defined on a separate level? It would be possible to define "Presentations" of resource items (from several resource collections), as opposed to collections of content/URIs!

Availability management

The availability of a resource (e.g. an image) is defined through a combination of attributes in multiple entities. The first two elements of the model are the availability scope and the access immediacy, which are defined for each abstract item and each quality class of a resource collection.

The availability scope is defined in the attributes **ItemAvailability** and **QualityAvailability**. Four user communities can be defined:

ID	*Description*
0	not available; access restricted to administrators and managers of the resource collection
1	locally available in the intranet ("in-house use")
2	publicly available in the internet to everyone

Note: The local/intranet user community may not be supported if the web server setup can not distinguish internal from external access (e.g. based on IP numbers).

The access immediacy is defined in the attributes **ItemAvailableOnline** and **QualityAvailableOnline**. It is a Boolean attribute that can be displayed in a user interface as a checkbox or using the following pick list:

ID	*Code*	*Description*
0	Request	Available to the public only on explicit request, e.g. by electronic or postal means
-1	Online	Directly online available within intranet or internet (depending on separate availability definition)

The possible interactions of availability scope and access immediacy is shown in the following table. Note that the access immediacy of resource collection managers can not be restricted, the instance are always online available to this user group!

Tab. Overview over the applicability and desirability of availability options:

	AvailableOnline	
Availability	***Online***	***On Request***
only to managers	Frequent	(Not applicable)
login/project	Frequent	Rarely desirable

local/intranet	Frequent	Desirable: colleagues may ask for access to, e.g., high resolution images
public/internet	Frequent	Meaningful situation, although less important

Availability scope and access immediacy must be defined separately for each quality class of a resource collection (**QualityAvailability**, **QualityAvailableOnline**) and for each resource item (**ItemAvailability**, **ItemAvailableOnline**). The definition in a quality class enables resource collection managers to provide low and medium resolution images on the internet, but reserve the use of high quality or original version images to the local intranet. The additional definition of an item specific availability allows to preliminary include resources in the collection that need clarification whether they should be made available to the public. Examples are images from a resource collection which should be visible only after the publication of a scientific article, or resources that need further copyright clearance. The availability of a resource instance is defined as the minimum of the quality and the item availability. If the item is available in the internet, but the quality class of the instance only to managers, the instance can be accessed only by managers. Similarly, the instance is available online if both the **QualityAvailableOnline** and **ItemAvailableOnline** are 'true'. The default value for **ItemAvailability** and **QualityAvailability** is 'public/internet', the default value for **QualityAvailableOnline** and **ItemAvailableOnline** is 'true'.

While it is not possible to define the availability of an entire collection for different user communities (internet, local intranet, etc.), a separate mechanism exists to restrict access to the entire resource collection up to a specific date in the future (**ReleaseDate**). The release date is set by the collection manager and may be up to two years in the future. The mechanism is intended to provide researchers an option to prepare the collection for personal use, but prevent the public from viewing it while it is under construction or until the publication date of a scientific publication. The default release date is set for immediate release. **ReleaseNotes** are required whenever a **ReleaseDate** postpones the public release and are intended to document the motives for the postponement. Note that it is also possible to disable all quality classes to make the entire collection unavailable on the internet. However, the preferred method is to use the **ReleaseDate** for this purpose.

Finally, a similar mechanism automatically prevents resources being released to the public that have not been reviewed by an editor responsible for the web site. The editorial review mechanism implements the necessary editorial supervision of the web site, preventing illegal or undesirable information from being published. Editorial review allows resource collections to be managed by many collaborators, including those outside of the organization responsible for the content of the web site. The mechanism is especially important to prevent abuse if a mechanism to upload images by external users is provided.

The resource collection attributes **EditorialReleaseBy** and **EditorialReleaseWhen** are set by privileged users (responsible for the content of the web site) after the content of new items or instance in a collection has been reviewed. If a resource collection contains only unmanaged

content (links pointing to resources on other web sites) the responsibility for the resource collection may be released from editorial control, by setting **EditorialControl** to false. This effectively grants editorial control to the managers of the resource collection. **EditorialControl** can be changed only by responsible managers of the internet site, not by resource collection managers. If it is set to false, **EditorialReleaseBy** and **EditorialReleaseWhen** are updated to document by whom and when editorial control was granted to the resource collection managers.

Note that the release date restricts the availability of the entire resource collection and is compared with the current date, whereas the editorial release date restricts the availability of instances and is compared with the creation date of an instance.

> ***Note:*** *The web interface should offer an option in the direct presentation procedure to display all items and instances of a collection that were changed after* **EditorialReleaseWhen**. *The mode should only be available to privileged editors responsible for the web site. A button would be displayed to release all items and instance in the collection after they have been reviewed.*

In summary, a resource instance is available to a client if the following conditions are met:

- the quality class for the collection must be available to the client, and
- the resource item (abstract item) must be available to the client, and
- the current date must be after the release date of the resource collection (**ReleaseDate**), and
- the editorial release date (**EditorialReleaseWhen**) of the collection must be later than the creation date of the instance or editorial control is not required (**EditorialControl** = False).

Presentation of resources using icons

For resources accessed through hyperlinks (i.e. which are not directly embedded into the presentation), it is often desirable to display a resource specific icon instead of or in addition to the textual link. Three cases should be distinguished:

a) The resource is an image and a strongly reduced preview version of the image itself ("thumbnail") shall be displayed as an icon. A presentation procedure that wants to achieve this should search for a quality class with the "**Abridged**" attribute set. If the instances of this type have a web compatible image media type, they can be used as thumbnails.

Point for discussion: This seems to be relatively complicated, since it is possible to have a quality class with quality level defined as **Abridged** that does not refer to media (e.g. to excerpts from a video). Can this case be made simpler, without complicating the model too much, or asking too much of the resource collection author?

b) If the presentation procedure intends to symbolize the media type or the quality level of the resource, it can use the entity **SysRsc_MediaTypeIcon**. This entity defines standard icon URIs for all recognized media types and all quality levels. Different media types are

represented by different icons to easily distinguish text, formatted text, images, etc., and different quality levels of the same media type (e.g. resolutions of images) can similarly be distinguished through different icons. However, the definitions in **SysRsc_MediaTypeIcon** apply to the entire resource management module. It is not possible to use different icons for different resource collections.

Task: The table needs to be filled. I have a collection of some media type dependent icons. These need to be placed on the web server, preferably in a separate folder, and entered in the table, always for all recognized media types. I have preset the tables media types and media type icons, but the icon URI that are entered do not exist yet! Perhaps also use the IP number instead of the server name, to speed up things!

c) If a resource has a specific icon or logo associated with it, this icon can be manually defined for each abstract item in the attribute **LinkAssociatedIcon**. The attribute contains a complete URI pointing to an icon. Examples where resource specific icon would be desirable are: a list of taxonomic software applications, a list of societies, or a list of online journals. Adding icons or logos to these list enables the user to recognized known entries more quickly because of the recognition value of the icon images. If no icon or logo is available for some entries, these could either be displayed without an icon, or generic icons (e.g. for "Windows program", "Unix program", "online journal", etc.) could be added.

> ***Note:*** *The URI in* **LinkAssociatedIcon** *is not managed by the DiversityResources management system. Most importantly, it is not included in the regular link checking mechanism implemented in ResourceInstances. An alternative implementation would have been to point to a ResourceID that defines the image to be used as icon. However, this would have meant that a separate ResourceCollection would have to be created prior to using the icons in the main collection. It was decided that the considerable implementation overhead as well as the complication of use does not balance the advantages of such a solution. (Note that it is not very practical to define icons as a quality class in addition to a mixed quality class for html-links, since the abstract item (including URI_Part3) is assumed to apply to all classes within a collection. The icons would thus have to have a similar name and URI path to the target resource.)*

Finally, a related task is to mark new or updated resources with an icon, alerting the user who is reviewing a resource collection repeatedly to new material. The resource item contains the attribute **UpdateStatus** where the status (normal, new, updated) can be defined. The icon associated with an update status is defined in the web presentation procedure, not in the database.

Retrieval and presentation usage cases

This chapter discusses the retrieval of information from resource collections and different types of presentation modes. The following usage cases are discussed:

Direct presentation of an image collection
Direct presentation of a link list
Explicit integration into external applications
Automatic supplementation of information

Direct presentation of an image collection

For resources containing images, the resource manager should contain all necessary elements to create a web presentation similar to a picture-based identification book. The information necessary for the title page of the resource collection includes: Title, authors or editors of collection, first publication date. The images should then be arranged by a primary sorting criteria (e.g. species/infraspecific), possibly grouped by headers (e.g. genus/family, obtained from DiversityTaxonomy) and structured into several pages (genus/family?). A secondary sorting criterion could be used to create an index.

The interface would present first preview ("thumbnail") images. Each thumbnail has a link that points to the next higher quality class which is available in the given access mode (internet or intranet). Alternatively, the presentation of the entire collection can be changed from preview mode to full view. This is especially desirable in the context of biological identification, since it is often important to study details while browsing through the images and comparing them with the specimen being identified.

Any collection can be viewed in presentation mode, allowing authors and editors to browse through the entire collection and search for inconsistencies. It is not possible to prevent a resource collection, the resources of which have been made publicly available, to be viewed as a collection. However, access to the resource collection requires knowledge of an URI for that collection. If this URI is not integrated in another web page, it is practically not visible.

Point for discussion: this implies that no publicly available list of "All resource collections" is published!

Task: To motivate external contributors, the editing form of a collection should always inform (e.g. at the end) about the URI by which the resource collection can be publicly browsed. Copying this link into another web page should be as easy as possible. A text for this message could be: "If you would like to create a link to the resource collection, the URI is: http://www.xxx.net/xy/xy.cfm"

Direct presentation of a link list

The case of a direct presentation of a link list occurs if a resource collection is used as an explicit element of a web presentation. The link list may be reached from a link on a static web page, or from other dynamic content (e.g. a menu system). A typical links list supported by Diversity References has the following elements:

Heading
(Optional author, editor, acknowledgement for entire collection)
Subheading
Title of link 1: Description of link 1

Title of link 2: Description of link 2

Regarding the presentation of link, link title, and description, the standard presentation interface could provide several alternative formats:

Format 1	**Format 2**	**Format 3**	**Format 4**
No descriptions	Descriptions in same paragraph	Descriptions as separate paragraph	Descriptions displayed in separate pop-up window
Title of link 1	Title of link 1: Description of link 1	Title of link 1 Description of link 1	Title of link 1 (more)
Title of link 2	Title of link 2: Description of link 2	Title of link 2 Description of link 2	Title of link 2 (more)
Title of link 3	Title of link 3: Description of link 3		Title of link 3 (more)

In the last format, more could be replace by an icon image. Similarly, the link itself could be normal text, followed or preceded by an icon symbolizing a link, behind which the URI for the resource is placed. All elements should be formatted through cascading style sheets.

In addition, an icon can be associated with the textual hyperlinks, compare the chapter "Presentation of resources using icons" for a full discussion of the various situations. This creates additional formatting options. An example for such a link list could be a list of online-journals, with the journal title, the icon of the journal if present, and a short description describing the thematic scope of the journal.

Interface definition for procedure generating link lists

The procedure (as COM or JAVA object, or Coldfusion custom tag) should define the following parameters:

LinkTitleMode: 0 = resource title is output without hyperlink (meaningful only in combination with **LinkIconURI**); 1 = normal, underlined link; 2 = hidden link without colouring or underlining
LinkIconURI: the URI of an icon image symbolizing a link, can be output before or after the link, can be used to link instead of in addition to the hyperlink behind the title text.
LinkIconPlacement: 0 = not output; 1 = before title text; 2 = after title text.

LinkDescriptionMode: 0 = no description, 1 = same paragraph, 2 = separate paragraph, 3 = in pop-up window

??? **LinkWithSubheading** ????
??? **LinkSorting** ???

Explicit integration into external applications

Resources may be integrated manually into presentations generated by other elements of the information system. An example is a digital taxonomic monograph (for example on a genus or family of organisms), with taxonomic information obtained from DiversityTaxonomy, interactive identification keys and descriptions obtained from DiversityDescriptions, and images obtained from DiversityResources. In this case author of the monograph would explicitly define within DiversityDescriptions which resources shall be displayed for a given character definition or item description. Often these resources will be managed by the author of the monograph in her or his own resource collection, but if well managed resource collections from other workers are available, these may be used as well.

Explicit integration is also relevant for interactive identification applications, e.g. to select images illustrating the states of a character. The identification interface could directly use the images to select a character state, or provide the images only when a detailed definition of a character or character state is requested.

Important entry points: ResourceID, optional QualityLevel to define a resource instance, and optional PartID to refer to a resource part. Resource collection ID is not necessary, since ResourceID is globally unique.

Point for discussion: link to resource item or instance?

Automatic supplementation of information

In this case the information system automatically attempts to find relevant information from all available resources for a given information context. The most frequent use will be that the identification system (DiversityIdentify) could narrow the choice of taxa to one or few scientific names and that further information to finish or verify the identification shall be presented. Together with the presentation of these tentatively identified names, the information system automatically queries DiversityResources about supplementary information on these names.

Table X: Examples for queries for supplementary resources

Retrieve all resources that concern ...	**Media types**
Unspecific taxon queries:	any
a given taxon	any

Specific taxon queries:	text
the taxonomy or nomenclature of a given taxon	text
the geographic distribution of a given taxon	text, images (maps!)
the association of a given taxon with other organisms (e.g. host-parasite interaction)	text (images)
the morphological description of a given taxon	text, images
other properties (chemical, acoustic, molecular, etc.) of a given taxon	any
the usages or applications of a given taxon	text, images
bibliographic references covering a given taxon	text
additional static or interactive keys to the taxon or related taxa	text
Specific taxon, media type and feature queries:	
images of a given taxon displaying the germination of spores	images, text with embedded images
Non-taxon queries:	
definition or description of a descriptive characteristic or feature	any
any given keyword (example: 'mycological on-line journals')	any

Further interesting: something is a "taxon page", i.e. an author tried to provide comprehensive information on a taxon, including e.g. identification key, description, images, and perhaps references and links to further resources

Presentation: If the query is relatively wide (e.g. show all resources concerning given taxa), a presentation of automatically retrieved resources could be first structured by taxon, and then further structured using

- thematic subheadings, or
- icons added before or after the resource link
- list of keywords added after the resource link (perhaps only those of a given category like morphology)

Examples for resources of special interest to the user that could be marked using icons are: images, static keys or interactive identification interfaces, high quality, comprehensive pages like taxon pages.

xx_IsIdentificationInterface i.e. query interface, interactive identification

Problem: xx_IsIllustrationsPage: Name quatsch, aber Frage was machen mit Seite die nur etliche Bilder hat, nicht wirklich TaxonPage ...

In this process the adequacy of information has to be assessed and redundant resources should be filtered out. An option should also be presented to retry the search for supplementary

information by going up one level in the taxonomic hierarchy as presented in DiversityTaxonomy (e.g. find information on the genus, if the information on the species is unsatisfactorily).

The information should be presented in a meaningful, structured way, with subheadings for different information categories (e.g. descriptive illustrations, references, geographic distribution (this may include the media type image again!) or links to further interactive identification systems.

Since the information comes from several collections and possibly from several servers in this case, it is important to adequately acknowledge the source of information. This will usually be done in the form of hyperlinked endnote markers, linking to a list of acknowledgements at the end of the web page. If images are directly embedded in the web page, the acknowledgement is added directly to the image caption. The format of the acknowledgement is defined in the acknowledgement definition (a separate module of the DiversityWorkbench).

Perhaps general Keywords, definitely Taxon Names (as string, ?? also as NomID)

a) Technical

- Query interface e.g. identification or references
- Text only
- Article or monographic item (perhaps only chapter)
- "Multimedia"/"Multiresource" page: text plus images, reference, hyperlinks etc.
  - Species page = containing descriptions and further information (distribution, references, etc.) on a single species. Illustrations or other media content may be linked or embedded.
  - Taxon page = containing comprehensive information on a higher taxon (genus, family, etc.), possibly including identification keys
- Illustration (drawing, photograph, diagram)
- Video stream
- Audio stream

→ partly covered by media type in instances!

b) Information covered (thought for taxon):

- Nomenclature
- Description
- Geographic Distribution
- Identification Key (dicho/polytomous, tabular, etc.)
- Interactive identification interface
- References
- Applications/Uses = including ethnobotany

c) Not taxon

- //? Character definition
- //? Online Journal

Hierarchy of links, e.g. book -> chapter ???

Editing and workflow usage cases

This chapter discusses the editing of resource collections building them up. The following usage cases are discussed:

@@@

Automatic keywords

When adding items to a collection it should be possible to define keywords and organism names that apply to all items of a collection.

A simple implementation is to define default keywords and taxon names which are automatically added to any new item. However, if a collection is build without prior planning, the result may be surprising

Collection of resources from external coworkers (including upload procedures for images).

Ask for information (name, postal address, phone number, email address and also password, but no login name), then send email containing the password. This establishes a connection, which for normal purposes may be trusted (although it can be faked).

During this process the user must assert that he or she will observe all relevant regulations regarding intellectual property and copyright.

Procedure must in any case be well defined and documented, so that copyright infringement processes are unlikely.

Ideas for the workflow and layout of the web interface

1. Login of new user

Login through DiversityUsers. If no login exists yet, offer a web form for self-registration as new contributor. Full Name, telephone number, email, Address, and Login Name and Password are required, other items like homepage are optional. The external users are entered in DiversityUsers (with function = 5, Workgroup = "EXTERNAL", and permissions for resources set).

2. Switch board

Ask what to do:

- Edit existing resource collection
- Create a new resource collection for web page links
- Create a new resource collection for images available in a single resolution
- Create a new resource collection for images available in multiple resolutions
- Create a new resource collection in minimal mode

3. Create a new resource collection:

Depending on the options in the switchboard dialog, offer the editor with the appropriate quality classes already added. .Always display and edit the resource collection together with all quality classes already defined. The quality class attributes should be displayed in a tabular arrangement after the resource collection attributes. Provide buttons to delete a quality class or add another quality class (use a pick list for both buttons, do not add a "Delete" button after each existing class, deleting is rare!).

"A resource collection defines a sharply defined collection of images, e.g. all images for a fungal group that have a similar provenance and copyright situation. An examples for a well defined collection are all images that have been scanned (be sure to have the permission!) from a publication. The resource collection simplifies the management of these images: some information needs to be entered just once. Normally, all images in a collection should have a single path. If you store different image resolutions in different paths, this can be specified in the quality class."

A resource collection should not be identified with a taxonomic or thematic group, however. The collections of all images of the fungus *Ascochyta* that have been drawn or photographed personally, or a collection scanned with permission from a publication are both good collections. They should not, however, be managed in a single collection, since the copyright situation (and the necessary acknowledgements) are strongly different. The taxonomic of thematic circumscription of a collections should be defined in the description

3.x Minimal mode

Minimal mode is designed to allow the management of resource collections with minimal effort and without requiring to understand the full structure of the resource management system. The first time a user requests to enter resources in minimal mode, a new resource collection is created with a title based on the name of the user (for example for the user name "Gregor Hagedorn (Berlin)" the title would be "Gregor Hagedorn (Berlin) default resource collection"). Only a single QualityClass is added and all attributes of the quality class are disabled in the editing forms for minimal mode.

The only functionality is to import or define URIs directly, using the **URI_ManualOverride** attribute and to request a title, an optional caption and optional keywords for each resource item. All other attributes are set to their default values and are not available in the editing form.

Point for discussion: Should minimal mode require consent to upload of a mirror copy of the images?

Batch import

To simplify importing many resource files or URIs, it is possible to provide a item import batch mode, where the web editor presents a large text box for each quality class. The user can then enter the URIs, one per line, into the text box, and a program attempts to create the items. The URI-parts for collection and quality classes should already be appropriately defined at this time. The import procedure can then trim the redundant parts off the item URIs, and create the necessary item and instance record. If the data entered do not fit the predefined URI model, ie. the instances for multiple quality classes can not be simplified to a single item name, a warning should be offered. The user can then cancel the import, or accept that all instances are imported using the instance specific URI override.

4. Editing abstract items

The following figures (fig. XXX-XXX) show outlines how an editor could be implemented. The use of resource parts is infrequent. Therefore, if no resource parts have been defined so far, the initial form does not offer an empty resource part subform. To enter the first resource part, the button "Define new resource part" must be clicked. Two empty subforms are then provided, allowing the definition of two resource parts. Clicking on the button "Define new resource part" again adds two further resource part subforms. If a resource parts already exist, all parts are output plus an additional empty resource part subform.

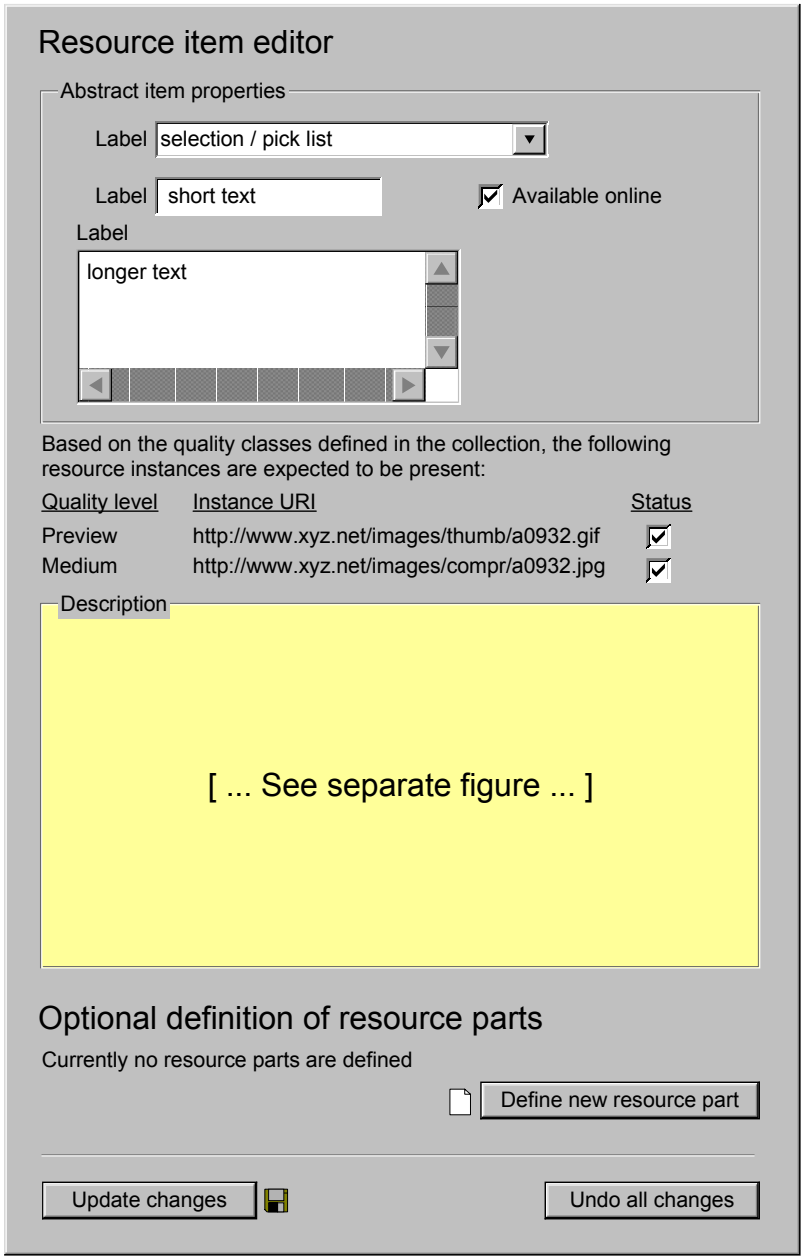

Fig. EditItem1: An outline of the abstract item editor. If no resource part has been defined, the optional subform is not displayed.

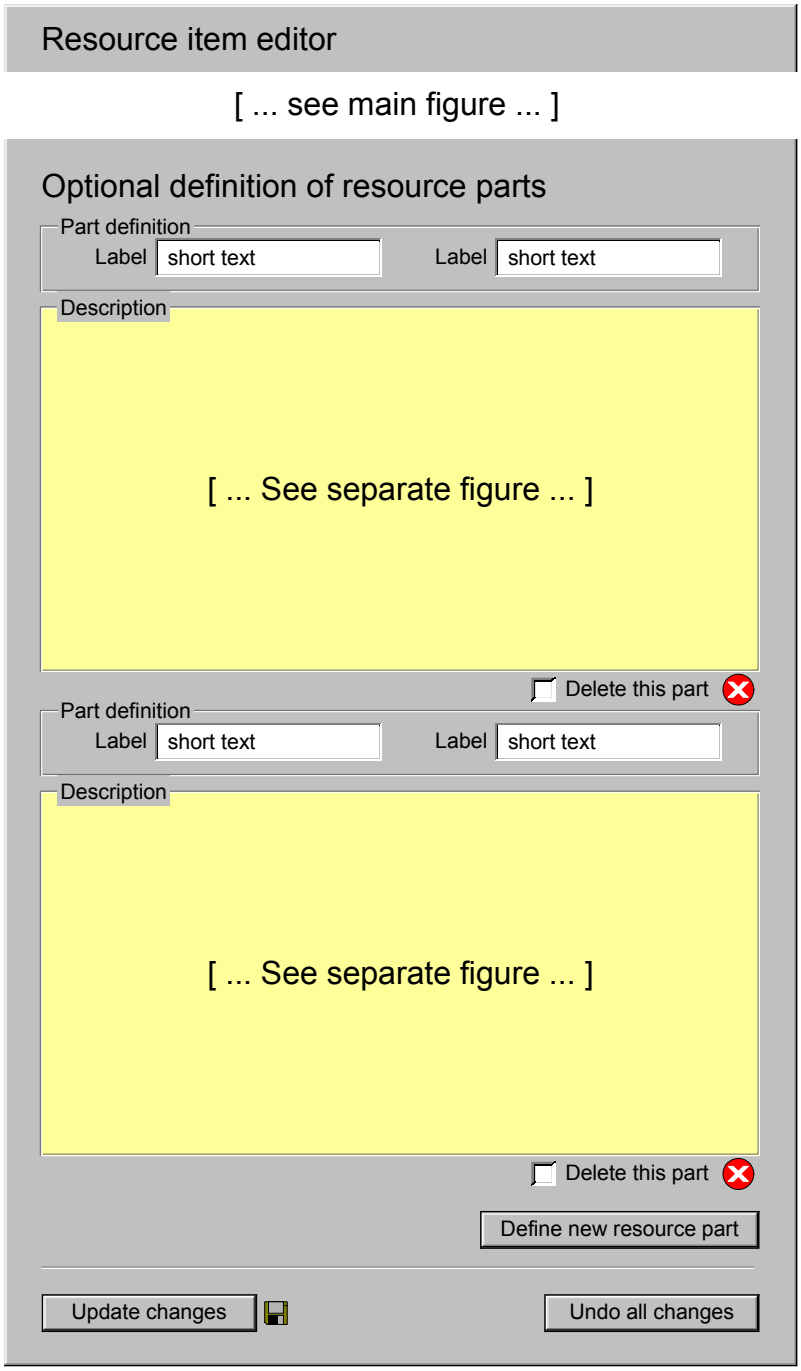

Fig. EditItem2: Resource item editor as in the previous figure, with the optional subform for two resource parts displayed.

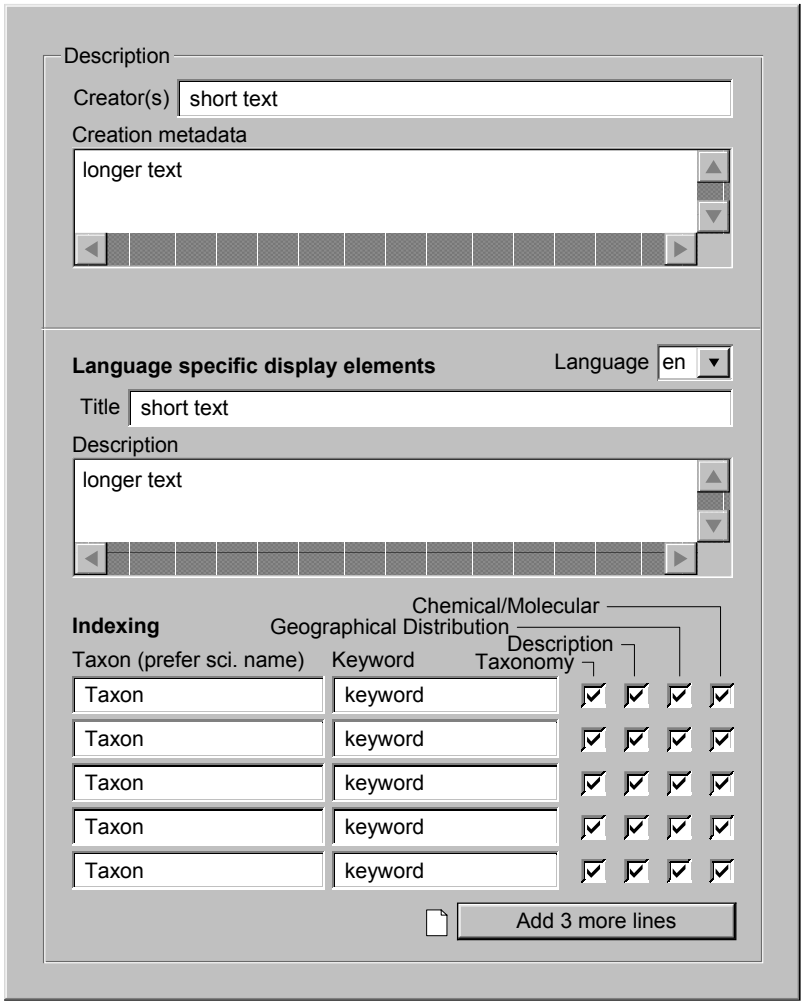

Fig. EditItem3: An outline of the description part. A resource description may apply to the entire resource, or multiple resource parts.