

SYNTHESYS – NA3  
Task 1.4

## **Annotation system report**

J. Holetschek

Botanischer Garten und Botanisches Museum  
Berlin

# Contents

1. Introduction and Objectives .....	2
2. Current Implementation .....	3
2.1 Basic Annotation Workflow.....	3
2.2 Creating an Annotation .....	5
2.3 Displaying Annotations in Data Portals .....	6
3. Limitations of the current System and recommendations for an improved system.....	7
3.1 Missing Functionalities as claimed by users .....	7
3.2 Usability .....	8
3.3 Other issues .....	10
4. Priorities and Feasibility.....	11
4.1 Making the XML an optional part of an annotation.....	11
4.2 Advanced XML editing features .....	12
4.3 Redesign of collection manager's notification.....	13
5. Conclusions and Outlook .....	13
References .....	16
Appendix: List of Recommendations.....	17

## 1. Introduction and Objectives

Over the past ten years, international initiatives such as the Global Biodiversity Information Facility (GBIF<sup>1</sup>), the Biological Collection Access Service for Europe (BioCASE<sup>2</sup>) and the Network Activity D of the SYNTHESYS project<sup>3</sup> have set up an infrastructure for connecting primary biodiversity data from various distributed sources. The goal is to make the existing data on planetary biodiversity freely and universally available for everyone on the Internet, to be used for research and as an information resource for decision makers and nature conservationists.

The information in question is primary biodiversity data, i.e. direct observations of organisms in nature, or of specimens either preserved or cultivated. It embraces the billions of specimens from all organism groups in preserved and living collections worldwide as well as the billions of observations that have been recorded, e.g. from vegetation surveys or bird monitoring. Currently, GBIF and SYNTHESYS provide access to nearly 196 million records in 10,361 datasets from 313 different data providers, each one documenting the existence of a species on a certain location at a certain point in time.

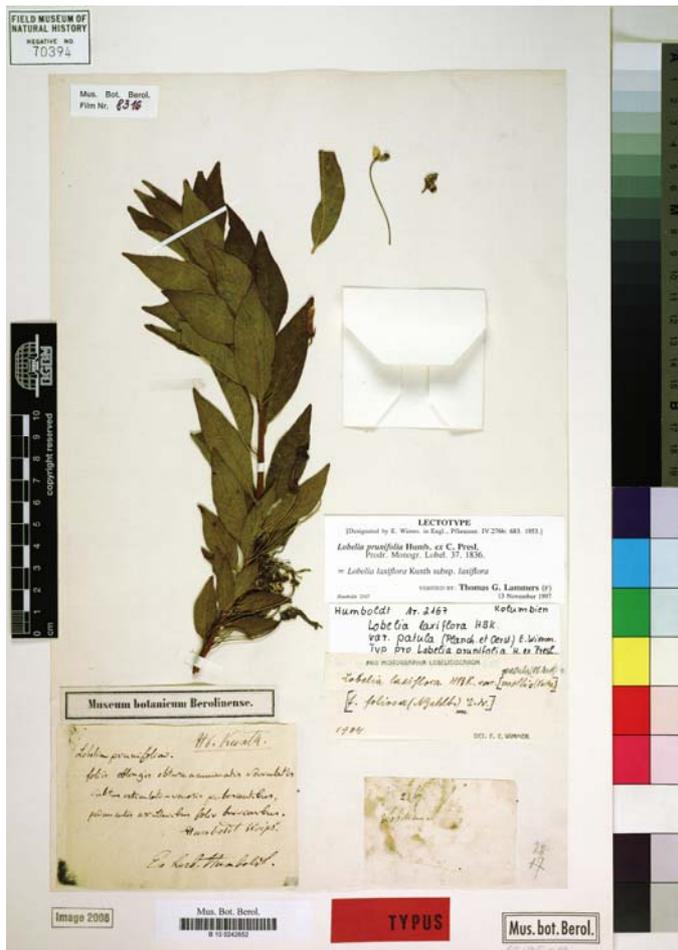
As for the data stemming from traditional natural history collections, each record corresponds to a real world specimen located in some institution, preserved in some way, for example in alcohol, stuffed or dried and mounted on a cardboard. Each specimen is usually labelled with a piece of paper holding storage details and information provided by the original collector. Over time, a variety of other information is added in the form of annotations, representing either corrections, confirmations or results of scientific analyses (see figure 1). Annotations increase the value of biological collections and ensure that scientists working with the collections have access to the results of previous scientific studies associated with the objects of interest.

---

<sup>1</sup> <http://www.gbif.org>

<sup>2</sup> <http://www.biocase.org>

<sup>3</sup> [http://www.synthesys.info/na\\_d.htm](http://www.synthesys.info/na_d.htm)



**Figure 1: Annotations on the physical specimen in the collection**

The systems described above now provide access to many million of objects through web-based data portals, but have also interrupted the traditional flow of annotation information in both directions: Neither will an annotation added after the digitisation of a specimen be visible on the data portals (unless the specimen is digitised again), nor will a user of the data portal be able to add an annotation to the digital record of the specimen. The first problem could be solved by providing additional personnel for digitisation efforts; the second one lead to the idea of “virtual annotations”, which is the rationale for the SYNTHESYS annotation system. The goal is to create a mechanism which enables a user to add annotations to the digital representation of a collection specimen, of which the collection manager will be informed and, moreover, which will be displayed to subsequent users on the data portal.

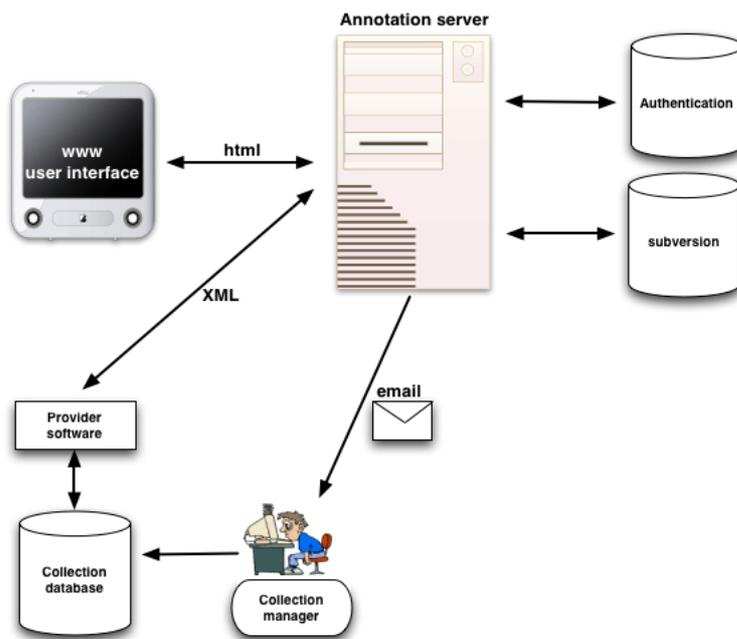
Sections 2.1 and 2.2 were taken from Güntsch et al. 2009 after slight modifications.

## 2. Current Implementation

### 2.1 Basic Annotation Workflow

The possibility of comparing the annotations for a biological collection object with its original label information is central to both the traditional annotation system and its electronic surrogate in biodiversity information networks. The SYNTHESYS annotation system was

therefore built around a central server for handling versions of documents. Rather than developing this server ourselves, we decided to use the existing open-source revision control system Subversion<sup>4</sup>, which is primarily intended for handling documents in the context of software programming, but is also an ideal platform for version control related to collection records. In addition to the subversion document repository, the annotation server connects to an authentication database containing registered users of the system and their passwords (Figure 2). Within the present architecture, there are no further checks of the trustworthiness of authenticated users.



**Figure 2: Annotation data flow (taken from Güntsch et al. 2009)**

Being based on the GBIF/SYNTHESYS technical infrastructure, the collection data themselves remain with the collection holders in their own database systems. Network accessibility is realized by means of standard provider software implementing either of the protocols for networking biodiversity data, BioCASE<sup>5</sup>, DiGIR<sup>6</sup>, or TAPIR<sup>7</sup>, together with the XML data schemas ABCD<sup>8</sup> or DarwinCore<sup>9</sup>. Thanks to the provider software, collection data are always retrievable in a standardized format no matter how they are stored in the collection holding institution.

Once a user has changed a record using the annotation system, it is stored in the subversion database together with all previous versions as a new version of this collection record. The collection holder then receives a message by email containing the information that one of his messages has been changed, as well as the URLs of both the original and the changed records on the server. There are innumerable freely available software tools supporting subversion which can then be used to analyze the changes. These include diff-tools to compare documents, as well as tools to graphically display the version history of a record.

<sup>4</sup> <http://subversion.tigris.org>

<sup>5</sup> <http://www.biocase.org/products/protocols>

<sup>6</sup> <http://www.digir.net>

<sup>7</sup> <http://www.tdwg.org/standards/449>

<sup>8</sup> <http://www.tdwg.org/standards/115>

<sup>9</sup> <http://www.tdwg.org/standards/450>

Based on these data, collection holders are then able to make an informed decision about whether they want to ignore an annotation or import the modified record into their database.

The software interface between collection-portals and the annotation system is simply the GBIF “triple-ID” consisting of institution, collection identifier, and an identifier of the particular collection record, with which the system can be “plugged in” in to different data portals with comparatively little effort. In addition, because it is based on transfer and storage of XML-documents, it is generic with respect to the collection data schema used.

## 2.2 Creating an Annotation

**BioCASE** **SYNTHESYS Annotation System**

- Prototype -

[Change password](#) [Logout](#)

The edited XML file is valid!

**Annotate XML document:**  
Note: date must be entered in ISO format (for example 2002-06-03T23:02:42)

```
<Identifications>
<Identification>
<Result>
<TaxonIdentified>
<HigherTaxa>
<HigherTaxon>
<HigherTaxonName>PANDANACEAE</HigherTaxonName>
<HigherTaxonRank>familia</HigherTaxonRank>
</HigherTaxon>
</HigherTaxa>
<ScientificName>
<FullScientificNameString>Pandanus utilis</FullScientificNameString>
<NameAtomised>
<Botanical>
<GenusOrMonomial>Pandanus</GenusOrMonomial>
<FirstEpithet>utilis</FirstEpithet>
</Botanical>
</NameAtomised>
</ScientificName>
</TaxonIdentified>
</Result>
```

**Comments (will be displayed on the BioCASE/SYNTHESYS/EDIT Portals):**

**Personal message to the provider :**

© 2007 Freie Universität Berlin, Botanischer Garten und Botanisches Museum Berlin-Dahlem  
Version 1.03, A. Güntsch, P. Kelbert, W.-H. Kusber, C. Oancea, E. Zippel & W. G. Berendsohn  
[BGBM Impressum/Imprint](#); [Haftungsausschluss/Disclaimer](#)

**Figure 3: Interface for creating an annotation**

At the current stage, creating an annotation can be accomplished by using a plain HTML interface which can be used with any World Wide Web browser. It is simply linked to each collection record (details view) in the BioCASE portal<sup>10</sup>. Users using the system for the first time are prompted to register themselves with a spam-safe registration form. Once logged in successfully, the annotation form is opened offering two fields: one for editing the collection record, and one for entering a free-text comment (figure 3).

<sup>10</sup> <http://search.biocase.org/europe>

Editing the collection record is done straightforwardly in its XML-representation (ABCD or one of the various variants of DarwinCore). When submitting modified records, an XML-validation algorithm checks whether the records are well-formed and valid. If not, the system comes back to the user with a list of potential problems. Occasionally, users may be unable to express their annotation in the terminology provided by the collection record itself. In these cases, the comment field can be used and the collection record remains unchanged.

## 2.3 Displaying Annotations in Data Portals

The screenshot shows the BioCASE web interface. The header includes the BioCASE logo and the text 'Biological Collection Access Service for Europe'. The navigation bar contains links for Home, Search, Units, Annotation, Thesaurus configuration, Preferences, Help, and a European Union flag. The main content area is titled 'Annotations' and displays a table with one row of annotation data:

Date	Title	Author	
2009-09-22	Scientific name should read: <i>Polygonum cedrorum</i> Boiss. & Kotschy according to IPNI (ID 695536-1).	Wolf-Henning Kusber	See

Below the table, there are sections for 'Units details', 'Identification(s)', and 'Gathering'. The 'Units details' section shows the unit ID (271690) and source institution (INPA). The 'Identification(s)' section shows the name of the species (*Polygonum cedrorum*). The 'Gathering' section shows the locality (Israel, Haifa region) and collector(s) (Kupat Rimonim).

Figure 4: Annotation in Track Changes Mode

Presenting annotations for a digital specimen imposes three tasks for a data portal, namely

1. Getting a list of potential annotations,
2. Retrieving data records for each of the annotations, and
3. Displaying the annotations along with attached comment, and, optionally, displaying the modifications, i.e. the differences between original and annotated record, in a convenient manner.

For the first task, SYNTHESYS has developed an annotation feed: Provided with the triple ID (institution, collection and record identifier) of a collection record, it returns a list of past annotations, each item consisting of annotation date, comment, the creator's name and an URL of both the annotated and the original record. These URLs simply direct to Subversion's

repository, reducing the second task of retrieving the data to a plain HTML GET request for the data portal.

The third task is a bit more cumbersome: In addition to just displaying the list of annotations and the annotated records themselves, the pieces of the record that the annotation refers to should be presented in a highlighted manner. The BioCASE portal, for example, uses a view known from word processor's track changes mode (see figure 4): Changes and additions are printed in a different colour, whereas deletions are crossed out.

### **3. Limitations of the current System and recommendations for an improved system**

#### ***3.1 Missing Functionalities as claimed by users***

##### **Insufficient availability of original XML documents**

The XML document to be annotated will be retrieved from the original data provider before it is presented to the user. If it is not available (because the server is not accessible or it was not able to fulfil the request), an annotation cannot be created. Unfortunately, this happens quite often, since most data providers are not very reliable. This can be very frustrating for a user of the data portals who has discovered a supposed error and just wants to annotate this.

- R1. In case the original XML from the data provider is not available, the system should still allow the user to send a notification to the collection manager. The annotation will consist of the comment only and no XML records will be stored in the subversion repository.*

##### **Plain notification function**

It was noted by users of the annotation system that in some cases it would be useful to have the opportunity to send a non-public comment to the collection manager. Even though this might be useful indeed, it is contrary to the concept of an annotation, which is by nature intended for the public. Therefore this is not a needed functionality of the annotation system, but the data portals itself.

##### **Bulk annotations**

An annotation is currently bound to a single occurrence record. However, if there are systematic errors in the records, an annotation may refer to multiple records - for example if the geographic coordinates are swapped for all records, or equal 0 instead of being empty, a very common mistake. This should be supported by the annotation system.

- R2. It should be possible to bind an annotation to a single record, a number of records or all records of a dataset.*

This has two consequences:

1. The triple ID (institution\_code, collection\_code, catalogue\_number), which holds the information which records the annotation refers to, must be enabled to hold a set of

values for the catalogue\_number as well as a NULL value (in case all records are affected).

2. Both the data portals and the annotation system must support the creation of bulk annotations. The editing of XML documents must be disabled; a bulk annotation cannot contain modified XML documents.

### **Missing “Reverse Wrapper” functionality**

Currently, the annotation system only informs the collection manager of supposed errors in certain occurrence records of his or her dataset. It is up to him or her to decide whether or not to accept this suggestion, and to change the data in the underlying collection database. There is no support whatsoever by the annotation system for this, it serves purely as a communications tool.

In a perfect world, the annotation sent to the collection manager would feature an “Accept” button, which directly changes the affected data in the collection database. Of course this is not feasible, since neither the database nor the underlying data model is known to the annotation system.

- R3. The annotation system should try to deliver the proposed changes to the collection manager in a format that easily allows the transfer into the collection database. This could be a tabular overview of several annotations bundled together, one row per record, with column pairs of old and proposed value. This would allow a print-out of the corrections that could be easily handed to someone for putting them into effect. This would be a big step ahead, compared to the system right now, with each annotation resulting in a separate email that doesn't even list the changes explicitly, but only points to XML documents.*

### **Missing version control**

If the database (and thus the delivered records) is changed in reaction to an annotation, there is currently no mechanism in place to track the changes connected to the respective annotations, thus leaving in place annotations which may refer to errors already corrected in the underlying database.

- R4. The annotation system should support the change towards globally unique identifiers (GUIDs) that allow the versioning of collection records. Annotations will then always be connected to the exact version of the record they originally referred to. However, as in other parts of the biodiversity information domain (e.g. names), the exact mechanisms for managing versions has still to be investigated.*

## **3.2 Usability**

### **Low responsiveness**

In a case study (Kusber et al. 2009) it was discovered that only a very small percentage of annotations were reacted upon by the collection managers. In contrast, the responsiveness was higher when an email with a personal message was sent to the manager directly, instead of using the annotation system.

This is very likely due to the fact that the message sent to the collection manager by the annotation does look too much like an automated message (which it is, of course). Moreover, it does not give clear instructions which steps should be done in order to tackle the problem.

R5. *The message sent by the system to the collection holder should be redesigned completely. It should address the recipient in a personal way and present the content of the annotation (the proposed changes) in a more apparent way, instead of just placing links to the original and the changed records, so it will be easier for the collection holder to decide on whether to accept or decline the annotation. It should describe clearly the steps necessary to accept the proposed changes (edit collection database, inform GBIF in order to trigger re-indexing).*

### **Errors and hints of XML validation are presented in too technical words**

Before an annotation gets accepted by the system, the XML document edited by the user will be validated against the schema definition. Invalid documents will be declined, the user will be prompted the error messages of the validation check in order to re-edit the XML (figure 5). These messages are much too technical and arcane for inexperienced users.

R6. *Error messages and hints of the XML validation process should be translated into a more easily understandable format.*

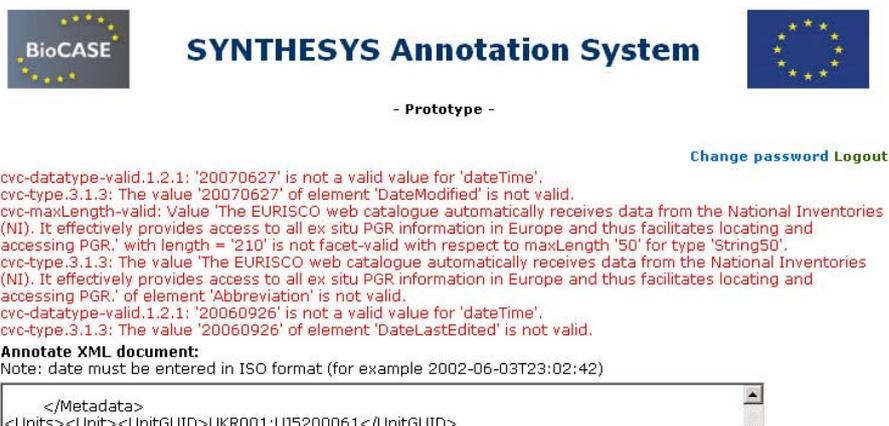


Figure 5: Arcane XML validation error messages

### **Editing the record data**

Currently, an annotation consists of two parts: The main part is the raw XML document of the occurrence record, which can be edited in a text box in order to reflect the changes proposed by the annotator. This can include editing the content of ABCD/DwC elements, adding new elements and removing data (even though this will be rarely the case). In addition, the annotation comprises a free-text comment field, which can be used for suggestions the annotator is unable or unwilling to enter as XML.

Directly editing XML code is not a user-friendly approach. Adding new elements is only possible with detailed, in-depth knowledge about the structure of ABCD/DwC, which only very few of the users will have. Editing existing content is easier to accomplish, but as XML documents can become quite large, finding the right piece of information can be difficult.

Even removing elements must be done with utmost care in order not to invalidate the XML document.

*R7. For editing the occurrence record, a substantially extended interface needs to be developed. At least for the most common elements in ABCD/DwC, some form should be created (probably semi-automatically from the XML document) allowing the user to directly edit the contents of the fields. Adding new elements (either completely empty sub-trees or new items for repeatable elements) should be possible for common ABCD/DwC fields.*

### **Simple Mode versus Expert Mode**

Even if editing the XML document can be done by pushing buttons for adding elements and by editing contents in input fields rather than in one textbox (see above), the user has to think of the annotation in a structured manner. At least some basic understanding of XML documents and the underlying data standards will be necessary. This is a high barrier for an inexperienced user, who is probably not willing to spend a lot of time for creating a simple annotation.

*R8. Besides the opportunity to edit the XML document (= Expert Mode), it should be possible to switch to a very simple and basic interface allowing only to send a comment to the collection manager (= Simple Mode).*

## **3.3 Other issues**

### **Link between annotation and annotated record**

In the GBIF and BioCASE networks, an occurrence record is uniquely defined by three identifiers, namely one for the providing institution, the source collection and a unique identifier of the record within the collection, for example a catalogue number. This triplet is called the “triple ID”, an example is (BGBM, AlcaterraTypes, 1584-A) and serves as a link between an annotation and the annotated record; it is used directly as a storage path within the subversion repository.

Consequently, changing one of these three identifiers will detach an annotation from the occurrence record and make it an orphan. This is a serious problem, but since the cause is not within the architecture of the annotation system, there is no way for the annotation system to handle this. As long as identifiers, which by nature should be stable, change, it will be impossible to keep track of the linkage because it's impossible to identify the original record.

### **Invalid Source XML documents**

The XML document edited by the user will be validated against the schema definition before it gets accepted. Unfortunately, many of the original XML documents are already invalid. There are two main reasons for this:

1. Some Provider Software packages deliver invalid XML documents in some cases.

2. The schema used is not a well-defined standard. This holds true especially for the DarwinCore standard, which has several flavours used in the networks. Not all of them have a unique namespace, so validation is performed against the wrong schema definition.

Both problems are not within the scope of the annotation system. As long as there is not strict XML validation implemented in all network components and no strict standards compliance, these problems will persist.

*R9. The annotation system should, under certain circumstances, accept invalid documents. However, this should only happen after the user has been prompted and agreed. In any case, the XML document should be required to be well-formed.*

## 4. Priorities and Feasibility

In discussions with users of the annotation system it became clear that there are three areas of issues that have a high priority.

1. The first group are the restrictions imposed by the fact that a valid XML document is mandatory for being able to create an annotation. This is highly problematic if the original XML document from the provider is unavailable, invalid or the user has not the knowledge to edit XML (recommendations R1, R9 and R8). Making the XML optional would lower the barrier for creating an annotation considerably.
2. The current way of changing the XML document – editing in a text box – is unacceptable for a user not familiar with the used XML schemas. Making the XML an optional part of the annotation will make this less urgent, but nevertheless an easier interface for XML manipulation is desirable.
3. Apart from just being visible on the data portals, collection managers/holders should react to the annotations, either by rejecting because the proposed change was found wrong, or acceptance, which ultimately would result in a change of the original data. A successful annotation is one that leads to a clear reaction on the collection holder side. The system should provide for such annotation answering statements.

Judged from this perspective, the annotation system has not provided sufficient incentive, because most of the annotations triggered no reaction at all. (Of course, this could mean rejection by the collection holder. However, most of the annotations were created after thorough checking by a recognised biologist, so this is very unlikely). So of the main priorities should be to raise the level of acceptance of annotations by collection managers (R3).

### 4.1 Making the XML an optional part of an annotation

In the current implementation, the XML is a mandatory part of an annotation, whereas the comment (free text) is optional. Removing this restriction would ease some of the problems encountered with the current system.

This would affect all components of the system, which would need to be modified:

- The interface for creating an application would have to accept comment-only annotations. In case the user does not want to edit the XML, or in case the original XML is unavailable or invalid, the comment becomes mandatory; otherwise an annotation would be completely empty.
- The annotation database would have to allow comment-only annotations to be stored.
- The data portals must be prepared for the case that an annotation consists of only comment or the XML. In addition, they must allow an annotation to be created always, regardless of whether the original data is available.

All of these modifications should be possible with moderate efforts. However, since all components are affected by the changes, a considerable amount of development time is required.

## 4.2 Advanced XML editing features

Editing XML documents consists of three basic operations:

### Editing the content of XML elements

A possible solution for this could be the automated form creation by means of style sheet transformation. For example, the source XML

```
<ScientificName>
  <FullScientificNameString>Juncus triglumis L.</FullScientificNameString>
  <NameAtomised>
    <Botanical>
      <GenusOrMonomial>Juncus</GenusOrMonomial>
      <FirstEpithet>triglumis</FirstEpithet>
      <AuthorTeam>L.</AuthorTeam>
    </Botanical>
  </NameAtomised>
</ScientificName>
```

Could be transformed into four input boxes:

ScientificName	
ScientificNameString	<input type="text" value="Juncus triglumis L."/>
NameAtomised	
Botanical	
GenusOrMonomial	<input type="text" value="Juncus"/>
FirstEpithet	<input type="text" value="triglumis"/>
AuthorTeam	<input type="text" value="L."/>

This would, of course, only allow the editing of already existing elements. Moreover, this could sometimes produce large forms. But since input boxes would be created only for already existing content, and if the form reflects the XML structure as indicated in the mock-up picture above, it would still allow easy editing.

### Removing existing XML elements

This task could be easily accomplished by removing empty elements from the XML after editing. The user would only need to delete the contents in the input boxes; the XML validator could eliminate these elements subsequently.

### **Adding new XML, including whole sub-trees**

This is definitely the most difficult editing functionality; it requires a far more complex interface than the one simple JSP forms used in the current implementation. Since most users have no deeper knowledge of the XML schemas used, it is questionable whether a function that for example allows adding an arbitrary ABCD 2.06 element would be used at all. ABCD 2.06 has approximately 1,000 elements, finding the desired one is very difficult, no matter how the interface would look like.

So reducing this functionality from “adding any XML element” to “adding XML elements for commonly existing content” would both increase the usability for the annotator and reduce the efforts required for development. A range of buttons, e.g. “add identification” or “add geographic coordinates” would shield the annotator from the details of the underlying data schema and easy-to-use. The disadvantage would be that it’s not possible to add any element of the used schema; but still, this could be included in the accompanying free-text comment.

### ***4.3 Redesign of collection manager’s notification***

As described before, experience has shown that the responsiveness is relatively low. Changing the email notification so that it addresses the collection manager more directly, clearly displays the annotation contents and describes the necessary steps for accepting the annotation should be possible with moderate efforts.

## **5. Conclusions and Outlook**

The resources for enhancements of the annotation system are very limited within the SYNTHESYS II work plan, which allots only two person months for developments. Consequently, not all of the recommendations listed in section 3 (and the appendix) can be implemented. Even the changes discussed in section 4 must be prioritised in order to solve the most urgent issues with the limited resources.

The most pressing restrictions are the ones discussed in section 4.1. The changes proposed are rather minor, but affect all components of the annotation system. But as a side effect, it diminishes the need to implement the modifications listed in section 4.2, since the XML document is no longer a mandatory part of an annotation. Keeping this side effect in mind, the redesign of the notification message discussed in 4.3 should be given the higher priority, so it would end up as second item in the priority list.

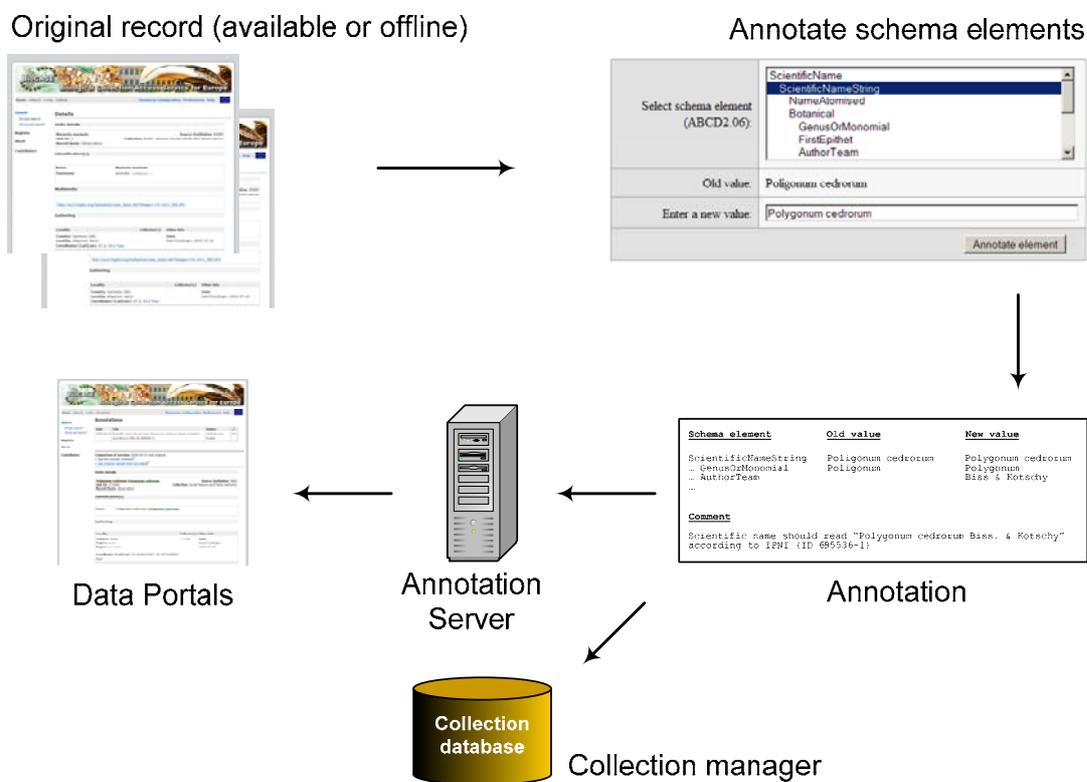
This would make the priority as follows:

1. Making the XML optional, discussed in section 4.1,
2. Redesigning the notification message, discussed in 4.3, and
3. Enhancing the XML editing functions, as described in 4.2.

Judging the basic concept of the annotation system after some experience with the used prototype leads to a mix of conclusions:

The fundamental idea of seeing an annotation as the difference between original and modified XML document, and storing it exactly according to this notion, makes the annotation system generic in respect to the used data schemas, and hence also applicable for data standards that will be developed in the future. Storing the XML documents in a Subversion repository makes them easily available on the web and allows the usage of the many available diff tools for tracking changes.

However, real-world XML documents (being invalid in many cases), real-world web services (being unavailable for many data providers) and the much too permissive usage of data standards lead to severe problems in everyday use. As long as these deficiencies exist, strictly binding an annotation to the XML will be problematic. Moreover, the issue of unstable identifiers makes virtual annotations vulnerable to data loss; the usage of long-run permanent identifiers, in conjunction with central registries for these, might overcome this.



**Figure 6: Potential architecture for future annotation system**

Future developments should concentrate on a more pragmatic approach, taking into account the mentioned deficiencies. Splitting up each annotation into several, independent pieces could restrict problems to certain parts of the annotation, not blocking creating an annotation altogether. A closer look at general initiatives like Annotea<sup>11</sup> and their findings is helpful.

Figure 6 shows a potential architecture for a future system: It would treat annotations in a semi-structured manner and can be seen as a compromise between the two extremes of storing an annotation as a highly structured XML document - as it is done by the current implementation - and storing it in a completely unstructured way, for example as a simple free-text. Thereby, it would avoid many of the problems described earlier while not

<sup>11</sup> <http://www.w3.org/2001/Annotea/>

sacrificing all advantages a structured storage offers, namely the opportunity of automated processing.

An annotation would consist of one or several pieces, each one referring to one of the concepts used by the data schema the original record was provided in. The user could be provided with a list of all concepts of the used data schema and a text box for entering the new value. This would result in a number of triples (schema\_concept, old\_value, new\_value). Together with an additional comment they would be seen as one annotation for the record and could be stored on the annotation server.

This would eliminate many of the drawbacks of the current system:

- An annotation could be created for a record whose XML document is currently not available. In this case, the original value could not be stored, but it would still be possible to annotate the record. For element schemas that are stored in the indexing database, these values could be used.
- Invalid XML documents would not cause troubles, since there will be no XML validation any more.
- Editing an XML source document is not need any more; adding and removing elements would be as easy as editing the content.
- The user would not be required to understand the structure of the XML schema any more. Nevertheless, the XPath elements used to identify schema elements can be quite unhandy. Creating an interface that shields the user from these internals would be one of the main challenges.
- The annotation would only consist of changes. Thereby it would be easier for the collection manager to fully understand the annotation than it is now.

## References

Anton Güntsch, Walter G. Berendsohn, Pepé Ciardelli, Andrea Hahn, Wolf-Henning Kusber & Jinling Li 2009: “Adding content to content – a generic annotation system for biodiversity data”. *Studi Trent. Sci. Nat.* 84: 123-128.

Wolf-Henning Kusber, Elke Zippel, Patricia Kelbert, Jörg Holetschek, Anton Güntsch & Walter G. Berendsohn (submitted): “Verbesserung europäischer Biodiversitätsdaten im Rahmen der EU-SYNTHESYS Networking Activities”. Deutsche Gesellschaft für Limnologie (DGL), Erweiterte Zusammenfassungen der Jahrestagung 2009 (Oldenburg).

Wolf-Hennin Kusber, Elke Zippel, Patricia Kelbert, Jörg Holetschek, Anton Güntsch & Walter G. Berendsohn 2009: „From cleaning the valves to cleaning the data: Case studies using diatom biodiversity data on the Internet (GBIF, BioCASE)“. – *Studi Trent. Sci. Nat.* 84: 111-122.

Anonymous: The Annotea project. <http://www.w3.org/2001/Annotea/>

## Appendix: List of Recommendations

- R10. In case the original XML from the data provider is not available, the system should still allow the user to send a notification to the collection manager. The annotation will consist of the comment only and no XML records will be stored in the subversion repository.*
- R11. It should be possible to bind an annotation to a single record, a number of records or all records of a dataset.*
- R12. The annotation system should try to deliver the proposed changes to the collection manager in a format that easily allows the transfer into the collection database. This could be a tabular overview of several annotations bundled together, one row per record, with column pairs of old and proposed value. This would allow a print-out of the corrections that could be easily handed to someone for putting them into effect. This would be a big step ahead, compared to the system right now, with each annotation resulting in a separate email that doesn't even list the changes explicitly, but only points to XML documents.*
- R13. The annotation system should support the change towards globally unique identifiers (GUIDs) that allow the versioning of collection records. Annotations will then always be connected to the exact version of the record they originally referred to. However, as in other parts of the biodiversity information domain (e.g. names), the exact mechanisms for managing versions has still to be investigated.*
- R14. The message sent by the system to the collection holder should be redesigned completely. It should address the recipient in a personal way and present the content of the annotation (the proposed changes) in a more apparent way, instead of just placing links to the original and the changed records, so it will be easier for the collection holder to decide on whether to accept or decline the annotation. It should describe clearly the steps necessary to accept the proposed changes (edit collection database, inform GBIF in order to trigger re-indexing).*
- R15. Error messages and hints of the XML validation process should be translated into a more easily understandable format.*
- R16. For editing the occurrence record, a substantially extended interface needs to be developed. At least for the most common elements in ABCD/DwC, some form should be created (probably semi-automatically from the XML document) allowing the user to directly edit the contents of the fields. Adding new elements (either completely empty sub-trees or new items for repeatable elements) should be possible for common ABCD/DwC fields.*
- R17. Besides the opportunity to edit the XML document (= Expert Mode), it should be possible to switch to a very simple and basic interface allowing only to send a comment to the collection manager (= Simple Mode).*
- R18. The annotation system should, under certain circumstances, accept invalid documents. However, this should only happen after the user has been prompted and agreed. In any case, the XML document should be required to be well-formed.*

*RI.*